Lecture Notes in Networks and Systems 798

Subarna Shakya
João Manuel R. S. Tavares
Antonio Fernández-Caballero
George Papakostas *Editors*

Fourth International Conference on Image Processing and Capsule Networks

ICIPCN 2023

Springer

Lecture Notes in Networks and Systems

Volume 798

Series Editor

Janusz Kacprzyk, Systems Research Institute, Polish Academy of Sciences, Warsaw, Poland

Advisory Editors

Fernando Gomide, Department of Computer Engineering and Automation—DCA, School of Electrical and Computer Engineering—FEEC, University of Campinas—UNICAMP, São Paulo, Brazil

Okyay Kaynak, Department of Electrical and Electronic Engineering, Bogazici University, Istanbul, Türkiye

Derong Liu, Department of Electrical and Computer Engineering, University of Illinois at Chicago, Chicago, USA
Institute of Automation, Chinese Academy of Sciences, Beijing, China

Witold Pedrycz, Department of Electrical and Computer Engineering, University of Alberta, Alberta, Canada
Systems Research Institute, Polish Academy of Sciences, Warsaw, Poland

Marios M. Polycarpou, Department of Electrical and Computer Engineering, KIOS Research Center for Intelligent Systems and Networks, University of Cyprus, Nicosia, Cyprus

Imre J. Rudas, Óbuda University, Budapest, Hungary

Jun Wang, Department of Computer Science, City University of Hong Kong, Kowloon, Hong Kong

The series "Lecture Notes in Networks and Systems" publishes the latest developments in Networks and Systems—quickly, informally and with high quality. Original research reported in proceedings and post-proceedings represents the core of LNNS.

Volumes published in LNNS embrace all aspects and subfields of, as well as new challenges in, Networks and Systems.

The series contains proceedings and edited volumes in systems and networks, spanning the areas of Cyber-Physical Systems, Autonomous Systems, Sensor Networks, Control Systems, Energy Systems, Automotive Systems, Biological Systems, Vehicular Networking and Connected Vehicles, Aerospace Systems, Automation, Manufacturing, Smart Grids, Nonlinear Systems, Power Systems, Robotics, Social Systems, Economic Systems and other. Of particular value to both the contributors and the readership are the short publication timeframe and the world-wide distribution and exposure which enable both a wide and rapid dissemination of research output.

The series covers the theory, applications, and perspectives on the state of the art and future developments relevant to systems and networks, decision making, control, complex processes and related areas, as embedded in the fields of interdisciplinary and applied sciences, engineering, computer science, physics, economics, social, and life sciences, as well as the paradigms and methodologies behind them.

Indexed by SCOPUS, INSPEC, WTI Frankfurt eG, zbMATH, SCImago.

All books published in the series are submitted for consideration in Web of Science.

For proposals from Asia please contact Aninda Bose (aninda.bose@springer.com).

Subarna Shakya · João Manuel R. S. Tavares ·
Antonio Fernández-Caballero · George Papakostas
Editors

Fourth International Conference on Image Processing and Capsule Networks

ICIPCN 2023

Springer

Editors
Subarna Shakya
Department of Electronics and Computer Engineering, Pulchowk Campus, Institute of Engineering
Tribhuvan University
Lalitpur, Nepal

Antonio Fernández-Caballero
Universidad de Castilla-La Mancha
Albacete, Spain

João Manuel R. S. Tavares
Faculdade de Engenharia
Universidade do Porto
Porto, Portugal

George Papakostas
Department of Computer Science
International Hellenic University
Kavala, Greece

ISSN 2367-3370 ISSN 2367-3389 (electronic)
Lecture Notes in Networks and Systems
ISBN 978-981-99-7092-6 ISBN 978-981-99-7093-3 (eBook)
https://doi.org/10.1007/978-981-99-7093-3

© The Editor(s) (if applicable) and The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2023

This work is subject to copyright. All rights are solely and exclusively licensed by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.
The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.
The publisher, the authors, and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Singapore Pte Ltd.
The registered company address is: 152 Beach Road, #21-01/04 Gateway East, Singapore 189721, Singapore

Paper in this product is recyclable.

We would like to dedicate this proceeding to all members of advisory committee and program committee for providing their excellent guidance. We also dedicate this proceeding to the members of the review committee for their excellent cooperation throughout the conference. We also record our sincere thanks to all the authors and participants.

Preface

It is our pleasure to welcome you to the 4th International Conference on Image Processing and Capsule Networks [ICIPCN 2023] organized on August 10–11, 2023. A major goal and feature of the conference is to bring academia and industries together to share and exchange their significant research experiences and results in the field of imaging science, with a particular interest on the capsule network algorithms and models by discussing the practical challenges encountered and solutions adopted to it.

This conference will deliver a technically productive experience to the budding researchers in the field of image processing and capsule networks by stimulating a good awareness on this emerging research field. ICIPCN promises to provide a produce a bright landscape for the image processing research works, while the response received, and research eagerness witnessed, have truly exceeded our expectations. At the end of the conference event, we are overwhelmed with the high level of satisfaction.

The response for the conference event has been increasing at an unprecedented rate both from Thailand and overseas. Due to the professional expertise of both internal and external reviewers, the papers have been selectively accepted based on their extensive research and publication quality. We have received a total submission of 366 out of which only 48 papers were accepted for publication based on its research effectiveness and applicability. We would also like to extend our thanks to the members of the organizing committee for their hard work in delivering spontaneous response to all the conference participants.

We are now enthusiastic to get the proceedings of the ICIPCN conference event covered by Springer. We also appreciate all the authors of ICIPCN 2023 for their timely response to all the queries raised from the conference.

Finally, we would like to thank the Springer publications for producing this volume.

Lalitpur, Nepal — Prof. Dr. Subarna Shakya
Porto, Portugal — Dr. João Manuel R. S. Tavares
Albacete, Spain — Dr. Antonio Fernández-Caballero
Kavala, Greece — Dr. George Papakostas

Contents

Modern Challenges and Limitations in Medical Science Using Capsule Networks: A Comprehensive Review 1
Milind Shah, Nikunj Bhavsar, Kinjal Patel, Kinjal Gautam, and Mayur Chauhan

Studies on Movie Soundtracks Over the Last Five Years 27
Sofía Alessandra Villar-Quispe and Adriana Margarita Turriate-Guzman

Blind Source Separation of EEG Signals Using Wavelet and EMD Decomposition 39
H. Massar, M. Miyara, B. Nsiri, and T. Belhoussine Drissi

Image Extraction Approaches for Density Count Measurement in Obstruction Renography Using Radiotracer 99mTc-DTPA 57
Pradnya N. Gokhale, Babasaheb R. Patil, and Sameer Joshi

Deep Short-Term Long Memory Technique for Respiratory Lung Disease Prediction 73
B. Dhiyanesh, Y. Baby Kalpana, S. Rajkumar, P. Saraswathi, R. Radha, and S. Suresh

Utilizing Satellite Imagery for Flood Monitoring in Urban Regions 89
Priyanka Sakpal, Shashank Bhosagi, Kaveri Pawar, Prathamesh Patil, and Pratham Ghatkar

Optimizing Permutations in Biclustering Algorithms 115
Aditya Shreeram, Tanmayee Samantaray, and Cota Navin Gupta

Extracting Graphs from Plant Leaf Venations Using Image Processing 131
Ashlyn Kim D. Balangcod and Jaderick P. Pabico

Multispectral Fusion of Multisensor Image Data Using PCNN for Performance Evaluation in Sensor Networks 145
S. Dharini and Sanjay Jain

U-Net-Based Segmentation of Coronary Arteries in Invasive Coronary Angiography 157
A. Azeroual, Y. El Ouahabi, W. Dhafer, M. H. El yousfi Alaoui, B. Nsiri, and A. Soulaymani

Change Detection for Multispectral Remote Sensing Images Using Deep Learning 169
M. Lasya, Radhesyam Vaddi, and S. K. Shabeer

Explainable AI for Black Sigatoka Detection 181
Yiga Gilbert, Emmy William Kayanja, Joshua Edward Kalungi, Jonah Mubuuke Kyagaba, and Ggaliwango Marvin

Modified U-Net and CRF for Image Segmentation of Crop Images 197
Shantanu Chakraborty, Rushikesh Sanap, Muddayya Swami, and V. Z. Attar

Securing Data in the Cloud: The Application of Fuzzy Identity Biometric Encryption for Enhanced Privacy and Authentication 213
Chandrasekar Venkatachalam, K. Manivannan, and Shanmugavalli Venkatachalam

Quantum Convolutional Neural Network for Agricultural Mechanization and Plant Disease Detection 225
Musa Genemo

Innovative Method for Alzheimer's Disease Detection Using Convolutional Neural Networks 239
Md. Sajid Anam Ifti, Md. Redwan Ahmed, S. M. Arafat Rahman, Sheikh Shemanto Afridi, Sanjeda Sara Jennifer, and Ahmed Wasif Reza

Segmentation of White Matter Lesions in MRI Images Using Optimization-Based Deep Neural Network 253
Puranam Revanth Kumar, Rajesh Kumar Jha, and P. Akhendra Kumar

A New Multi-level Hazy Image and Video Dataset for Benchmark of Dehazing Methods 269
Bedrettin Çetinkaya, Yücel Çimtay, Fatma Nazli Günay, and Gökçe Nur Yılmaz

Creative AI Using DeepDream 281
Rakhi Bhardwaj, Tanmay Kadam, Shubham Waghule, Sahil Shendurkar, and Bhakti Sarag

Tuberculosis Bacteria Detection Using Deep Learning Techniques 297
Sharmin Akther Rima, Meftahul Zannat, Samia Shameem Haque, Al Kawsar, Sanjeda Sara Jennifer, and Ahmed Wasif Reza

An Enhanced Real-Time System for Wrong-Way and Over Speed Violation Detection Using Deep Learning 309
A. Manasa and S. M. Renuka Devi

U-Net-Based Denoising Autoencoder Network for De-Speckling in Fetal Ultrasound Images 323
S. Satish, N. Herald Anantha Rufus, M. Antony Freeda Rani, and R. Senthil Rama

Galo Lexical Tone Recognition Using Machine Learning Approach 339
Bomken Kamdak, Gom Taye, and Utpal Bhattacharjee

Tea Leaf Disease Classification Using an Encoder-Decoder Convolutional Neural Network with Skip Connections 353
Swati Shinde and Sagar Lahade

Deep Learning-Based Classification of Lung Cancer Lesions in CT Scans: Comparative Analysis of CNN, VGG-16, and MobileNet Models 373
P. M. Hareesh and Sivaiah Bellamkonda

A Novel Fingerprint-Based Age Group Classification Approach Using DWT and DFT Analysis 389
Ashwini Sanjay Gaikwad and Vijaya B. Musande

Exploring the Potential Use of Infrared Imaging in Medical Diagnosis: Comprehensive Framework for Diabetes and Breast Cancer Screening 411
Asok Bandyopadhyay, Himanka Sekhar Mondal, Barnali Pal, Bivas Dam, and Dipak Chandra Patranabis

Precision Agriculture Through Stress Monitoring in Crops with Multispectral Remote Sensing Data 425
Konumuri Kalyan Suhas, G. Kalyani, and Mandava Venkata Sri Sai Surya

EEG Signal Feature Extraction Using Principal Component Analysis and Power Spectral Entropy for Multiclass Emotion Prediction 435
S. Babeetha and S. S. Sridhar

Semantic Image Segmentation of Agricultural Field Problem Areas Using Deep Neural Networks Based on the DeepLabV3 Model 449
Aleksey Rogachev, Ilya Belousov, and Dmitry Rogachev

A Unified Deep Learning Model for Multi-Satellite Image Classification of Land Use and Land Cover 463
M. S. Babitha, A. Diana Andushia, and A. Mehathab

Multi-class Plant Leaf Disease Classification on Real-Time Images Using YOLO V7 475
P. Sajitha, Diana A Andrushia, and S. S. Suni

City Road Anomaly Alert for Autonomous Vehicles: Pothole Dimension Estimation with YOLOv5 491
Varun Vinod Kulkarni, S. Vishal, Monish Mohanty, and Peeta Basa Pati

Craniofacial and Cervical Aesthetic Surgical Outcome Analysis Using Visual Similarity Metrics 511
S. Abhishek, Prathibha Prakash, T. Anjali, and Sundeep Vijayaraghavan

Brain Tumor Recognition from MRI Using Deep Learning with Data Balancing Methods and Its Explainability with AI 523
Abdullah Al Noman and Abu Shamim Mohammad Arif

Histopathology Breast Cancer Classification Using CNN 539
M. Venkateshwara Rao, Rajesh Saturi, D. Srinivas Goud, G. Srikanth Reddy, and N. Venkatesh

Automated Wave Height Measurement and Analysis Through Image Processing: A Computer Vision Approach for Oceanography 551
Medha Wyawahare, M. Selva Balan, Anvay Deshpande, Priyanka Acharya, Ankita Kumari, and Ashfan Khan

Enhanced Feature Fusion from Dual Attention Paths Using Feature Gating Mechanism for Scene Categorization of Aerial Images 563
G. Akila and R. Gayathri

Thermal Image-Based Fault Detection Using Machine Learning and Deep Learning in Industrial Machines: Issues-Challenges and Emerging Trends 581
Vaishnavi V. Kulkarni, Vishwanath R. Hulipalled, Mayuri Kundu, Jay B. Simha, and Shinu Abhi

AdaptPSOFL: Adaptive Particle Swarm Optimization-Based Layer Offloading Framework for Federated Learning 597
Rachit Verma and Shajulin Benedict

The Brain Tumors Identification, Detection, and Classification with AI/ML Algorithm with Certainty of Operations 611
Pranay Meshram, Tushar Barai, Mohammad Tahir, and Ketan Bodhe

A Dimensionality Reduction Method for the Fusion of NIR and Visible Image 629
Lokesh Gopinath and A. Ruhan Bevi

Hand Gesture-Based AI System for Accessing Windows Applications 647
Lakshmi Harika Palivela, V. Premanand, and Afruza Begum

FlameGuard: A Smart System for Forest Fire Detection and Control 665
Medha Wyawahare, Shreya Daga, Chashu Agrawal, Pranav Dalve, and Chirag Bhandari

Classification and Analysis of Chilli Plant Disease Detection Using Convolution Neural Networks 677
Zameer Gulzar, Sai Chandu, and K. Ravi

Sign Language Recognition Using Long Short-Term Memory Deep Learning Model 697
Prabhat Mali, Aman Shakya, and Sanjeeb Prasad Panday

Explainable Artificial Intelligence and Deep Transfer Learning for Skin Disease Diagnosis 711
James Mayanja, Enoch Hall Asanda, Joshua Mwesigwa, Pius Tumwebaze, and Ggaliwango Marvin

A Comparison Study Between Otsu's Thresholding, Fuzzy *C*-Means, and *K*-Means for Breast Tumor Segmentation in Mammograms 725
Moustapha Mohamed Saleck, Nagi Ould Taleb, Mohamed El Moustapha El Arby Chrif, and El Benany Mohamed Mahmoud

Author Index 735

Editors and Contributors

About the Editors

Subarna Shakya is currently a professor of Computer Engineering, Department of Electronics and Computer Engineering, Central Campus, Institute of Engineering, Pulchowk, Tribhuvan University, Coordinator (IOE), LEADER Project (Links in Europe and Asia for Engineering, Education, Enterprise, and Research Exchanges), ERASMUS MUNDUS. He received M.Sc. and Ph.D. degrees in Computer Engineering from the Lviv Polytechnic National University, Ukraine, 1996 and 2000 respectively. His research area includes e-government system, computer systems and simulation, distributed and cloud computing, software engineering and information system, computer architecture, information security for e-government, and multimedia system.

João Manuel R. S. Tavares graduated in Mechanical Engineering at the Universidade do Porto, Portugal, in 1992. He also earned his M.Sc. degree and Ph.D. degree in Electrical and Computer Engineering from the Universidade do Porto in 1995 and 2001 and attained his Habilitation in Mechanical Engineering in 2015. He is a senior researcher at the Instituto de Ciência e Inovação em Engenharia Mecânica e Engenharia Industrial (INEGI) and an associate professor at the Department of Mechanical Engineering (DEMec) of the Faculdade de Engenharia da Universidade do Porto (FEUP).

João Tavares is a co-editor of more than 55 books and co-author of more than 50 book chapters, 650 articles in international and national journals and conferences, and three international and three national patents. He has been a committee member of several international and national journals and conferences, is co-founder and co-editor of the book series *Lecture Notes in Computational Vision and Biomechanics* published by Springer, founder and editor-in-chief of the journal *Computer Methods in Biomechanics and Biomedical Engineering: Imaging and Visualization* published by Taylor & Francis, editor-in-chief of the journal *Computer Methods in Biomechanics and Biomedical Engineering* published by Taylor & Francis, and co-founder

and co-chair of the international conference series: CompIMAGE, ECCOMAS VipIMAGE, ICCEBS and BioDental. Additionally, he has been a (co-)supervisor of several M.Sc. and Ph.D. thesis and supervisor of several post-doc projects and has participated in many scientific projects both as researcher and as scientific coordinator.

Antonio Fernández-Caballero received his Master in Computer Science from the School of Computer Science at the Technical University of Madrid, Spain, and he received his Ph.D. from the Department of Artificial Intelligence of the National University for Distance Education, Spain. He is a full professor at the University of Castilla-La Mancha, Albacete, Spain. His interests enforce him to be part of the membership in National Society Networks AERFAI (Spanish Association of Pattern Recognition and Image Analysis), RTNAC (National Natural and Artificial Computation Network), RedAF (Physical Agents Network), AIPO (Association of Human–Computer Interaction) and Spanish Technology Platform on Robotics (Hisparob) and European networks euCognition (The European Network for the Advancement of Artificial Cognitive Systems), and SIMILAR (The European taskforce Creating human–machine interfaces SIMILAR to human–human communication).

George Papakostas received the Diploma degree in electrical and computer engineering in 1999 and the M.Sc. and Ph.D. degrees in electrical and computer engineering in 2002 and 2007, respectively, from the Democritus University of Thrace (DUTH), Greece. From 2007 to 2010, he served as an adjunct lecturer with the Department of Production Engineering and Management, DUTH. He currently serves as an adjunct assistant professor with the Department of Computer and Informatics Engineering, Technological Educational Institution, Eastern Macedonia and Thrace, Greece. In 2012, he was elected as a full professor in the aforementioned Department of Computer and Informatics Engineering. He has co-authored more than 70 publications in indexed journals, international conferences, and book chapters. His research interests include pattern recognition, computer/machine vision, computational intelligence, machine learning, feature extraction, evolutionary optimization, and signal and image processing.

Contributors

Shinu Abhi REVA Academy for Corporate Excellence (RACE) REVA University, Bangalore, India

S. Abhishek Department of Computer Science and Engineering, Amrita School of Computing, Amrita Vishwa Vidyapeetham, Amritapuri, India

Priyanka Acharya Vishwakarma Institute of Technology, Pune, Maharashtra, India

Sheikh Shemanto Afridi Department of Computer Science and Engineering, East West University, Dhaka, Bangladesh

Chashu Agrawal Department of Electronics and Telecommunication Engineering, Vishwakarma Institute of Technology, Pune, India

P. Akhendra Kumar Department of Electronics and Communication Engineering, Faculty of Science and Technology (IcfaiTech), ICFAI Foundation for Higher Education, Hyderabad, India

G. Akila Department of Electronics and Communication Engineering, Meenakshi Sundararajan Engineering College, Chennai, India

Abdullah Al Noman Computer Science and Engineering Discipline, Khulna University, Khulna, Bangladesh

Diana A Andrushia Karunya Institute of Technology and Sciences, Coimbatore, India

T. Anjali Department of Computer Science and Engineering, Amrita School of Computing, Amrita Vishwa Vidyapeetham, Amritapuri, India

M. Antony Freeda Rani Department of Electrical and Electronics Engineering, DMI College of Engineering, Chennai, Tamil Nadu, India

S. M. Arafat Rahman Department of Computer Science and Engineering, East West University, Dhaka, Bangladesh

Abu Shamim Mohammad Arif Computer Science and Engineering Discipline, Khulna University, Khulna, Bangladesh

Enoch Hall Asanda Makerere University, Kampala, Uganda

V. Z. Attar School of Computational Science, COEP Technological University, Pune, India

A. Azeroual Research Center STIS, M2CS, National Graduate School of Arts and Crafts of Rabat, Mohammed V University in Rabat, Rabat, Morocco;
Group of Biomedical Engineering and Pharmaceuticals Sciences, National Graduate School of Arts and Crafts of Rabat (ENSAM), Mohammed V University, Rabat, Morocco

S. Babeetha Department of Computing Technologies, College of Engineering and Technology, SRM Institute of Science and Technology, Chengalpattu, Tamil Nadu, India

M. S. Babitha Department of Electronics and Communication Engineering, Karunya Institute of Technology and Sciences, Coimbatore, India

Y. Baby Kalpana CSE, P.A. College of Engineering and Technology, Pollachi, India

M. Selva Balan Central Water and Power Research Station, Pune, Maharashtra, India

Ashlyn Kim D. Balangcod Department of Mathematics and Computer Science, University of the Philippines Baguio, Baguio, Philippines

Asok Bandyopadhyay ICT and Servics Group, Centre For Development of Advanced Computing, Kolkata, India

Tushar Barai G.H.Raisoni College of Engineering Nagpur, Nagpur, Maharastra, India

Afruza Begum School of Computer Science Engineering, Vellore Institute of Technology, Chennai, India

T. Belhoussine Drissi Laboratory of Electrical and Industrial Engineering, Information Processing, Informatics, and Logistics (GEITIIL), Faculty of Science Ain Chock, University Hassan II—Casablanca, Casablanca, Morocco

Sivaiah Bellamkonda Computer Science and Engineering Department, Indian Institute of Information Technology, Kottayam, Kerala, India

Ilya Belousov Volgograd, Russian Federation

El Benany Mohamed Mahmoud Department of Mathematics and Computer Science, University of Nouakchott AL Aasriya, Nouakchott, Mauritania

Shajulin Benedict Indian Institute of Information Technology Kottayam, Kottayam District, Kerala, India

Chirag Bhandari Department of Electronics and Telecommunication Engineering, Vishwakarma Institute of Technology, Pune, India

Rakhi Bhardwaj Department of Computer Engineering, Vishwakarma Institute of Technology, Pune, India

Utpal Bhattacharjee Department of Computer Science and Engineering, Rajiv Gandhi University, Itanagar, Arunachal Pradesh, India

Nikunj Bhavsar Department of Computer Engineering, School of Engineering, P P Savani University, Kosamba, Surat, Gujarat, India

Shashank Bhosagi JSPM's Rajarshi Shahu College of Engineering, Pune, Maharashtra, India

Ketan Bodhe G.H.Raisoni College of Engineering Nagpur, Nagpur, Maharastra, India

Bedrettin Çetinkaya Computer Engineering, Ted University, Ankara, Turkey

Shantanu Chakraborty Department of Computer Engineering and IT, COEP Technological University, Pune, India

Sai Chandu SR University, Warangal, Telangana, India

Yücel Çimtay Computer Engineering, Ted University, Ankara, Turkey

Mayur Chauhan Department of Computer Engineering, Gujarat Power Engineering and Research Institute (GPERI), Mehsana, Gujarat, India

Shreya Daga Department of Electronics and Telecommunication Engineering, Vishwakarma Institute of Technology, Pune, India

Pranav Dalve Department of Electronics and Telecommunication Engineering, Vishwakarma Institute of Technology, Pune, India

Bivas Dam Department of Instrumentation and Electronics, Jadavpur University, Kolkata, India

Anvay Deshpande Vishwakarma Institute of Technology, Pune, Maharashtra, India

W. Dhafer Research Center STIS, M2CS, National Graduate School of Arts and Crafts of Rabat, Mohammed V University in Rabat, Rabat, Morocco;
Group of Biomedical Engineering and Pharmaceuticals Sciences, National Graduate School of Arts and Crafts of Rabat (ENSAM), Mohammed V University, Rabat, Morocco

S. Dharini CMR Institute of Technology, Bengaluru, India

B. Dhiyanesh CSE, Dr. N.G.P. Institute of Technology, Coimbatore, India

A. Diana Andushia Department of Electronics and Communication Engineering, Karunya Institute of Technology and Sciences, Coimbatore, India

Mohamed El Moustapha El Arby Chrif Department of Mathematics and Computer Science, University of Nouakchott AL Aasriya, Nouakchott, Mauritania

Y. El Ouahabi Laboratory Health and Biology, Faculty of Sciences, Ibn Tofail University, Kenitra, Morocco

M. H. El yousfi Alaoui Research Center STIS, M2CS, National Graduate School of Arts and Crafts of Rabat, Mohammed V University in Rabat, Rabat, Morocco;
Group of Biomedical Engineering and Pharmaceuticals Sciences, National Graduate School of Arts and Crafts of Rabat (ENSAM), Mohammed V University, Rabat, Morocco

Kinjal Gautam Department of Computer Science and Engineering, School of Computer Science Engineering and Technology (SoCSET), ITM (SLS) Baroda University, Vadodara, Gujarat, India

R. Gayathri Department of Electronics and Communication Engineering, Sri Venkateswara College of Engineering, Sriperumbudur, India

Musa Genemo Gumushane University, Gumushane, Turkey

Pratham Ghatkar JSPM's Rajarshi Shahu College of Engineering, Pune, Maharashtra, India

Yiga Gilbert Department of Computer Science, Makerere University, Kampala, Uganda

Pradnya N. Gokhale Sardar Patel College of Engineering, University of Mumbai, Mumbai, Maharashtra, India

Lokesh Gopinath Department of Electronics and Communication Engineering, SRM Institute of Science and Technology, Kattankulathur, Chengalpattu, Tamil Nadu, India

Zameer Gulzar SR University, Warangal, Telangana, India

Fatma Nazli Günay Computer Engineering, Ted University, Ankara, Turkey

Cota Navin Gupta Neural Engineering Laboratory, Department of Biosciences and Bioengineering, Indian Institute of Technology Guwahati, Guwahati, Assam, India

Samia Shameem Haque Department of Computer Science and Engineering, East West University, Dhaka, Bangladesh

P. M. Hareesh Computer Science and Engineering Department, Indian Institute of Information Technology, Kottayam, Kerala, India

N. Herald Anantha Rufus Department of Electronics and Communication Engineering, Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, Chennai, Tamil Nadu, India

Vishwanath R. Hulipalled School of C&IT, REVA University, Bangalore, India

Sanjay Jain CMR Institute of Technology, Bengaluru, India

Sanjeda Sara Jennifer Department of Computer Science and Engineering, East West University, Dhaka, Bangladesh

Rajesh Kumar Jha Department of Electronics and Communication Engineering, Faculty of Science and Technology (IcfaiTech), ICFAI Foundation for Higher Education, Hyderabad, India

Sameer Joshi Miraj Nuclear Medicine AND Molecular Imaging Center, Miraj, Maharashtra, India

Tanmay Kadam Department of Computer Engineering, Vishwakarma Institute of Technology, Pune, India

Joshua Edward Kalungi Department of Computer Science, Makerere University, Kampala, Uganda

G. Kalyani Department of Information Technology, Velagapudi Ramakrishna Siddhartha Engineering College, Vijayawada, India

Bomken Kamdak Department of Computer Science and Engineering, Rajiv Gandhi University, Itanagar, Arunachal Pradesh, India

Al Kawsar Department of Computer Science and Engineering, East West University, Dhaka, Bangladesh

Emmy William Kayanja Department of Computer Science, Makerere University, Kampala, Uganda

Ashfan Khan Vishwakarma Institute of Technology, Pune, Maharashtra, India

Vaishnavi V. Kulkarni School of CSE, REVA University, Bangalore, India; Department of Computer Science and Engineering, Amrita School of Computing Bengaluru, Amrita Vishwa Vidyapeetham, Bengaluru, India

Varun Vinod Kulkarni Department of Computer Science and Engineering, Amrita School of Computing Bengaluru, Amrita Vishwa Vidyapeetham, Bengaluru, India

Puranam Revanth Kumar Department of Electronics and Communication Engineering, Faculty of Science and Technology (IcfaiTech), ICFAI Foundation for Higher Education, Hyderabad, India

Ankita Kumari Vishwakarma Institute of Technology, Pune, Maharashtra, India

Mayuri Kundu School of CSE, REVA University, Bangalore, India

Jonah Mubuuke Kyagaba Department of Computer Science, Makerere University, Kampala, Uganda

Sagar Lahade Pimpri Chinchwad College of Engineering, Pune, India

M. Lasya Department of IT, VR Siddhartha Engineering College, Vijayawada, India

Prabhat Mali Department of Electronics and Computer Engineering, Pulchowk Campus, Institute of Engineering, Tribhuvan University, Lalitpur, Bagmati, Nepal

A. Manasa Department of ECE, G. Narayanamma Institute of Technology and Science (for women), Hyderabad, Telangana, India

K. Manivannan Department of Computer Science and Engineering, Faculty of Engineering and Technology, Jain (Deemed-to-be University), Bengaluru, Karnataka, India

Ggaliwango Marvin Department of Computer Science, Makerere University, Kampala, Uganda

H. Massar Laboratory of Electrical and Industrial Engineering, Information Processing, Informatics, and Logistics (GEITIIL), Faculty of Science Ain Chock, University Hassan II—Casablanca, Casablanca, Morocco;
Research Center STIS, M2CS, National School of Arts and Crafts of Rabat (ENSAM), Mohammed V University in Rabat, Rabat, Morocco

James Mayanja Makerere University, Kampala, Uganda

A. Mehathab Data Scientist with Specialization in Remote Sensing, Expertz Lab Technologies, Kochi, Kerala, India

Pranay Meshram G.H.Raisoni College of Engineering Nagpur, Nagpur, Maharastra, India

M. Miyara Computer Science and Systems Laboratory (LIS), Faculty of Science Ain Chock, University Hassan II—Casablanca, Casablanca, Morocco

Moustapha Mohamed Saleck LAROSERI Laboratory, Department of Computer Sciences, Faculty of Sciences, University of Chouaib Doukkali, El Jadida, Morocco

Monish Mohanty Department of Computer Science and Engineering, Amrita School of Computing Bengaluru, Amrita Vishwa Vidyapeetham, Bengaluru, India

Himanka Sekhar Mondal ICT and Servics Group, Centre For Development of Advanced Computing, Kolkata, India

Vijaya B. Musande Department of Computer Science and Engineering, Jawaharlal Nehru Engineering College, Aurangabad, Maharashtra, India

Joshua Mwesigwa Makerere University, Kampala, Uganda

B. Nsiri Research Center STIS, M2CS, National Graduate School of Arts and Crafts of Rabat (ENSAM), Mohammed V University in Rabat, Rabat, Morocco;
Group of Biomedical Engineering and Pharmaceuticals Sciences, National Graduate School of Arts and Crafts of Rabat (ENSAM), Mohammed V University, Rabat, Morocco

Nagi Ould Taleb Computer Sciences Department, Higher Institute of Digital, Nouakchott, Mauritania

Jaderick P. Pabico Institute of Computer Science, University of the Philippines Los Baños, College, Laguna, Philippines

Barnali Pal ICT and Servics Group, Centre For Development of Advanced Computing, Kolkata, India

Lakshmi Harika Palivela School of Computer Science Engineering, Vellore Institute of Technology, Chennai, India

Sanjeeb Prasad Panday Department of Electronics and Computer Engineering, Pulchowk Campus, Institute of Engineering, Tribhuvan University, Lalitpur, Bagmati, Nepal

Kinjal Patel Faculty of Computer Applications and Information Technology, Gujarat Law Society University, Ahmedabad, Gujarat, India

Peeta Basa Pati Department of Computer Science and Engineering, Amrita School of Computing Bengaluru, Amrita Vishwa Vidyapeetham, Bengaluru, India

Babasaheb R. Patil VIMEET, University of Mumbai, Mumbai, Maharashtra, India

Prathamesh Patil JSPM's Rajarshi Shahu College of Engineering, Pune, Maharashtra, India

Dipak Chandra Patranabis Department of Instrumentation and Electronics, Jadavpur University, Kolkata, India

Kaveri Pawar JSPM's Rajarshi Shahu College of Engineering, Pune, Maharashtra, India

Prathibha Prakash Department of Computer Science and Engineering, Amrita School of Computing, Amrita Vishwa Vidyapeetham, Amritapuri, India

V. Premanand School of Computer Science Engineering, Vellore Institute of Technology, Chennai, India

R. Radha EEE, Study World College of Engineering, Coimbatore, India

S. Rajkumar Sona College of Technology, Salem, India

K. Ravi SR University, Warangal, Telangana, India

Md. Redwan Ahmed Department of Computer Science and Engineering, East West University, Dhaka, Bangladesh

S. M. Renuka Devi Department of ECE, G. Narayanamma Institute of Technology and Science (for women), Hyderabad, Telangana, India

Ahmed Wasif Reza Department of Computer Science and Engineering, East West University, Dhaka, Bangladesh

Sharmin Akther Rima Department of Computer Science and Engineering, East West University, Dhaka, Bangladesh

Aleksey Rogachev Volgograd, Russian Federation

Dmitry Rogachev VNIIGiM named after Kostyakov, Moscow, Russian Federation

A. Ruhan Bevi Department of Electronics and Communication Engineering, SRM Institute of Science and Technology, Kattankulathur, Chengalpattu, Tamil Nadu, India

Md. Sajid Anam Ifti Department of Computer Science and Engineering, East West University, Dhaka, Bangladesh

P. Sajitha Karunya Institute of Technology and Sciences, Coimbatore, India

Priyanka Sakpal JSPM's Rajarshi Shahu College of Engineering, Pune, Maharashtra, India

Tanmayee Samantaray Neural Engineering Laboratory, Department of Biosciences and Bioengineering, Indian Institute of Technology Guwahati, Guwahati, Assam, India

Rushikesh Sanap Department of Computer Engineering and IT, COEP Technological University, Pune, India

Ashwini Sanjay Gaikwad Department of Computer Science and Engineering, Jawaharlal Nehru Engineering College, Aurangabad, Maharashtra, India

Bhakti Sarag Department of Computer Engineering, Vishwakarma Institute of Technology, Pune, India

P. Saraswathi IT, Velammal College of Engineering and Technology, Madurai, India

S. Satish Department of Electronics and Communication Engineering, Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, Chennai, Tamil Nadu, India

Rajesh Saturi Department of CSE, Vignana Bharathi Institute of Technology, Ghatkesar, Hyderabad, India

R. Senthil Rama Department of Electrical and Electronics Engineering, DMI College of Engineering, Nagercoil, Tamil Nadu, India

S. K. Shabeer Department of IT, VR Siddhartha Engineering College, Vijayawada, India

Milind Shah Department of Computer Science and Engineering, Krishna School of Emerging Technology and Applied Research (KSET), Drs. Kiran and Pallavi Patel Global University (KPGU), Vadodara, Gujarat, India

Aman Shakya Department of Electronics and Computer Engineering, Pulchowk Campus, Institute of Engineering, Tribhuvan University, Lalitpur, Bagmati, Nepal

Sahil Shendurkar Department of Computer Engineering, Vishwakarma Institute of Technology, Pune, India

Swati Shinde Pimpri Chinchwad College of Engineering, Pune, India

Aditya Shreeram Neural Engineering Laboratory, Department of Biosciences and Bioengineering, Indian Institute of Technology Guwahati, Guwahati, Assam, India

Jay B. Simha REVA Academy for Corporate Excellence (RACE) REVA University, Bangalore, India

A. Soulaymani Laboratory Health and Biology, Faculty of Sciences, Ibn Tofail University, Kenitra, Morocco

S. S. Sridhar Department of Computing Technologies, College of Engineering and Technology, SRM Institute of Science and Technology, Chengalpattu, Tamil Nadu, India

G. Srikanth Reddy Department of CSE, Vignana Bharathi Institute of Technology, Ghatkesar, Hyderabad, India

D. Srinivas Goud Department of CSE, Vignana Bharathi Institute of Technology, Ghatkesar, Hyderabad, India

Konumuri Kalyan Suhas Department of Information Technology, Velagapudi Ramakrishna Siddhartha Engineering College, Vijayawada, India

S. S. Suni Ilahia College of Engineering and Technology, Ernakulam, India

S. Suresh CSE, KPR Institute of Engineering and Technology, Coimbatore, India

Mandava Venkata Sri Sai Surya Department of Information Technology, Velagapudi Ramakrishna Siddhartha Engineering College, Vijayawada, India

Muddayya Swami Department of Computer Engineering and IT, COEP Technological University, Pune, India

Mohammad Tahir G.H.Raisoni College of Engineering Nagpur, Nagpur, Maharastra, India

Gom Taye Department of Computer Science and Engineering, Rajiv Gandhi University, Itanagar, Arunachal Pradesh, India

Pius Tumwebaze Makerere University, Kampala, Uganda

Adriana Margarita Turriate-Guzman Universidad Privada del Norte, Lima, Peru

Radhesyam Vaddi Department of IT, VR Siddhartha Engineering College, Vijayawada, India

Chandrasekar Venkatachalam Department of Computer Science and Engineering, Faculty of Engineering and Technology, Jain (Deemed-to-be University), Bengaluru, Karnataka, India

Shanmugavalli Venkatachalam Department of Computer Science and Engineering, KSR College of Engineering, Tiruchengode, Tamil Nadu, India

N. Venkatesh School of CS & AI, SR University, Warangal Urban, Telangana, India

M. Venkateshwara Rao Department of CSE, Vignana Bharathi Institute of Technology, Ghatkesar, Hyderabad, India

Rachit Verma Indian Institute of Information Technology Kottayam, Kottayam District, Kerala, India

Sundeep Vijayaraghavan Department of Plastic and Reconstructive Surgery, Amrita School of Medicine, Kochi, India

Sofía Alessandra Villar-Quispe Universidad Privada del Norte, Lima, Peru

S. Vishal Department of Computer Science and Engineering, Amrita School of Computing Bengaluru, Amrita Vishwa Vidyapeetham, Bengaluru, India

Shubham Waghule Department of Computer Engineering, Vishwakarma Institute of Technology, Pune, India

Medha Wyawahare Vishwakarma Institute of Technology, Pune, Maharashtra, India;
Department of Electronics and Telecommunication Engineering, Vishwakarma Institute of Technology, Pune, India

Gökçe Nur Yılmaz Computer Engineering, Ted University, Ankara, Turkey

Meftahul Zannat Department of Computer Science and Engineering, East West University, Dhaka, Bangladesh

Modern Challenges and Limitations in Medical Science Using Capsule Networks: A Comprehensive Review



Milind Shah, Nikunj Bhavsar, Kinjal Patel, Kinjal Gautam, and Mayur Chauhan

Abstract Capsule networks (CapsNet), an emerging neural network architecture, is now used in medical science to develop potential tools and applications. Particularly, in the domain of medical image analysis, CapsNet outperforms the existing CNN models in terms of disease detection and classification tasks, such as identifying abnormalities in retinal images for diabetic retinopathy and tumor detection. Moreover, capsule networks are now used in analyzing the electronic health records (EHRs) such as hospital readmissions and mortality rates. However, the implementation of capsule networks in medical science is still in the nascent stage facing several challenges due to the limited availability of high-quality medical data, lack of interpretability, and ethical considerations. In order to overcome these challenges, more research and collaboration works should be encouraged between medical professionals and artificial intelligence (AI) experts. This research study discusses about the modern challenges faced by medical science and how the challenges can be solved by using capsule networks and algorithms.

M. Shah (✉)
Department of Computer Science and Engineering, Krishna School of Emerging Technology and Applied Research (KSET), Drs. Kiran and Pallavi Patel Global University (KPGU), Vadodara, Gujarat, India
e-mail: milindshahcomputer@gmail.com

N. Bhavsar
Department of Computer Engineering, School of Engineering, P P Savani University, Kosamba, Surat, Gujarat, India

K. Patel
Faculty of Computer Applications and Information Technology, Gujarat Law Society University, Ahmedabad, Gujarat, India

K. Gautam
Department of Computer Science and Engineering, School of Computer Science Engineering and Technology (SoCSET), ITM (SLS) Baroda University, Vadodara, Gujarat, India

M. Chauhan
Department of Computer Engineering, Gujarat Power Engineering and Research Institute (GPERI), Mehsana, Gujarat, India

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2023

S. Shakya et al. (eds.), *Fourth International Conference on Image Processing and Capsule Networks*, Lecture Notes in Networks and Systems 798,
https://doi.org/10.1007/978-981-99-7093-3_1

Keywords Neural network · Capsule networks · Machine learning · Medical science · Healthcare · Disease diagnosis · Imaging analysis

1 Introduction

Recently, medical science is facing significant challenges in diagnosing and treating complex diseases. One of the most promising solutions to solve the emering disease detection and classification challenges is the use of capsule networks. Capsule network revolutionizes the process of medical data analysis by enabling a more accurate and efficient data processing and interpretation techniques to analyze complex datasets.

Even though the research and development in the domain of capsule networks is still in nascent stage, it is now being used in medical research and patient care. One of the main advantages of capsule networks is its ability to handle the complex data structures such as 3D images or sequences of data. This makes them well suited for medical imaging applications, where accurate and detailed information of a medical image is required for the purpose of diagnosis and treatment planning [1, 2].

Another potential application of capsule networks in medical science is drug discovery. Capsule networks can be used to analyze the large datasets of molecular structures and predict how different elements will interact with each other. This helps to identify the required drugs more quickly and accurately and leverage new treatments [3].

The purpose of initiating medical research using capsule networks is to analyze its potential in addressing the emerging data processing challenges and enhancing healthcare outcomes. Medical science research intends to advance the application of these models in various areas of health care by improving healthcare diagnostics, treatment methods, patient care, and overall health outcomes [2].

Figure 1 represents some popular machine learning techniques used in medical science research. Each technique plays a unique role in addressing various medical challenges and responsibilities.

These techniques together provide an efficient tool for addressing a variety of medical science challenges, such as diagnosis, prediction, patient stratification, and data analysis.

This paper is divided into the following subsections: Sect. 1 is an introduction that discusses the modern challenges in medical science and how they can be addressed using capsule networks. Section 2 describes the modern challenges in medical science. Section 3 explains how medical science problems can be solved using capsule networks. Section 4 describes how to analyze a large amount of data in drug development using capsule networks. Section 5 includes research questions, which outline the topics discussed in this paper. Section 6 focuses on related work. Section 7 discusses the limitations of existing work, Sect. 8 describes the methodology, and Sect. 9 explains the research motivation. Section 10 presents the problem statement. Section 11 consists of the discussion. Section 12 highlights the existing

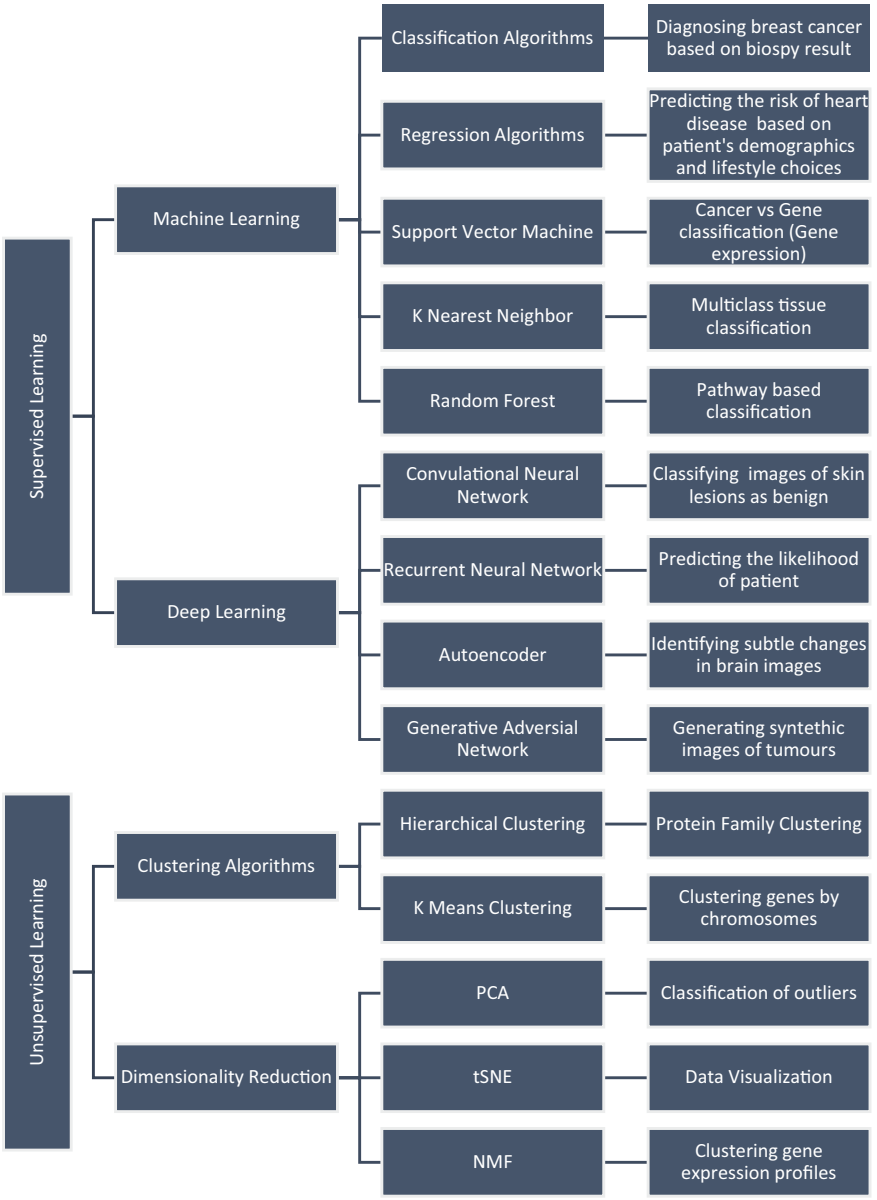


Fig. 1 Various machine learning algorithms used for medical science research

research limitations. Section 13 outlines the identified research gaps. Section 14 discusses the limitations of capsule networks for Medical Science Research, Sect. 15 focuses on the current application of Medical Science, Sect. 16 describes the open challenges and future directions, and finally, Sect. 17 provides the conclusion and future work, which concludes the paper.

2 What Are the Modern Challenges in Medical Science?

Capsule networks, a relatively new type of neural network architecture, have the potential to address several challenges in medical science (Fig. 2).

(1) Medical image classification: With the ability to capture the spatial relationships between different parts of an image, capsule networks are now used to classify the medical images with higher accuracy when compared to the traditional convolutional neural networks (CNNs).
(2) Disease diagnosis: Capsule networks can be used to diagnose diseases by considering a combination of symptoms and medical test results. Here, capsules are used to represent different symptoms and medical test results and further the

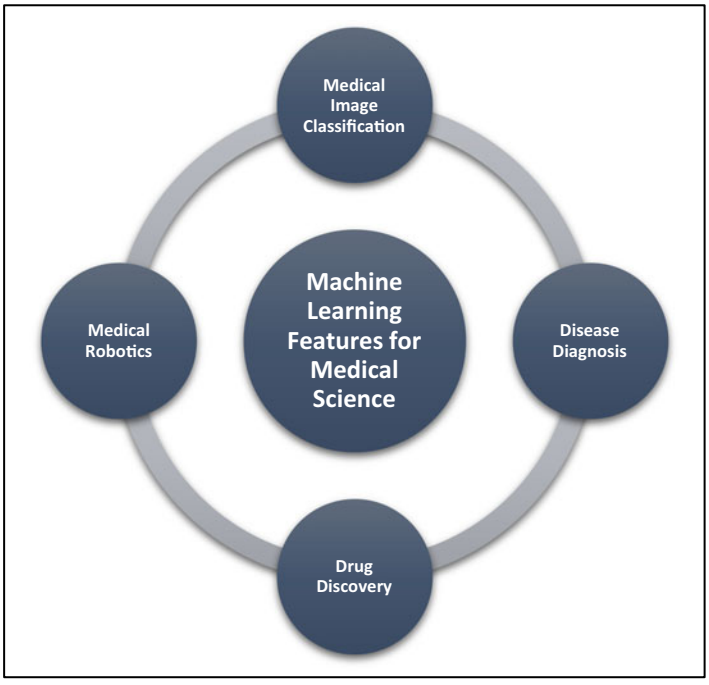


Fig. 2 Modern challenges in medical science

network can capture the complex relationships between them and make more accurate diagnosis.

(3) Drug discovery: Capsule networks can be used to predict the effectiveness of drugs by analyzing the molecular structure of human body. Capsules can be used to represent different chemical properties, and the network helps to learn about the different combinations of properties to find out the effective drug.

(4) Medical robotics: Capsule networks can be used to improve the performance of medical robots by allowing them to better understand and interpret their environment. Capsules can be used to represent different objects present in the environment.

Overall, the capsule networks have the potential to revolutionize many areas of medical science to perform more accurate and efficient analysis of complex data [2–6].

3 How Medical Science Problems Can Be Solved Using Capsule Networks?

Capsule network, a type of neural network, is proposed as a potential solution to overcome some of the limitations faced by traditional convolutional neural networks (CNNs) in image recognition tasks as mentioned below:

(1) Improved medical image recognition: One potential application of capsule networks in medical science is to improve the accuracy of medical image recognition tasks. Capsule networks are designed to be better able to recognize complex patterns in images and to handle variability in image orientation and scale, which are common challenges in medical imaging. This could potentially lead to more accurate diagnosis based on medical images.

(2) Better disease diagnosis and treatment: Capsule networks can be used to identify various medical data patterns that might not be manually diagnosed. For example, capsule networks could be used to analyze medical records and identify patients at risk for a specific disease based on data patterns in their medical history.

(3) Improved drug development: Capsule networks could also be used to analyze large amounts of data on drug interactions and side effects, which may potentially lead to the development of more safe and effective drugs.

(4) Personalized medicine: Finally, capsule networks could be used to develop personalized treatment plans based on a patient's specific medical history and genetic profile. By analyzing large amount of data obtained from patients with similar conditions, capsule networks could identify the most effective treatment strategies for individual patients.

While capsule networks are still an emerging technology in the medical field, there are several ways in which they could be potentially used to solve the emerging problems and improve patient outcomes [2–4, 7–9].

4 How to Analyze a Large Amount of Data in Drug Development Using Capsule Networks?

Capsule networks require a number of important phases to analyze large amounts of data in the process of drug development. Here is an overview of the procedure:

(1) Begin by acquiring and preprocessing the data related to drug development. This can include information from clinical trials, structures of molecules, expression of genes data, patient records, and other sources. Cleanup the data, account for missing values, standardize attributes, and perform any required transformations.
(2) Dataset partitioning: Separate the dataset into training, validation, and testing subsets. To avoid introducing bias, ensure that the partitioning preserves the distribution and characteristics of the data.
(3) Capsule network architecture design: Based on the specific objectives of the drug development task, select a suitable capsule network architecture. Consider variables such as the problem's complexity, the availability of computational resources, and your prior knowledge of the dataset.
(4) Training the capsule network: Use the training dataset to train the capsule network. This requires forward propagation, the computation of capsule activations, and iterative adjustments to the network's parameters using techniques such as gradient descent or backpropagation through time.
(5) Performance evaluation: Using the validation dataset, evaluate the performance of the trained capsule network. Evaluate parameters such as precision, recall, $F1$ score, and the area under the receiver operating characteristic curve (AUC-ROC). Adjust hyperparameters as needed to optimize the efficiency of the model.
(6) Testing and generalization: Finally, evaluate the trained capsule network on an independent testing dataset in order to determine its generalization capability. Validate the model's reliability and efficiency in drug development responsibilities by evaluating its performance on unidentified information.
(7) Interpretability and visualization: Through capsule routing and concept learning, capsule networks offer interpretability advantages. Utilize techniques such as capsule visualization, attention mechanisms, and concept activation mapping to obtain insight into the model's decision-making process and to identify significant data features or patterns.
(8) Iterative refinement: Repeat the processes mentioned above in order to refine and enhance the capsule network model. Experiment with various architectures, hyperparameters, and training techniques to improve the efficiency and robustness of the model.

(9) Consider how the capsule network analysis works within the larger drug development workflow [2, 3, 10].

5 Research Questions

This review paper discusses the following research questions.

Q1: What are the recent challenges in medical science using capsule neural network?
Q2: How medical science problems can be solved using capsule networks?
Q3: Which capsule neural network algorithms are used to solve challenges of medical science?
Q4: How to analyze large amount of data in drug development using capsule networks?
Q5: What are the existing research limitations in medical science using capsule networks?

Through the discussion of these research questions, this research aims to bridge the gap between capsule networks and medical science, clearing the way for effective incorporation into healthcare practices and contributing to improvements in diagnostics, treatment planning, and patient care.

6 Related Work

Till now, a lot of researches has been done to solve challenges of medical science using capsule neural network.

In Quan et al. [7], new coronavirus infections control deep learning analysis of chest X-rays (CXRs). Convolutional neural and capsule networks solve this challenge. DenseCapsNet, a novel deep learning framework, uses the strengths of dense convolutional networks (DenseNet) and capsule neural networks (CapsNet) to reduce data reliance in convolutional neural networks. 750 CXR images of the lungs of healthy people, patients with various diseases, and new coronavirus patients can detect COVID-19 with 90.7% accuracy, 90.9% $F1$ score, and 96% sensitivity. DenseCapsNet can detect novel coronaviruses in CXR radiography. This research uses widely analyzed COVID-19 datasets. 9432 CXR images were obtained in three COVID-19 datasets. Normal, COVID-19, and other pneumonia were classified by COVID-Net. The radiologist sent four CXRs. Radiologists use a computer to randomly choose 200 chest X-rays for diagnosis after getting typical patient lesion characteristics from a public dataset. Radiology specialists agree that the public dataset classifications and pneumonia patient features are typical, justifying its research use. To test our neural network architecture, a computer randomly picked

750 CXR images from the dataset—250 normal, 250 pneumonia, and 250 new coronavirus pneumonia. Randomly split 750 CXR images into training, validation, and testing sets. DenseNet121 extracts features and CapsNet packages capsules. Reduce neural network data reliance. 250 normal chest radiographs, 250 pneumonia radiographs, and 250 COVID-19 radiographs showed good diagnostic findings with 90.7% accuracy and 95%sensitivity.

In Modi et al. [4], a detail-oriented capsule network classifies COVID-19 in this research. In this research, the detailed oriented capsule network classifies chest CT scan images to predict COVID-19 and non-COVID-19 (DOCN). The model evaluation uses accuracy, precision, and sensitivity. The method offers 98% accuracy, 81% sensitivity, and 98.4% precision. This research used Soares et al. and Italian datasets. This collection contains 360 COVID-19 images and 397 CT scans of both sick and healthy individuals. Italy contains a total of one hundred COVID-19 CT images. There is a possibility that the first model pre-training can improve COVID-diagnostic CAPS. It was decided to create and use datasets based on CT scans. Because Image Net's natural images are distinct from the COVID-19 scan dataset, it is not used for the pre-training phase of the process. A previously trained model may be improved using COVID-CAPS to provide more similar X-ray images. First, the entire COVID-CAPS model must be trained using data from external sources. Outer classes determines the capsule amount. Two capsules take the place of the models deep network so that it can accommodate the COVID-19 data. Before beginning weight training, all capsules were easily switched out, and the usual layers were attached in place. TensorFlow, Scikit learn, and a few other similar packages are used in Python 3 for the work proposed. Evaluations are made on the accuracy, specificity, and sensitivity of the model. The research used capsule networks with many convolution layers to outperform earlier models. We will optimize model processing speed.

In Wirawan et al. [11], this research addressed EEG-based emotion identification issues using the continuous capsule network technique. The effective architecture of the first, second, third, and fourth continuous convolution layers with values of 64, 256, and 64, respectively, with kernel sizes of $2 \times 2 \times 4$, $2 \times 2 \times 64$, and $2 \times 2 \times 128$ for the first three layers and $1 \times 1 \times 256$ for the fourth layer achieved this. Layers 1–3 have 64 and 64 values, respectively. Extracting and representing differential entropy and three-dimensional cube characteristics improved the continuous capsule network. These algorithms extracted low-frequency and spatial information from electroencephalogram data. Four sentiments, two arousal classes, and valence are 91.35, 93.67, and 92.82% accurate with DEAP. The proposed approaches have DREAMER dataset accuracies of 94.23, 96.66, and 96.05% for four sentiments, two arousal classes, and valence. The AMIGOS dataset has four sentiment categories, two arousal classes, and two valence classes. The proposed algorithms have accuracies of 96.20, 97.96, and 97.32%. The 3D cube technique and DE methodology for feature extraction enable the continuous capsule network in the classification phase. DE and 3D cube are used to describe low-frequency and spatial data. The continuous

capsule network detects four and two emotions more accurately than some previously published methods on three separate datasets. Recent research addresses more parameters than continuous capsules.

In Sanchez et al. [1], explore capsule networks with ConvNets in medical image analysis datasets with minimal labeled data and class imbalance. Our research uses public MNIST, Fashion-MNIST, and medical (histological and retina imaging) datasets. Our method is suitable for medical imaging because capsule networks can be trained with minimal input and resist uneven class distribution. LeNet, baseline ConvNet, and CapsNet were tested for limited training data, class imbalance, and data augmentation usefulness. Three designs, nine data conditions, four repetitions, and four public datasets taught 432 networks. MNIST and Fashion-MNIST, with ten classes, sixty thousand training images, and ten thousand test images, are the first two datasets. CapsNet consistently beats LeNet and baseline. With additional data, the MNIST gap reduces. For DIARETDB1, LeNet with 5% of the data performs similarly to CapsNet and better than baseline with 1%. Structures in this image cause this behavior. All research except TUPAC16 had significance tests with p-values below 0.05.

In Jaber et al. [2], this paper provides an overview of several popular deep learning algorithms, describing their architectural designs and practical implementations. The algorithms discussed include backpropagation, autoencoders, variational autoencoders, restricted Boltzmann machines, deep belief networks, convolutional neural networks, recurrent neural networks, generative adversarial networks, CapsNet, the transformer, embeddings from language models, bidirectional encoder representations from transformers, and absorption. These techniques are widely utilized in the field of artificial intelligence. This research addresses several deep learning issues. Adaptive deep learning, neural architecture search, and AutoML-Zero are these challenges. Examine these algorithms' benefits and drawbacks, healthcare usage, and future. This research defines and inspires new usage of popular algorithms. This paper introduces new deep learning researchers to the benefits, drawbacks applications, and processes for many algorithms. We describe how to utilize many deep learning algorithms in health care, including the COVID-19 pandemic. We discuss deep learning challenges in one part to increase awareness and provide solutions. This may motivate researchers.

In Saif et al. [12], since 2019, COVID-19 has affected the world in unforeseen ways. COVID-19 is extremely dangerous and propagates quickly, killing millions despite our attempts at fighting it. Controlling this deadly disease or sickness requires quick and effective therapy. This research proposes a parallel concatenated convolutional block-based capsule network for COVID-19 diagnosis from multimodal medical images. By concurrently modifying the receptive field, deep convolutional blocks with varying filter sizes can incorporate discriminative spatial information and increase model scalability. By supplying fine-to-coarse information, capsule layers enable the model to represent more complex information. The model was tested on two chest radiograph datasets and one ultrasound imaging dataset. The proposed architecture's impressive COVID-19 detection test results show the models potential. Our deep convolutional capsule network has a deep parallel convolutional neural

block. Hence, a larger dataset improves our proposed network performance. COVID-19 has no such medical dataset. US COVID-19 numbers are scarce. We pre-train and fine-tune. Transfer learning involves training a deep neural network using a large source dataset before freezing its early layers. The target dataset retains parts of the networks deeper layers. Because a large dataset can be combined, this pre-training strategy works well for numerous computer vision issues.

In Zhao et al. [8], manual diagnosis of medical images is efficient, and long-term eye exhaustion affects diagnostic precision. To help in the processing and analysis of medical images, a completely automated system is necessary. Machine learning can quickly differentiate COVID-19 from other pneumonia-related diseases or healthy individuals. Nevertheless, the minimal number of labeled images and repetition of the models and data bias the learning outcomes, resulting in an inaccurate additional diagnosis. A deep channel-attention correlative capsule network for spatial, correlative, and fused feature categorization is used to solve these problems and resolve the related concerns. Experiments were confirmed using X-ray and CT imaging datasets, and they outperformed several prior findings that were considered as state-of-the-art research. Using two different datasets, we analyzed the DCACorrCapsNet model for COVID-19 recognition. The computer was equipped with an i5-7300HQ central processing unit, a GTX1050Ti graphics processing unit, and 8 gigabytes of random access memory (RAM). In this section, we will cover the settings of the multifeature extractor, the multi-level capsule, and the classification layer module. The deep architecture is responsible for the generation of correlative data from the nearby capsules, the extraction of attention-based spatial characteristics, and the integration of time- and frequency-based features as a Fisher vector. The feature representation may be obtained by merging the outputs. The DCACorrCapsNet algorithm was validated using X-ray and CT datasets from the medical field. In simulations based on real-world data, the DCACorrCapsNet model performed far better than its competitors.

In Zhang et al. [10], a capsule network model for EMR categorization can extract advanced Chinese medical text features. This model uses LSTM, GRU, and novel routing architecture. This model outperformed baseline techniques on the Chinese electronic medical record dataset with an $F1$ score of 73.51%. It outperformed several baseline models. CHIP2019 task-3 utilized China Clinical Trials Registry screening criteria. Hierarchical clustering and human induction identified seven individuals, 44 semantic categories, descriptive data, and labeling criteria for each category. Capsule + LSTM's average $F1$ is 73.51%. CNN beat GRU and LSTM due to shorter data. GRU outperformed LSTM by 4.5%. The experimental text is shorter than the control text, and therefore, both networks may have poor gating unit output. Longer experimental texts may improve LSTM and GRU network performance and narrow the gap. The capsule network increases $F1$ values by 5.46 and 10.92% in both models when integrated with the LSTM and GRU networks to classify Chinese medical text.

In Zhang et al. [5] presented an improved capsule network for medical image classification. The proposed capsule network adds feature fragmentation and multi-scale feature extraction to the original capsule network. The feature decomposition

reduces computations and improves network convergence by extracting richer information. The multiscale feature extraction unit transmits crucial data from low- to high-level capsules. Implementing capsule networks in PCam. Experimental findings show that it can classify medical images well, which might be a model for future image classification applications. Datasets were used to analyze the proposed network. We outperformed the capsule network on MNIST and CIFAR-10. We then used the proposed network to PCam medical imaging data with good results. Deep learning datasets are like MNIST. NIST collects this data. 10,000 testing samples and 60,000 training samples. Alex Krizhevsky and Ilya Sutskever created CIFAR-10 to find universal things. 10 RGB color image categories. 500,000 training images and 10,000 test images are 32×32 pixels.

In Afriyie et al. [9] provided a variant of the capsule network that is less complicated but trustworthy and can extract features for the improved classification. For challenging image datasets such as CIFAR 10, Fashion-MNIST, and kvasir-dataset-v2, the proposed model obtains test accuracies of 87.3%, 93.84%, and 93.84%, and 85.50%, respectively. For datasets containing few parameters, the proposed model achieves results comparable to those achieved using state-of-the-art approaches. Squash and amplifier approaches are effectively used in dynamic routing methodology. Our capsule maps onto the primary capsule and provides amplified features. It also contains three convolutional layers and four amplified layers. In order to identify and categorize kvasir-v2, our architecture magnifies photos to a size of 28×28. Customized layers with 3×3 kernels generate 28×28 feature maps from photos. Conv1 uses modified layer 1's 96, 26×26 kernels and Rely of 1. Model experimental results are analyzed using performance evaluation methods. We investigate the model accuracy and validation. Our technique-produced results comparable to the baseline model for all test datasets. The proposed PC layer has all necessary classification criteria. This strategy also helps explain and comprehend CapsNets for practical usage in key applications.

In Wang et al. [6], the research demonstrates two contributions of its own. First, in contrast to standard surveys that separate literature on deep learning on medical image segmentation directly into many categories and introduce literature according to a multi-level structure from rough to precise. Second, this research concentrates on supervised and weakly supervised learning techniques and on unsupervised approaches because they have been covered in many older surveys and are now controversial. Regarding supervised learning techniques, we analyze three elements of the literature: the selection of backbone networks, the design of network blocks, and the enhancement of loss functions. Separately, we analyze the literature on poorly supervised learning algorithms according to data augmentation, transfer learning, and interactive segmentation. It has been shown that completely automated deep neural network-based segmentation of medical images is extremely helpful. By analyzing the development of deep learning in the segmentation of medical images, we have highlighted possible challenges. Researchers successfully utilized a range of techniques to increase the segmentation accuracy of medical images. Accuracy improvement by itself can explain algorithm performance, especially in the field of

medical image analysis where class imbalance, noise interference, and the severe implications of incomplete tests must be considered as well.

In Akay et al. [3] describe the attempt to apply deep learning to analyze molecular shuttle swarm behavior, highlighting machine learning's advantages and demands in both established and emerging domains. To understand complex events and their functions in the production, maintenance, and management of nano- to macro-scale systems, more data must be collected and processed. There has been progress, but systems lacking the vast quantities of data required by DL algorithms require further development. Complexity of data necessitates additional procedures to validate that the appropriate DL approaches are used and to fully understand how the algorithms arrived at their conclusions. DL algorithms may be utilized to analyze the dynamics of molecular shuttle swarming. We estimate ML's future impact.

In Yu et al. [13], we develop 36,360 question combinations that fulfill Chinese medical questions. This corpus is randomly divided into three sets: 32,360 question pairs for training, 2,000 for development and 2000 for testing. We tested the proposed CapsTM on this corpus and compared it with state-of-the-art methods. CapsTM scores 0.8666. Research shows that CapsTM matches Chinese medical inquiries better than other cutting-edge technologies. Two medical experts interpret 36,360 Chinese medical question pairs. The dataset has 12,610 positive training samples, 19,750 negative training samples, 801 positive development samples, 1199 negative development samples, 798 positive test samples, and 1202 negative test samples.

In Heidarian et al. [14], convolutional neural networks are used in most deep learning-based COVID-19 pneumonia detection algorithms (CNNs). CNNs require large datasets and data augmentation to detect image samples with exact spatial correlations. Earlier CT scan methods used a simple threshold method to extend predictions made at the slice level to those made at the patient level or complex infection segmentation to identify disease. COVID-FACT, a two-stage automated CT-based approach, detects COVID-19-positive patients. Capsule networks gather COVID-FACT spatial data. In the first step, layers with infection are identified, and in the second phase, patients are classified as COVID or non-COVID. COVID-FACT identifies layers with infection and positive cases by using an in-house CT scan dataset consisting of COVID-19, community-acquired pneumonia, and normal people. COVID-FACT requires less supervision and annotation than its rivals, yet it has an accuracy of 90.82%, sensitivity of 94.5%, and specificity of 86.0%, as well as an area under the curve (AUC) of 0.98. From April 2018 to May 2020, 171 COVID-19 patients, 60 CAP patients, and 76 healthy people underwent volumetric chest CT scans for COVID-CTMD. 50–16, 183 males, 124 women. Free dataset with annotations from Figshare. Data analysis COVID-FACT. COVID-19s and 43 non-COVID cases—19 community-acquired pneumonia and 24 normal—were tested. Adam optimizer, 1e 4 initial learning rate, 16 batch size, 100 iterations. The test set model had the lowest validation loss. COVID-FACT had 90.82% accuracy, 94.5% sensitivity, 86.0% specificity, and 0.97 AUC. In another experiment, we trained our model on CT scans without lung segmentation. The model had 0.95 AUC, 90.82 accuracy, 92.72 sensitivity, and 88.37 specificity.

In Tang et al. [15], CapsNet, introduced recently, allows for a broad spatial hierarchy between features and improves linear translation robustness. This research analyzes whole-slide pathological images using CapSurv and survival loss. VGG16 semantic-level features separate discriminative patches from entire slide diseased images for optimal CapSurv training. A public cancer dataset predicts glioblastoma and lung squamous cell carcinoma survival. There is a possibility that CapSurv will improve survival model predictions. Comparisons are made between CapSurv and seven other sample survival models, including DeepConvSurv, Cox, LASSO-Cox, EnCox, PCRM with logistic, exponential, Weibull distributions, and BoostCI. The objective is to enhance the effectiveness of CapSurv. The survival analysis statistic is referred to as the C-index. It conducts model regression analysis. The C-index for CapSurv's GBM and LUSC survival is the highest possible. DeepConvSurv outperformed CapSurv by 5.8% and 7.8%, respectively, in both the TCGA-GBM and TCGA-LUSC datasets. These findings have been confirmed by CapSurv capsule and survival loss. When it came to the identification of survival-related properties, CapSurv fared better than both human design and the convolutional neural network. The AUC values generated by classification tasks for CapSurv and other survival models measure how effectively the models can differentiate between long- and short-term survivors. CapSurv is able to accurately forecast survival for GBM as well as LUSC. In terms of area under the curve (AUC), CapSurv performs better in the TCGA-GBM dataset than DeepConvSurv, Cox, Lasso-Cox, En-Cox, logistic, exponential, Weibull, and BoostCI. Since capsule orientation displays a variety of instantiation factors associated with survival time and survival loss is effective for survival prediction, CapSurv is a reliable method for analyzing survival data. The survival loss presented is compared to loss functions that contain either one or two survival loss elements.

In Monday et al. [16] propose a novel COVID-19 classification. A novel neurowavelet capsule network classifies COVID-19. We use a multi-resolution discrete wavelet transform to decrease noisy and inconsistent CXR data and increase network feature extraction robustness. Subsampling diminishes spatial feature loss in the discrete wavelet transform of multi-resolution analysis, hence enhancing classification performance. We analyzed the proposed approach using COVID-19-confirmed cases and CXR images of healthy individuals. The approach has 99.6% accuracy, 99.2% sensitivity, 99.1% specificity, and 99.7% precision. Our COVID-19 screening approach is current based on experimental results. This innovative approach will assist in COVID-19 and other disease prevention. Keras on NVIDIA GTX1080 creates the model. 104 was the learning rate for a 16-batch size. Thirty Adam epochs train the model. Regularization and batch normalization used a 50% dropout to reduce overfitting (BN). The dataset has three parts—training, validation, and testing. During training, model performance is verified. To evaluate the model performance, the test dataset was used. Our technique is evaluated using accuracy (ACC), sensitivity (SEN), receiver operating characteristic (ROC), and specificity (SPE). According to all facts, our strategy outperformed deep learning algorithms.

7 Existing Work Limitations

See Table 1.

8 Methodology

Capsule networks have shown great promise in various medical science challenges, including medical image analysis, disease diagnosis, and drug discovery. There are various capsule network algorithms that have been used to solve these challenges, including.

8.1 CapsNet

CapsNet stands for dynamic routing between capsules. CapsNet is the original capsule network algorithm proposed by Geoffrey Hinton and his team in 2017. It has been used for various medical image analysis tasks, including injury classification, medical image segmentation, and disease diagnosis. CapsNet is designed to model the part-whole relationships between different features in an image, which makes it well suited for medical image analysis tasks.

There are a few ways in which CapsNet algorithm solves medical science challenges:

1. ***Improved Feature Extraction***: The CapsNet algorithm extracts features via a two-step procedure involving convolutional layers and primary capsules. Typically, the CapsNet algorithm begins with one or more convolutional layers. These layers extract features from the given data. Each convolutional layer consists of a set of filters or kernels that can be learned. The filters are convolved with the input data using a sliding window operation to extract features and capture local patterns. The output of the convolutional layer is a collection of feature maps, each of which represents the activation of a particular learnt feature. After the convolutional layers, the obtained feature maps are used to construct the primary capsules. Each primary capsule represents a collection of neurons that encode the presence of a specific visual entity or component. In the case of image recognition, for instance, primary containers could represent fundamental features such as boundaries or corners. CapsNet uses capsules, which are groups of neurons that represent different features of an image. Capsules allow the network to model the hierarchical relationships between features, which makes it easier to distinguish between different types of medical images. This can improve the accuracy of tasks such as lesion classification and disease diagnosis.
2. ***Robustness to Deformations***: CapsNet is designed to be robust to deformations in an image, which is particularly useful for medical image analysis tasks. For

Table 1 Differential analysis

Author name	Publication with year	Methods/ algorithms used	Dataset used	Accuracy (%)	Technologies used	Limitations
Quan et al. [7]	Elsevier, 2021	DenseCapsNet	NLMMC	90.7	Python and neural network	We will also look into lightweight models, trainable parameters, and model training efficiency
Sanchez [1]	Springer, 2018	LeNet, baseline ConvNet and CapsNet	MNIST and fashion—MNIST	–	Python	This branch's completely linked layers regularize parameter optimization but lose a lot of information. To better understand the learning latent space, we will deconvolve these layers
Saif [12]	IEEE, 2021	CapsConvNet	Point of care ultrasound (POCUS)	100	Python	–
Zhang et al. [10]	MDPI, 2022	Bi-directional LSTM	CHIP2019		Python	We want to improve medical classification network models by expanding the dataset
Zhang et al. [5]	IEEE, 2020		PatchCamelyon (PCam)	88.23	Python	We want to analyze deeper capsule networks and novel routing methods to increase network performance and apply to medical image datasets

example, a lesion in a medical image may appear in different orientations or sizes, but CapsNet can still identify the lesion based on its distinctive features.
3. ***Efficient Routing***: Utilizing dynamic routing allows the CapsNet algorithm to accomplish efficient routing. Dynamic routing enables capsules in various layers to effectively communicate and coordinate their outputs, resulting in enhanced performance and robustness. CapsNet uses a dynamic routing algorithm to calculate the agreement between different capsules in the network. This allows the network to focus on relevant features and ignore irrelevant ones, which can improve the efficiency of tasks such as medical image segmentation.
4. ***Flexibility***: CapsNet is a flexible algorithm that can be adapted to different medical image analysis tasks. For example, researchers have proposed different variations of CapsNet for tasks such as breast cancer diagnosis, brain tumor segmentation, and diabetic retinopathy classification.

Overall, CapsNet is a promising algorithm for medical image analysis tasks, and ongoing research is expected to further improve its performance and applicability to various medical science challenges [1, 7, 14, 15].

8.2 ConvCaps

ConvCaps, which is short for convolutional capsule networks, is a variant of the capsule network (CapsNet) algorithm that incorporates convolutional layers before the capsule layers. ConvCaps is a variant of the CapsNet algorithm that incorporates convolutional layers before the capsule layers. It has been used for medical image analysis tasks such as brain tumor segmentation and classification of breast cancer. It has been shown to achieve state-of-the-art results in several benchmark datasets for image classification and object detection.

There are some ways in which the ConvCaps algorithm can solve medical science challenges:

1. ***Improved Feature Extraction***: ConvCaps' feature extraction process: ConvCaps uses convolutional capsules instead of layers. These capsules collect visual entity and instantiation parameters in input data. ConvCaps starts with main convolutional capsules as feature extractors. Each primary convolutional capsule receives feature mappings from the previous layer. Convolutional filters are used to create these feature maps. ConvCaps explicitly captures spatial correlations in capsules, enabling the network to learn and modify complicated patterns. Convolutional layers extract local features, whereas capsules capture higher-level correlations and agreement. ConvCaps uses convolutional layers to extract features from medical images, which allows it to capture important patterns and structures that are useful for medical image analysis tasks. This can improve the accuracy of tasks such as lesion detection, segmentation, and classification.
2. ***Efficient Routing***: The convolutional capsule (ConvCaps) algorithm achieves efficient routing by utilizing routing-by-agreement and iterative agreement

updates. ConvCaps is a capsule network (CapsNet) extension that integrates convolutional layers. Similar to CapsNet, ConvCaps uses a dynamic routing algorithm to calculate the agreement between different capsules in the network. This allows the network to focus on relevant features and ignore irrelevant ones, which can improve the efficiency of tasks such as medical image segmentation.
3. ***Robustness to Deformations***: ConvCaps is designed to be robust to deformations in an image, which is particularly useful for medical image analysis tasks. Medical images often contain distortions, noise, and artefacts, but ConvCaps can still identify and extract useful features from the images.
4. ***Transfer Learning***: ConvCaps can be pre-trained on large datasets of natural images and then fine-tuned on smaller medical image datasets. This transfer learning approach can help improve the performance of ConvCaps on medical image analysis tasks, even when the medical image datasets are small.

Overall, ConvCaps is a promising algorithm for medical image analysis tasks, and ongoing research is expected to further improve its performance and applicability to various medical science challenges [17].

9 Research Motivation

Capsule networks have recently shown the potential to address some of the challenges faced by the traditional convolutional neural networks (CNNs) to perform image recognition tasks. In the medical field, image recognition plays a crucial role in the areas of disease diagnosis and treatment planning. There are several key motivations to incorporate capsule networks for solving various medical science challenges, they are as follows:

1. ***Improved accuracy***: While performing image recognition tasks, capsule networks can improve the accuracy when compared to traditional CNNs, particularly in situations where a significant variation can be observed in the general appearance of objects (e.g., medical images of tumors or organs).
2. ***Robustness to image variability***: Medical images can often vary in appearance due to the differences in imaging techniques, patient anatomy, and disease progression. Capsule networks have the ability to recognize objects regardless of their position, rotation, or scale, and this ability is particularly useful in medical imaging applications.
3. ***Explainability***: One of the limitations of traditional CNNs is that it is difficult to interpret how the network arrived at a particular decision. Capsule networks explicitly represent the hierarchical structures present within an image, which could potentially make it easier to understand how the network arrived at its decision. This could be particularly useful in medical applications, where it is important to understand the reasoning behind a diagnosis.

4. ***Few-shot learning***: To solve the medical science challenges, it is often difficult to obtain large amounts of labeled data for training machine learning models. Capsule networks have the ability to learn from a small number of examples, which could be particularly useful in medical applications, where the number of available labeled images may be limited.

The use of capsule networks to solve medical science challenges has the potential to improve accuracy, robustness, explainability, and the ability to learn from limited data [1, 2, 4, 7–9, 12].

10 Problem Statement

Despite the vast opportunities of capsule networks in the field of medical science, there are some challenges that prevent their use in solving the issues related to health care. The limited availability of labeled medical data is one of these challenges, as it is the requirement for enhancing interpretability and explainability of capsule network predictions, robustness to variability, and noisy data.

11 Discussion

In [7], a scientific methodology was utilized to collect data, wherein the radiologist was requested to choose four chest X-ray (CXR) images and engage in a comprehensive analysis of the case characteristics. The size of the hilum in both lungs was relatively small, exhibiting an absence of any apparent nodules or protuberances. The dimensions and morphology of the cardiac silhouette exhibited characteristic features. Each of the two diaphragms located along the coast exhibited a planar surface as well as an acute angle. This phenomenon is not different from the standard. The enlargement of the lower left lobe of the lung refers to the process in which the lung tissue in this specific anatomy region becomes more compact and more solid due to various pathological conditions or infections. The establishment of a diagnosis for bacterial influenza holds significant importance. Within the lung region, an increased representation of a fragmented arrangement becomes apparent. The MRI scans revealed the presence of extensive, dense opacities distributed uniformly across the entirety of both lung fields. The testing for viral pneumonia holds significant importance. Radiologists employ a computer-based random selection process to obtain a sample of 200 chest X-ray images for diagnostic purposes. This selection is made following the provision of lesion characteristics associated with typical patients, as documented in a publicly available dataset. This facilitates a more precise determination of the condition. Consensus among radiology professionals exists regarding the reliability of the public dataset's classifications and the shared clinical characteristics observed in various types of pneumonia. These findings highlight

the dataset's potential utility for research purposes. To validate our neural network system, a computer randomly selected 750 chest X-ray (CXR) images from the dataset mentioned earlier. Out of these, 250 images represented normal cases, 250 represented other types of pneumonia, and 250 represented cases of pneumonia caused by the new coronavirus. Arbitrary differentiations were established among the 750 CXR images that constituted each set for training, validation, and testing.

In [4], we considered model pre-training for COVID-CAPS diagnosis. CT scan datasets were created and utilized. Image Net is not used for pre-training because its images are natural images and not COVID-19 scans. COVID-CAPS should enhance a pre-trained model for similar X-ray images. Initially, external data must train the full COVID-CAPS model. Outside classes determine the capsule number. Two capsules replace the final layer to modify the model for COVID-19 data. Before weight training, all capsules were transferred effortlessly. The capsule network dataset is label-compatible. COVID-19 is limited to severely imbalanced datasets because it is new. Due to this imbalanced dataset, we increased the loss margin for misclassified positive instances. Pre-trained capsule networks provide the limited dataset. A 5-class external dataset pre-trains five final capsules. The compression of five capsules into two capsules ultimately results in the entire COVID-19 dataset, which contains all capsule layers. In the case of Adam optimizers, we used a learning rate that consisted of 10, 3, 100 epochs, and 16 different batch sizes. For the purposes of model training (80%) and model selection (20%), respectively, we have two sets of data: training sets and validation sets. Evaluation will perform the test of the model using the test set. Accuracy, sensitivity, and specificity are the three measurements used.

In [1], we experimentally show that CapsNets' equivariance features minimize significant data needs, making them suitable for medical image processing. We handle class distribution imbalance and limited annotated data using computer-aided diagnostic (classification) tasks. We run several controlled tests on two visual (MNIST and Fashion-MNIST) and two medical datasets that target mitotic identification (TUPAC16) and diabetic retinopathy detection to verify our claims (DIARETDB1). This is the first study to use capsule networks to solve medical image analysis challenges. We focus on computer-aided diagnostic image classification challenges. We compare capsule networks (CapsNets) to conventional convolutional networks (ConvNets) under biomedical image database constraints such as class imbalance and limited labeled data. CapsNets create hierarchical image representations by passing images through several layers of the network, like ConvNets. Against the trend toward deeper models, the original CapsNet has only two layers: a main caps layer that captures low-level signals and a specialized secondary caps layer that can predict both the existence and position of an object in the image. We propose using CapsNets equivariance features to use structural redundancy in images and minimize the training image number. Image patterns may indicate a good diagnosis. Exudates and hemorrhages might be soft or solid. CapsNet routing algorithms learn to establish relations between features, unlike ConvNets, which use feature detection to make decisions. The routing method collects redundant features instead of repeating them

throughout the network to handle invariance. The above advantages directly impact the quantity of data samples needed to train networks.

In [12], an ablation researcher outlined each stage in the proposed process. Preprocessing and pre-training considerably increase the performance of the proposed method. The major preprocessing step, NCLAHE, improves the image resolution by 1.6–2.13%. Technical issues can reduce image quality, which can harm machine learning performance. N-CLAHE improves medical imaging quality and performance. When large-scale labeled data can be aggregated, pre-training improves performance. Pre-training the deep feature extractor block of the proposed technique, which takes a lot of training data, improves its performance. Our technique decreases false positives and false negatives more than current methods. COVID-19 and pneumonia patients share symptoms, making misclassification likely. Our approach handles these scenarios effectively. Most notably, our approach detects COVID-19 patients better from chest X-rays. Six of 1144 COVID-19 chest X-ray patients are misclassified, and only two are pneumonia. Our proposed approach performs well, properly identifying 228 COVID-19 instances and misclassifying only two, one of which is pneumonia. Our proposed approach correctly categorized 645 COVID-19 US frames and misclassified nine, including eight pneumonia frames. AUCs of receiver operating characteristics (ROC) are shown to demonstrate the proposed model's ability to differentiate classes.

In [10], capsule-based Chinese medical text categorization is proposed. Chinese medical text classification enhances the capsule network model in this research. We collected complex medical text characteristics using the capsule network's unique network topology and powerful feature extraction. When the capsule network and the long short-term memory (LSTMs) network handle medical text, $F1$ values are at least 4% higher than other basic models. Capsule network processes. Five methods were tested for screening clear text datasets in clinical trials. Researchers calculated the mean accuracy, recall, and $F1$ scores for 44 categories. According to the research, a shorter experimental text may have decreased the gating unit output in both networks. Longer experimental texts may increase LSTM and GRU network performance, decreasing the result gap. Integrating the capsule network with the LSTM and GRU networks increases $F1$ values by 5.46% and 10.92%, respectively. This shows that Chinese medical literature categorization depends on the capsule network. The research showed that GRU and LSTM with the capsule network model can categorize medical short texts. The accepted information will help categorize medical literature in future research and for the book "Electronics 2022, 11, 2229." 10 of 11 students can create medical knowledge graphs. This research's dataset is small, restricting the model's impact.

12 Existing Research Limitations

Some researchers have utilized capsule networks for lung nodule classification and observed that this approach outperformed traditional neural networks. Nevertheless, the authors acknowledged that the limited size and lack of diversity in the dataset could potentially restrict the generalizability of the findings. Due to the often restricted size of medical datasets, acquiring a sufficient amount of labeled data to effectively train capsule networks can be challenging. This limitation obstructs the generalizability of the results and may necessitate the application of transfer learning or other techniques to address the data shortage [2, 4, 7, 8, 14].

Some articles have summarized the current research on capsule networks in medical image analysis and discussed their potential advantages and limitations. The authors noted that the interpretability of capsule networks could pose a challenge and that there is still a lack of research on their use in many specific medical applications. Capsule networks utilize a complex set of hierarchical representations to make predictions, which can make interpreting the output difficult. This challenge hinders understanding the factors driving the predictions and can limit the ability to diagnose or treat medical conditions effectively. While some studies have investigated the use of capsule networks in specific medical applications, such as medical imaging or diagnosis, there is still a lack of research in many other areas. This limitation restricts the generalizability of the results and necessitates further research to comprehend the potential of capsule networks in various medical contexts [1, 3, 5, 15–17].

Some researchers have also utilized capsule networks for the classification of diabetic retinopathy and observed that this approach outperformed traditional neural networks. However, the authors acknowledged that the limited size of the dataset and the requirement for substantial computing resources could pose challenges. Training capsule networks can be computationally expensive, particularly when working with extensive medical datasets. This limitation can inhibit the utilization of capsule networks in real-time or resource-constrained environments, such as low-resource settings or mobile applications [1, 3].

13 Identified Research Gaps

Labeled data limitations make capsule networks for medical research challenging. Finding large datasets of labeled medical images for capsule network training may be difficult and expensive. Overfitting reduces capsule network performance on unseen data [8].

Computational complexity is another issue with capsule networks in medicine. Convolutional neural networks are computationally simpler than capsule networks. This makes capsule networks challenging to train on large datasets and inappropriate for real-time applications [10, 14].

14 Limitations of the Capsule Networks for Medical Science Research

Capsule networks are another kind of deep learning model that has been proven effective for a variety of tasks, including the classification of medical images. However, capsule networks have some limitations that must be addressed when using them for medical science research.

- Capsule networks have the drawback of requiring an extensive amount of training data. This is because capsule networks learn to precisely represent the spatial relationships between image features, which requires a large quantity of training data. In the field of medical science, it can be challenging to acquire huge amounts of labeled training data, which can restrict the application of capsule networks for certain tasks.
- Data noise can affect capsule networks. Because capsule networks learn to express spatial correlations between features, data noise can break these associations. Imaging artefacts and image acquisition variances cause noise in medical data. Noise inhibits capsule network learning.
- Finally, training capsule networks is computationally costly. Capsule networks contain many parameters to learn, making training tedious and expensive.
- Capsule networks can have difficulty generalize.
- Limited availability of labeled data.
- Capsule networks are difficult to read, making predictions difficult to understand.
- Capsule networks are less popular than other deep learning models, and hence, there is less research regarding how to use them.

Despite these drawbacks, capsule networks are promising for medical research. Capsule networks may become increasingly more powerful for medical science research as research continues to address these limitations [2].

15 Current Applications

Machine learning and deep learning have been widely utilized in both established and emerging domains. The following is a summary of areas that have utilized the current capabilities of machine learning (ML) and deep learning (DL).

15.1 Micro-Robot Adaptation

Certain researchers utilized a sophisticated convolutional neural network (CNN) architecture to enable the real-time adaptation of a minimally-invasive and therapeutic endoscopy robot specifically designed for the gastrointestinal (GI) system.

The methodology that was previously described enabled the robot to determine its spatial location and angular orientation in relation to a specified frame of reference. The combination of the CNN architecture with sophisticated sensor platforms has enabled the realization of precise and instantaneous robot adaptation within the gastrointestinal (GI) system. The integration of molecular robot technologies with CNN (and other) architectures presents an exciting opportunity for advancing research. This combination has the potential to facilitate accurate diagnosis and interventions, as well as enable the incorporation of molecular shuttles within living organisms [3].

15.2 Network Biology

The area of network biology focuses on analyzing the interactions among biomolecules that play a role in the organization and operation of living cells. This includes the analysis and reconstruction of extensive biological networks within cells, as well as the development and implementation of smaller synthetic gene networks. Machine learning algorithms have the capability to utilize and integrate diverse biological datasets in order to construct complex and layered models for analyzing the interactions of different phenomena. These analyses vary from within the cells interactions to intercellular interactions within tissues and organs, encompassing both healthy and diseased conditions. Significant advancements have been made in this field; however, additional research is required to comprehensively understand the mechanisms by which affecting biological process networks can give rise to various diseases. Capsule networks utilize methodologies related to convolutional neural networks (CNNs) in order to visually represent distinct and targeted objects toward specialized capsules or modules. A capsule network incorporates additional layers within each hidden layer, as opposed to simply increasing the number of layers [3].

16 Open Challenges and Future Redirections

Researchers effectively utilized a variety of techniques to improve the segmentation accuracy of medical images. Although accuracy improvement only cannot clarify algorithm performance, especially in the field of medical image analysis where class imbalance, noise interference, and the severe consequences of failed tests must be taken into consideration. Below sections will discuss possible future research directions for medical image segmentation.

16.1 Transfer Learning

Medical imaging is commonly accompanied by considerable interference. Furthermore, the process of highlighting medical images is frequently more costly in comparison with highlighting natural images. Hence, the exploration of utilizing pre-trained deep learning models on natural images for medical image segmentation represents an interesting field for future research. Furthermore, transfer learning plays a significant role in achieving weakly supervised medical image segmentation. Transfer learning refers to the utilization of pre-existing knowledge to acquire new knowledge, with a particular focus on identifying similarities between the existing and new knowledge domains. Transfer learning is a technique that leverages the correlation between data or tasks to facilitate the sharing of model parameters or knowledge acquired by a model. This sharing process enhances the efficiency of model learning by accelerating the learning process. Transfer learning is an effective way to address the issue of limited availability of labeled data [6].

17 Conclusion and Future Work

Capsule networks have shown great potential for addressing various challenges in medical science. These challenges include but are not limited to medical image analysis, disease diagnosis, drug discovery, and personalized medicine. From medical image analysis to personalized medicine, capsule networks have demonstrated improved performance over traditional convolutional neural networks.

In this review paper, we have highlighted what are the modern challenges in medical science, how medical science problems can be solved using capsule networks and the capsule network algorithms used to solve any medical science problems.

While capsule networks show great promise, there is still much work to be done in this field. One area of future research could focus on improving the interpretability of capsule networks, as understanding how these networks arrive at their decisions is crucial for building trust with medical professionals and patients. Furthermore, the efficiency of capsule networks can be improved by exploring novel applications and training methods that leverage their unique capabilities in medical science.

Despite these advantages, there are still several challenges that need to be addressed in order to fully realize the potential of capsule networks in medical science. One of the main challenges is the need for large amounts of labeled training data, which can be difficult to obtain in medical applications. Another challenge is the interpretability of the learned features, which is important for understanding the underlying mechanisms of disease and building trust in the predictions made by the model.

References

1. Jiménez-Sánchez A, Albarqouni S, Mateus D (2018) Capsule networks against medical imaging data challenges. Lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics), 11043 LNCS, pp 150–160. Available at https://doi.org/10.1007/978-3-030-01364-6_17
2. Abdel-Jaber H et al (2022) A review of deep learning algorithms and their applications in healthcare. Algorithms 15(2). Available at https://doi.org/10.3390/a15020071
3. Akay A, Hess H (2019) Deep learning: current and emerging applications in medicine and technology. IEEE J Biomed Health Inform 23(3):906–920. Available at https://doi.org/10.1109/JBHI.2019.2894713
4. Modi S et al (2021) Detail-oriented capsule network for classification of CT scan images performing the detection of COVID-19. Mater Today Proc [preprint]. Available at https://doi.org/10.1016/j.matpr.2021.07.367
5. Zhang Z et al (2020) Enhanced capsule network for medical image classification. In: Proceedings of the annual international conference of the IEEE engineering in medicine and biology society, EMBS, July 2020, pp 1544–1547. Available at https://doi.org/10.1109/EMBC44109.2020.9175815
6. Wang R et al (2022) Medical image segmentation using deep learning: a survey. IET Image Process 16(5):1243–1267. Available at https://doi.org/10.1049/ipr2.12419
7. Quan H et al (2021) DenseCapsNet: detection of COVID-19 from X-ray images using a capsule neural network. Comput Biol Med 133:104399. Available at https://doi.org/10.1016/j.compbiomed.2021.104399
8. Zhao A et al (2023) DCACorrCapsNet: a deep channel-attention correlative capsule network for COVID-19 detection based on multi-source medical images. IET Image Process 17(4): 988–1000. Available at https://doi.org/10.1049/ipr2.12690
9. Afriyie Y, Weyori BA, Opoku AA (2021) Exploring optimised capsule network on complex images for medical diagnosis. In: IEEE international conference on adaptive science and technology, ICAST, Nov 2021 (Jan 2022). Available at https://doi.org/10.1109/ICAST52759.2021.9682081
10. Chen W et al (2022) Research on medical text classification based on BioBERT-GRU-attention. In: 2022 IEEE international conference on advances in electrical engineering and computer applications, AEECA 2022, pp 213–219. Available at https://doi.org/10.1109/AEECA55500.2022.9919061
11. Wirawan IMA et al (2023) Continuous capsule network method for improving electroencephalogram-based emotion recognition. Emerg Sci J 7(1):116–134. Available at https://doi.org/10.28991/ESJ-2023-07-01-09
12. Saif AFM et al (2021) CapsCovNet: a modified capsule network to diagnose COVID-19 from multimodal medical imaging. IEEE Trans Artif Intell 2(6): 608–617. Available at https://doi.org/10.1109/TAI.2021.3104791
13. Yu X et al (2021) CapsTM: capsule network for Chinese medical text matching. BMC Med Inform Decision Making 21(2):1–9. Available at https://doi.org/10.1186/s12911-021-01442-9
14. Heidarian S et al (2021) COVID-FACT: a fully-automated capsule network-based framework for identification of COVID-19 cases from chest CT scans. Front Artif Intell 4:1–13. Available at https://doi.org/10.3389/frai.2021.598932
15. Tang B et al. (2019) CapSurv: capsule network for survival analysis with whole slide pathological images. IEEE Access 7:26022–26030. Available at https://doi.org/10.1109/ACCESS.2019.2901049
16. Monday HN et al (2022) COVID-19 pneumonia classification based on NeuroWavelet capsule network. Healthcare 10(3):1–18. Available at https://doi.org/10.3390/healthcare10030422
17. Stynes P. Brain age classification from brain MRI using ConvCaps framework Animesh Kumar National College of Ireland Supervisor

Studies on Movie Soundtracks Over the Last Five Years



Sofía Alessandra Villar-Quispe and Adriana Margarita Turriate-Guzman

Abstract In the field of film studies and, especially, music studies, the number of those who dare to go beyond the limits when researching film production is minimal. This is because the soundtrack as a subject has not always been an object of analysis that is investigated as rigorously as it is now. Therefore, this paper seeks to determine the trend of studies on movie soundtracks over the last five years in the Scopus database. This is a systematic review using the PRISMA method. Fifteen articles published between 2017 and 2022 were selected, and, once all the information was reviewed, it was divided into four categories. The main limitations were the lack of information and the number of papers written in English. Finally, it was concluded that the soundtrack fulfills several functions within the narrative of a movie and that the topic needs more academic attention in Ibero-America. Therefore, researchers are invited to continue to problematize this issue in future systematic reviews.

Keywords Soundtrack · Movies · Audiovisual · Film music

1 Introduction

In the field of film studies and, especially, music studies, the number of those who dare to go beyond the limits when researching film production is minimal. This is because the soundtrack as a subject has not always been an object of analysis that is investigated so rigorously [1]. Nowadays, in the research bases, there is diverse information and studies on movie soundtracks. The creation of the soundtrack has a very significant influence on the audiovisual narrative of movies. For example, Serrano [2] indicates that the minimalist music composed by Philip Glass for the soundtrack of the documentary *Koyaanisqatsi* manages to better communicate the

S. A. Villar-Quispe · A. M. Turriate-Guzman (✉)
Universidad Privada del Norte, Lima, Peru
e-mail: adriana.turriate@upn.edu.pe

S. A. Villar-Quispe
e-mail: N00170591@upn.pe

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2023

S. Shakya et al. (eds.), *Fourth International Conference on Image Processing and Capsule Networks*, Lecture Notes in Networks and Systems 798,
https://doi.org/10.1007/978-981-99-7093-3_2

projection of the reality in which we live by expressing what is inexpressible about our current state in the movie. Likewise, when associating certain sounds with something specific, such as a city, Grifol-Isely [3] argues that the mnemonics of a soundtrack are essential along with the hues, as this is how Barcelona managed to adopt a brand image. Meanwhile, González [4] states that Daniel Montorio's music for the Spanish Western highlights more the actions than the emotions since he uses musical blocks for them and not for the characters.

Regarding the creative process of the soundtrack for a movie, Fernandez [5] comments that, during the creation of *Carmen, la de Triana*, models of musical quality and coherence were found, such as the balance between cultural, folkloric, and the most popular elements to achieve the distinction from the German version of the movie. In turn, Oliveira [6] states that to achieve this new aesthetics supporting the future of cinema, it is necessary to provoke emotions by adapting the nature of the movie, as evidenced in the movies by Flora Gomes. Furthermore, in the movie *Das Kabinett des Doktor Caligari*, Amorós and Gómez [7] point out that, to provide the movie with greater depth and dynamism, Rainer Viertlböck turned away from classical music, plunged into jazz, and only with that genre created continuous leitmotifs to identify the characters and their actions.

On the other hand, music can also be linked to political communication. This happened in the late 1920s when sound was incorporated into cinema. Instrumentation and rhythm were prioritized as technical resources to represent Latin America through music in movies, and this, in turn, contributed as a precedent for later productions aligned with U.S. foreign policy [8]. Similarly, Farías [9] points out that the character and other timbral features of musical pieces, such as the *Nueva Canción Chilena*, used in the Chilean documentary during the Popular Unity, allowed the audience to connect with the ideas of the Left.

In other studies, varying from movie promotion to education, medicine, psychology, and computer science, the soundtrack serves as a useful tool. For the first category, music videos with movie soundtracks can fulfill their role thanks to commercial communication [10]. In the field of education, Porta and Herrera [11] determined that the soundtrack gives meaning and sense to both the narrative and the multimedia experience of children watching their favorite audiovisuals. Finally, in medicine and psychology, the research aims to obtain results focused on perception, emotions, bodily reactions, and moral judgments. Meanwhile, in computer science, movie genres are investigated according to their soundtrack [12–15]. Therefore, the topic addressed in this research is current and relevant. Therefore, the aim is to determine the trend of studies on movie soundtracks over the last five years in the Scopus database. For this purpose, this systematic review will be carried out, and the PRISMA method will be used. The methodology section is presented below.

2 Methodology

This research addresses the trend of studies on movie soundtracks over the last five years in the Scopus database. It was developed through a systematic review, a study in which information from published articles is managed. The researcher collects all those articles of his/her interest regarding a specific topic, analyzes them, compares them, and, in the end, draws his/her conclusions. This must be developed objectively, either from a qualitative or quantitative perspective [16].

The research was conducted using keywords such as "music AND films", "soundtrack AND movies", and "soundtrack AND films". There were only three searches performed in Scopus, a database of citations and abstracts from neutral sources with more than 1.8 billion referenced citations. This was the only database used due to its strict quality and ethical criteria for the selection of titles, which are selected by the independent Scopus Content Selection and Advisory Board (CSAB), an international group of scientists and researchers with experience in journal publishing, representing the main scientific disciplines [17].

The formulae provided by the system after applying the filters were as follows:

[17] TITLE-ABS-KEY (music AND films) AND (LIMIT-TO (OA, "all")) AND (LIMIT-TO (PUBYEAR, 2022) OR LIMIT-TO (PUBYEAR, 2021) OR LIMIT-TO (PUBYEAR, 2020) OR LIMIT-TO (PUBYEAR, 2019) OR LIMIT-TO (PUBYEAR, 2018) OR LIMIT-TO (PUBYEAR, 2017)) AND (LIMIT-TO (DOCTYPE, "ar")) AND (LIMIT-TO (LANGUAGE, "Spanish")) AND (LIMIT-TO (SRCTYPE, "j")) [17].

[17] TITLE-ABS-KEY (soundtrack AND movies) AND (LIMIT-TO (OA, "all")) AND (LIMIT-TO (PUBYEAR, 2022) OR LIMIT-TO (PUBYEAR, 2021) OR LIMIT-TO (PUBYEAR, 2020) OR LIMIT-TO (PUBYEAR, 2019) OR LIMIT-TO (PUBYEAR, 2018) OR LIMIT-TO (PUBYEAR, 2017)) AND (LIMIT-TO (DOCTYPE, "ar")) AND (LIMIT-TO (LANGUAGE, "English") OR LIMIT-TO (LANGUAGE, "Spanish")) AND (LIMIT-TO (SRCTYPE, "j")) [17].

[17] TITLE-ABS-KEY (soundtrack AND films) AND (LIMIT-TO (OA, "all")) AND (LIMIT-TO (PUBYEAR, 2022) OR LIMIT-TO (PUBYEAR, 2021) OR LIMIT-TO (PUBYEAR, 2020) OR LIMIT-TO (PUBYEAR, 2019) OR LIMIT-TO (PUBYEAR, 2018) OR LIMIT-TO (PUBYEAR, 2017)) AND (LIMIT-TO (DOCTYPE, "ar")) AND (LIMIT-TO (LANGUAGE, "English") OR LIMIT-TO (LANGUAGE, "Spanish")) AND (LIMIT-TO (SRCTYPE, "j")) [17].

Scopus detected a total of 80 articles, and the PRISMA method was applied to refine the search and answer the research question [18]. The 15 articles selected were published between 2017 and 2022, amidst which 2021 stands out with 6 publications. Likewise, it is important to note that Spain has the highest number of publications (i.e., 6) out of the studies collected from the Scopus database. On the other hand, the first reason for excluding some articles, in the end, was because they were essays, and the second was because one of them was an interview. Regarding language, 9 articles were published in Spanish and 6 in English. Finally, out of the keywords of

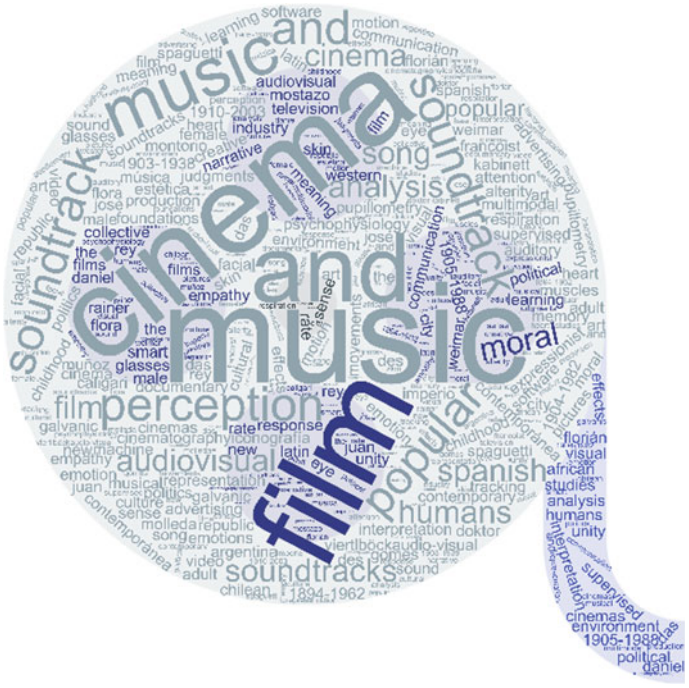


Fig. 1 Word cloud

all the articles, the most outstanding one was "music". Figure 1 illustrates the word cloud.

3 Results

After reviewing the articles, the information was divided into four categories. The list of categories is tabulated in Table 1.

The findings by category are detailed below.

3.1 Soundtrack Influence on the Audiovisual Narrative of Movies

During this research, three articles that reinforce the importance of a movie's soundtrack in determining its narrative were detected by scanning the papers and finding out that what they had in common was their authors pointing out how music, whether by choice of the director or not, need to fit the nature of the film. Firstly, the study

Table 1 List of categories

Category	Articles	Author	Synthesis
Soundtrack influence on the audiovisual narrative of movies	Aullidos de chatarra. Una aproximación a Koyaanisqatsi (1982) a través de la música de Philip Glass y su imbricación ideológica [Scrap howls. An approach to Koyaanisqatsi (1982) through the music of Philip Glass and its ideological interweaving] [2]	Serrano [2]	Minimalism in a documentary soundtrack
	Estrategias narrativas y musicales en el wéstern español de los años sesenta: dos filmes de Ramón Torrado con música de Daniel Montorio [Narrative and musical strategies in the spanish western of the 1960s: two films by Ramón Torrado with music by Daniel Montorio] [4]	Gonzáles [4]	Roles of music in the narrative of two Spanish Westerns
	Diálogos entre imágenes y sonidos en el imaginario urbano. Ritmando Barcelona(s) [Dialogues between images and sounds in the urban imaginary. Barcelona's rhythms] [3]	Grifol-Isely [3]	Importance of sound in the image of the city through audiovisuals
The creative process of a movie soundtrack	La creación musical en el proceso cinematográfico. Una perspectiva crítica [Music creation in the film process. A critical approach] [1]	Lázaro [1]	Problems with the musical creation of a movie
	"I make films to be seen": the narrative issue of Flora Gomes [6]	Oliveira [6]	Elements of narrative cinematography and the direction process
	La música de Carmen, la de Triana (1938): Las fuentes y el proceso creativo [The music of Carmen, la de Triana (1938): Its sources and its creative process] [5]	Fernández [5]	Important implications during the pre-production phase
	La música de Rainer Viertlböck en la versión restaurada del film Das Kabinett des Doktor Caligari, de Robert Wiene (1919) [Rainer Viertlböck's music in the restored version of the film Das Kabinett des Doktor Caligari by Robert Wiene (1919)] [7]	Amorós and Gómez [7]	Creative soundtrack proposal for an expressionist film
Music and political communication	La música en las representaciones de lo latinoamericano en los primeros filmes hollywoodenses con sonido incorporado (1927–1932) [Music in the Representations of Latin America in Hollywood Films of Early Sound Era (1927–1932)] [8]	Poveda [8]	Problems in the musical representation of Latin America

(continued)

Table 1 (continued)

Category	Articles	Author	Synthesis
	Música y comunicación política en el documental durante la Unidad Popular [Music and politic communication in the documentary during Chile's Popular Unity] [9]	Farías [9]	Use of music to achieve a political identity
The soundtrack as a study instrument	Soundtrack Music Videos: The Use of Music Videos as a Tool for Promoting Films [10]	Selva-Ruiz and Fénix-Pina [10]	Features of the soundtrack music video
	La música y sus significados en los audiovisuales preferidos por los niños [Music and its significance in children favourite audiovisuals] [11]	Porta and Herrera [11]	Soundtrack and children's interpretation
	A computational lens into how music characterizes genre in film [12]	Ma et al. [12]	Use of music according to the movie genre
	Locus of emotion influences psychophysiological reactions to music [14]	Merrill et al. [14]	Music listening and its psychophysiological responses
	How soundtracks shape what we see: analyzing the influence of music on visual scenes through self-assessment, eye tracking, and pupillometry [13]	Ansani et al. [13]	Soundtrack effects in the interpretation of visual scenes
	The influence of film music on moral judgments of movie scenes and felt emotions [15]	Steffens [15]	Specific emotions associated with moral judgments

conducted by Serrano [2], in 2018, on the documentary *Koyaanisqatsi*, demonstrates that the music of the whole movie performs an organizing role. In other words, it gives an internal structure to the movie. The beginning contains music that fulfills the introductory, structural, and incidental role; the climax and denouement present the musical notes more prominently. Secondly, Gonzáles [4], in 2021, through the analysis of the soundtrack of two Spanish Westerns by Daniel Montorio, demonstrated how the moral codes and some conventional narrative frameworks of the genre are strategically underlined through the conscious use of leitmotifs. For example, in Los Cuatreros, the villains are associated with the timbres of the horns and trombones, as they usually appear in the dark. Thirdly, Grifol-Isely [3], in a study on the sounds and the image-movement relationship associated with the city of Barcelona, points out how in the movie *Vicky Cristina Barcelona* by Woody Allen, when the first scene begins with the song Barcelona by Giulia y Los Tellarini, it is assigned the image of a labyrinthine place because the rhythm is fast and accentuated by the use of stringed instruments and an accelerated cadence. Thus, these three cases prove the influence of music within the audiovisual narrative of a documentary or fiction feature film.

3.2 The Creative Process of a Movie Soundtrack

Regarding the creation of music during the filmmaking process, four articles coincide. Basically, according to Lázaro [1], the movie director makes all the elements work in a strict film sense. However, to eliminate incompatibilities within the media of expression, the director, as well as the other heads of the filmmaking areas, must be a musician; consequently, the film musician must be a filmmaker. Indeed, Oliveira [6], in a study of the movies by Flora Gomes, in 2019, notes that the soundtrack is a mark and, in turn, it is highly important for the filmmaker. This occurs because music, along with other aesthetic aspects, has a great effect on the viewer: it arouses emotions. Moreover, it is also an adjustable component since it allows for adaptability in production as it can be included when shooting is finished.

Regarding the above, Fernández [5], in his analysis of the movie *Carmen, la de Triana*, in 2021, highlights the creative film process of Florián Rey, the director, and the success of the composers, Muñoz Molleda (soundtrack) and Juan Mostazo (songs). The latter composed a song in situ based on the recitation of a comedy. This, and the fact that the songs were recorded before filming, differentiated them from other productions of that time in Spain. On the other hand, Amorós and Gómez [7] indicate that Rainer Viertlböck innovated in musical composition, using synthesizers and other instruments characteristic of jazz. The result was a modern soundtrack for *Das Kabinett des Doktor Caligari*, an expressionist film from 1919 that, at the time, was not well received for resorting to the music of classical composers. In conclusion, although Lázaro's proposal is far from the current panorama, it is clear, from the examples provided by the other authors, the relevance of music within the creative process in cinema.

3.3 Music and Political Communication

Two articles coincide in the category of music and political communication. On the one hand, Poveda [8], in a study of the musical characterization of Latin America in three Hollywood productions with incorporated sound, from 1927 to 1932, noted three problems in the performances: the use of references to Spanish culture; the sexual connotations through the body, tango and rumba; and the interest in some resources and processes that are used later, during the era of the "Good Neighbor Policy". On the other hand, Farías [9], in 2020, in his analysis of the music of three documentaries produced during the Popular Unity in Chile, found that in *El Derecho al Descanso* certain modern music was used to portray the bourgeoisie and an instrumental huayno piece for the working class (only at the beginning). In contrast, in *Un Verano Feliz* and *Balnearios Populares*, rock-pop is used to symbolize enjoyment and leisure. The latter movie differs from the others because of the absence of the *Nueva Canción Chilena*, and because it was produced by a group of amateurs and not by a government institution. Undoubtedly, this demonstrates its great commitment to Popular Unity.

3.4 Soundtrack as a Study Instrument

Regarding studies in which the soundtrack is used as a study instrument, six articles were found. The first one, by Selva-Ruiz and Fénix-Pina [10], in 2021, focuses on the use of the soundtrack music video as an instrument for movie promotion. The authors analyzed the content of 119 music videos produced over 33 years. Then, according to internal and external characteristics, they determined that 93.3% included images of the artist. In the second study, from the field of education, Porta and Herrera [11] used the soundtrack to know its meaning from the children's interpretation. For this purpose, they showed different clips of 14 audiovisuals edited in 3 versions (sound, image, and everything) to 547 students between 9 and 15 years old. In the end, thanks to an assessment questionnaire, they indicated the superiority of the soundtrack in 3 out of 5 films: The Lion King, Titanic, and Oz, the fantasy world.

Regarding the previous paragraph, now from the computer science area, in the third study, Ma et al. [12] designed neural network models with various clustering mechanisms to predict the genres of 110 movies by listening to their music. With a macro-$F1$ score of 0.62, the models demonstrated higher accuracy than previous work; however, no evidence was found that music of a given genre implies common low-level visual features. Now, in the field of medicine and psychology, the fourth study, by Merrill et al. [14], in 2020, compared the effect of perceiving emotion in music with assessing the emotion induced in the listener by the music, both in psychophysiological responses such as muscle activity, respiration or heart rate, which were measured in 40 participants while listening to a soundtrack excerpt.

Thus, they found that, if more attention is paid to musical expression, bodily reactions improve.

Likewise, regarding the aforementioned, the latest studies are also within the psychology sector. The fifth one, carried out by Ansani et al. [13], in 2020, reports two studies on how music affects the interpretation of visual scenes. In the first one, conducted with an online survey, variables such as empathy toward the character or the perception of the environment were highly influenced by the soundtrack. In the second one, which was the same but added eye-tracking, the results were replicated. Finally, Steffens [15], for his research on the influence of music on perceptions, actions, and moral judgments, selected 81 participants and showed them 2 excerpts from 2 different movies with 4 different soundtrack music pieces. However, in 3 out of 4 cases, the desired emotional induction through film music was not successful. Thus, although in this last case no favorable results were evidenced, the efficiency of the soundtrack as a study instrument is evident through the other examples.

4 Discussion and Conclusions

The soundtrack used for specific purposes for the narrative of a movie has become one of the major sources of musical experience in our daily environment [19]. Currently, Porta and Herrera [11], to some extent, agree with the aforementioned, stating that the importance of the audiovisual pairing—music and media—is because it is part of children's daily life. Meanwhile, José Lázaro [1] indicates that music is infinitely expressive, so it can be used, for example, to accompany the speech of a character and convey both ideas and emotions. All this shows the relevance of the soundtrack for having stayed as an everyday thing for us.

Thus, this study has different scopes. Firstly, at the methodological level, it is identified as a systematic review [16], in which the PRISMA method was used [18]. Secondly, at a practical level, it may be useful to build the theoretical framework for future research on movie soundtracks. Finally, at a theoretical level, it updates the information available on the research topic addressed. Now, regarding the main limitations, the first one was the lack of information to specify the original research problem. Initially, it had been proposed to resort only to soundtrack studies carried out in Ibero-America; however, there were not enough studies to carry out a systematic review. Therefore, researchers in this geographic region must reflect and take an early interest in the subject. On the other hand, the second limitation was the large number of articles in English that are available in the database. Nevertheless, this provides an opportunity for those who are fluent in English to broaden their knowledge.

In summary, this research was carried out using the PRISMA method [18], which facilitated access to 15 documents from the Scopus database [17]. From this information, it was found that the highest publication trend was in the year 2021, that Spain leads in the number of publications, and that out of the keywords of all the articles, and the most outstanding one was "music" (this is depicted in Fig. 1). After a complete review of the material collected, it became evident that the theory can

be divided into four categories (for details, see Table 1). Firstly, the influence of the soundtrack on the audiovisual narrative of movies in which music fulfills several roles; secondly, the importance of music during the creative process of a movie; thirdly, the relationship between soundtrack and political communication, in which music is used to connote several elements. Finally, in the fourth category, the effectiveness of the soundtrack as a study instrument for different research projects ranging from the field of education to psychology is highlighted. However, aside from all of the above, the subject still has much to be investigated, especially in Ibero-America, where more academic attention to the subject is needed in the future. Therefore, other collaborators, colleagues, and students are invited to discuss the topic in future systematic reviews.

References

1. Lázaro López J (2021) Music creation in the film process. A critical approach. [La creación musical en el proceso cinematográfico. Una perspectiva crítica] Fonseca J Commun 22:173–188. https://doi.org/10.14201/fjc-v22-23477
2. Serrano J (2018) Scrap howls. An approach to Koyaanisqatsi (1982) through the music of Philip Glass and its ideological interweaving. [Aullidos de chatarra. Una aproximación a Koyaanisqatsi (1982) a través de la música de Philip Glass y su imbricación ideológica] Boletín De Arte 28:479–506. https://doi.org/10.24310/BOLARTE.2007.V0I28.4494
3. Grifol-Isely L (2022) Dialogues between images and sounds in the urban imaginary. Barcelona's rhythms. [Diálogos entre imágenes y sonidos en el imaginario urbano. Ritmando Barcelona(s)] Fotocinema 24:265–289. https://doi.org/10.24310/FOTOCINEMA.2022.VI24.13834
4. González C (2021) Narrative and musical strategies in the Spanish Western of the 1960s: two films by Ramón Torrado with music by Daniel Montorio. [ESTRATEGIAS NARRATIVAS Y MUSICALES EN EL WÉSTERN ESPAÑOL DE LOS AÑOS SESENTA: DOS FILMES DE RAMÓN TORRADO CON MÚSICA DE DANIEL MONTORIO] Anuario Musical 76:181–206. https://doi.org/10.3989/anuariomusical.2021.76.09
5. Fernández R (2021) The music of Carmen, la de Triana (1938): its sources and its creative process. [LA MÚSICA DE CARMEN, LA DE TRIANA (1938): LAS FUENTES Y EL PROCESO CREATIVO1] Anuario Musical 76:151–180. https://doi.org/10.3989/anuariomusical.2021.76.08
6. Oliveira J (2019) I make films to be seen: the narrative issue of Flora Gomes. J Sci Technol Arts 11(1 Special Issue):2-1–2-10. https://doi.org/10.7559/citarj.v11i1.587
7. Amorós A, Gómez N (2017) Rainer Viertlböck's music in the restored version of the film Das Kabinett des Doktor Caligari by Robert Wiene (1919) [A música de rainer viertlböck na versão restabelecida do filme das kabinett des doktor caligari de robert wiene (1919)] Cuadernos De Música, Artes Visuales y Artes Escénicas 12(2):2017. https://doi.org/10.11144/Javeriana.mavae12-2.mrvv
8. Poveda J (2020) Music in the representations of Latin America in hollywood films of early sound era (1927–1932). [La música en las representaciones de lo latinoamericano en los primeros filmes hollywoodenses con sonido incorporado (1927–1932)] Resonancias 24(46):55–77. https://doi.org/10.7764/RES.2020.46.4
9. Farías M (2021) Music and political communication in the documentary during Chile's popular unity. [MÚSICA Y COMUNICACIÓN POLÍTICA EN EL DOCUMENTAL DURANTE LA UNIDAD POPULAR] Universum 36(2):601–621. https://doi.org/10.4067/S0718-23762021000200601

10. Selva-Ruiz D, Fénix-Pina D (2021) Soundtrack music videos: the use of music videos as a tool for promoting films. Commun Soc 34(3):47–60. https://doi.org/10.15581/003.34.3.47-60
11. Porta A, Herrera L (2017) Music and its significance in children's favorite audiovisuals. Comunicar 25(52):83–91. https://doi.org/10.3916/C52-2017-08
12. Ma B, Greer T, Knox D, Narayanan S (2021) A computational lens into how music characterizes genre in film. PLoS One 16. https://doi.org/10.1371/journal.pone.0249957
13. Ansani A, Marini M, D'Errico F, Poggi I (2020) How soundtracks shape what we see: analyzing the influence of music on visual scenes through self-assessment, eye tracking, and pupillometry. Front Psychol 11. https://doi.org/10.3389/fpsyg.2020.02242
14. Merrill J, Omigie D, Wald-Fuhrmann M (2020) Locus of emotion influences psychophysiological reactions to music. PLoS One 15. https://doi.org/10.1371/journal.pone.0237641
15. Steffens J (2020) The influence of film music on moral judgments of movie scenes and felt emotions. Psychol Music 48(1):3–17. https://doi.org/10.1177/0305735618779443
16. Manterola C, Astudillo P, Arias E, Claros N (2013) Systematic reviews of the literature: what should be known about them. [Revisiones sistemáticas de la literatura. Qué se debe saber acerca de ellas] Cirugia Espanola 91(3):149–155. https://doi.org/10.1016/j.ciresp.2011.07.009
17. Elsevier (2022) Coverage you can count on. https://www.elsevier.com/solutions/scopus/how-scopus-works/content
18. Page MJ, McKenzie JE, Bossuyt PM et al (2021) The PRISMA 2020 statement: an updated guideline for reporting systematic reviews. Syst Rev 10:89. https://doi.org/10.1186/s13643-021-01626-4
19. Wingstedt J, Brändström S, Berg J (2010) Narrative music, visuals, and meaning in film. Vis Commun 9(2):193–210. https://doi.org/10.1177/1470357210369886

Blind Source Separation of EEG Signals Using Wavelet and EMD Decomposition



H. Massar, M. Miyara, B. Nsiri, and T. Belhoussine Drissi

Abstract Blind source separation (BSS) is a widely adopted approach for processing biomedical signals, particularly electroencephalography (EEG) signals, which indicate the electrical activities of the brain. Currently, EEG data is heavily polluted by various physiological artifacts such as muscular action, cardiac rate, and blinking, which call for the utilization of BSS techniques. This study intends to evaluate the performance of BSS methods by combining them with two other signal processing techniques, namely the discrete wavelet transform (DWT) and the empirical mode decomposition (EMD). This study proposes two approaches, the first of which involves combining the BSS and EMD algorithms. Here, the intrinsic mode function (IMFs) is extracted from each EEG signal and then separated those using BSS techniques. Based on the results, the ACSOBI-RO method was found to be the most effective. The second approach adds a DWT step to extract the approximation coefficients from the IMFs. The resulting coefficients are then fed into the BSS methods, and the SOBI method is found to be the most effective. In conclusion, the addition of the DWT block has significantly improved the performance of our methods. The obtained research findings demonstrate that the proposed approaches can effectively address the problem of physiological artifacts in EEG data and can be applied to other biomedical signals.

Keyword Blind source separation · Empirical mode decomposition · Discrete wavelet transform · Electroencephalogram

H. Massar (✉) · T. Belhoussine Drissi
Laboratory of Electrical and Industrial Engineering, Information Processing, Informatics, and Logistics (GEITIIL), Faculty of Science Ain Chock, University Hassan II—Casablanca, Casablanca, Morocco
e-mail: Hamzamassar10@gmail.com

M. Miyara
Computer Science and Systems Laboratory (LIS), Faculty of Science Ain Chock, University Hassan II—Casablanca, Casablanca, Morocco

H. Massar · B. Nsiri
Research Center STIS, M2CS, National School of Arts and Crafts of Rabat (ENSAM), Mohammed V University in Rabat, Rabat, Morocco

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2023

S. Shakya et al. (eds.), *Fourth International Conference on Image Processing and Capsule Networks*, Lecture Notes in Networks and Systems 798,
https://doi.org/10.1007/978-981-99-7093-3_3

1 Introduction

Blind source separation (BSS) is a set of methods that enables the identification of individual source components from the mixtures collected by numerous instruments. The sources of interest are often not readily accessible due to the sensitivity of the information to various disturbances. BSS utilizes a combination of different sources (observations) to predict the source signals, hence the term "blind" as only the mixtures are recognized as information. BSS has proven to be a solution to many previously unsolved issues in various application areas, including biomedicine, networking, and acoustics. The medical domain is one of the areas where BSS has numerous applications, such as its utilization in electroencephalography (EEG) analysis. EEG is a non-invasive method of observing and logging brain activity, which involves attaching electrodes to the forehead in order to identify the electrical impulses produced by the brain's neurons. It is frequently used to identify and explore various neurological conditions, including seizures, sleep disorders, and brain trauma. Moreover, cognitive functions such as focus, recollection, and language can be studied by using EEG data [1, 2].

Due to the great temporal accuracy of EEG, extraneous noise can cause a variety of artifacts in the data. Artifacts can be caused by measuring equipment as well as human beings, with the former comprising malfunctioning electrodes, line noise, and high electrode impedance. By employing a more accurate recording device and following stringent recording guidelines, these problems may be avoided, although physiological artifacts are more challenging to get rid of. Common physiological aberrations in the EEG data include eye movements, eye blinks, heart activity, and muscle activity [3]. Scientists widely used the BSS methods to separate the artifacts and the neural component to deal with these kinds of artifacts. Other techniques, like the wavelet transform, have been used to cope with artifacts. The wavelet transform (WT) is a technique used for processing and examining signals, enabling a localized breakdown of the spectrum in terms of either space or time. This method involves decomposing single-channel signals, such as EEG signals or other biomedical signals, aiding in their analysis and processing. An additional technique, empirical mode decomposition (EMD), has been used to analyze these data types. The empirical mode decomposition method makes the assumption that any given set of data (or signal), at any given moment, can be deconstructed into a number of coexisting, straightforward natural oscillations, referred to as intrinsic mode function (IMF), which superimpose one another [4].

Several publications have proposed and reported the processing and evaluation of EEG data using BSS, EMD, WT, or a combination of these techniques. Jamil et al. [5] utilized the discrete wavelet transform (DWT) and independent component analysis (ICA) to remove ocular artifacts as an illustration. The authors compared their approach to existing methods and found that it outperformed other techniques in terms of accuracy and reliability. They also discussed potential applications of EEG signal analysis in non-restricted environments, such as brain-computer interfaces and neurofeedback training. In a study by Zhao et al. [6], a multi-step strategy for blind

source separation (BSS) is proposed to mitigate ocular, movement, and myogenic artifacts commonly observed in mobile high-density EEG data. This approach effectively addresses various types of artifacts typically encountered in such data, leading to enhanced accuracy and dependability in the subsequent analysis and interpretation of brain activity. Chen et al. [7] introduced an alternate approach using canonical correlation analysis (CCA) in combination with multivariate EMD. Paradeshi et al. [8] employed the wavelet ICA technique to eliminate eye-related artifacts, assessing their method using metrics such as peak signal noise ratio (PSNR), standard deviation (SD), root mean square error (RMSE), and power spectral density (PSD). In another investigation, Mowla et al. [2] utilized a dual-step process involving CCA and SWT to address EMG irregularities initially, followed by the application of the SOBI algorithm and SWT to remove EOG artifacts.

The suggested approach integrates the DWT, EMD, and BSS methodologies. Following the computation of the approximation coefficient and the IMFs produced from the EEG data, this research study aims to evaluate the BSS approaches. Here, the success of the suggested strategy is evaluated using the Spearman correlation coefficient (SCC).

This research study is organized as follows; the methods, approaches, and databases used in this study are covered in Sect. 2; the findings are presented in Sect. 3; and the conclusion is presented in Sect. 4.

2 Material and Methods

2.1 Datasets

A semi-simulated dataset was used in this research study. 27 individuals' EEG data were gathered during an eyes-closed period. Here, 19 EEG sensors are inserted by using the 10–20 International Method. Further, two datasets were gathered for a total of 54 datasets totaling 30 s. The sounds are filtered by using a bandpass filter at 0.5–40 Hz before being notched and filtered at 50 Hz. The data collection occurred at a rate of 200 Hz. Also, by placing four electrodes above and below the left eye, two on each eye, and two more electrodes, they recorded the EOG signals from the same participants with their eyes open. This procedure resulted in the creation of two binary signals: the horizontal EOG (HEOG), obtained by subtracting the recordings from the left and right EOG electrodes, and the vertical EOG (VEOG), obtained by subtracting the recordings from the top and lower EOG electrodes.

The developed semi-simulated EEG dataset was produced by using the contamination model detailed below:

$$\mathrm{EEG}_c = \mathrm{EEG}_N + A \times \mathrm{VEOG} + B \times \mathrm{HEOG} \tag{1}$$

where EEG_N stands for the signals gathered during the session with the eyes closed, and EEG_c for the relevant EEG data that have been purposely polluted. The previously published VEOG and HEOG signals' contamination coefficients are defined by vectors *A* and *B*. For more details on this collection, please refer to [9].

2.2 Empirical Mode Decomposition

A potent method for nonlinear and non-stationary data processing is empirical mode decomposition (EMD) [10]. Such signals may be analyzed using its data-adaptive multi-resolution technique by dissecting them into multiple components at different resolutions. EMD divides the signal into intrinsic mode functions (IMFs), which are zero-mean signals with a range of amplitudes. EMD depicts the signal as an expansion of fundamental functions depending on the input signal and approximated by an iterative process known as sifting, in contrast to standard decomposition approaches like wavelets that analyze signals by projecting them onto a collection of specified basis vectors [11].

The decomposition technique primarily relies on the detection of extrema and a screening process. Interpolating the local minima yields a lower envelope, while interpolating the local peaks yields a higher envelope. The first component $C1$ is then derived as the difference between the original data $x(t)$ and the mean $m1$ calculated from the upper and lower envelopes [12]. By using EMD, we can analyze and interpret nonlinear and non-stationary signals more effectively than with traditional techniques. It allows us to uncover patterns and trends in data that may not be visible with other methods, making it an essential tool in fields such as finance, biology, and engineering.

$$C_1 = x(t) - m_1 \tag{2}$$

The *C* component will be an IMF. If not, new extrema may be formed, and the ones already there can be relocated or magnified. Therefore, to acquire an IMF, this screening process must be used many more once [1]. Each IMF must satisfy two requirements: that the mean value of the envelopes always be zero and that the number of zero crossings and extremes be equal or varied by one. The residue is mapped after the discovery of the first IMF.

$$r(t) = x(t) - \mathrm{IMF}(t) \tag{3}$$

With $r(t)$ representing the residue.

2.3 Wavelet Transform

The wavelet transform can analyze a signal and identify any abrupt shift in the temporal and frequency features of the signal, making it a complicated analytical tool for managing non-stationary data [3]. The wavelet transform breaks down the input signal into a collection of basic building components called ψ_(*a*, *b*). These are later produced by dilating and translating the mother wavelet (Eq. 3), a well-localized analysis function with few oscillations [13, 14].

$$\psi_{a,b} = \frac{1}{\sqrt{a}}\psi\left(\frac{t-b}{a}\right) \tag{4}$$

With b being the displacement (translation), associated with the idea of time location, and a being the step size (dilation), related to the idea of frequency (scale).

The continuous wavelet transform (CWT) for a signal $x(t)$ is defined as [15]:

$$C(a,b) = \frac{1}{\sqrt{a}}\int_{-\infty}^{+\infty} x(t)\Psi^*\left(\frac{t-b}{a}\right)\mathrm{d}t \tag{5}$$

where Ψ^* is the mother wavelet's complex conjugate.

In our study, we utilized the discrete wavelet transform (DWT) because of its rapidity and ease of simulation. By consecutively transmitting a signal through low-pass and high-pass filters, we calculated the DWTs, yielding approximation coefficients and details. The application of half-band filters reduces the signal sample's dimension by a factor of 2 with each decomposition stage [16]. Figure 2 shows the corresponding Mallat's decomposition tree (Fig. 1).

In this context, x represents the signal, h stands for the high-pass filters, g represents the low-pass filters, $d1$ and $d2$ correspond to the details at the first and second levels, while $a1$ and $a2$ denote the first and second-level approximation coefficients.

In EEG signal processing, the DWT offers several advantages. Firstly, it enables the identification and extraction of different frequency components present in the signals. EEG signals often consist of a combination of various frequency components that correspond to different brain activities. By decomposing the signal using the DWT, these components can be separated and analyzed individually. This is particularly useful in identifying specific brainwave patterns, such as alpha, beta, theta, and delta waves, which are associated with different cognitive states. Secondly, the DWT allows for efficient denoising and artifact removal from EEG signals. The DWT can help distinguish between the desired brain activity and unwanted artifacts by decomposing the signal into different frequency bands. By discarding or suppressing the wavelet coefficients corresponding to noise, the DWT enables the extraction of cleaner EEG signals for further analysis.

There are several types of wavelets; in our investigation, we used the wavelets that are shown in Table 1. These wavelet functions were selected based on their proven

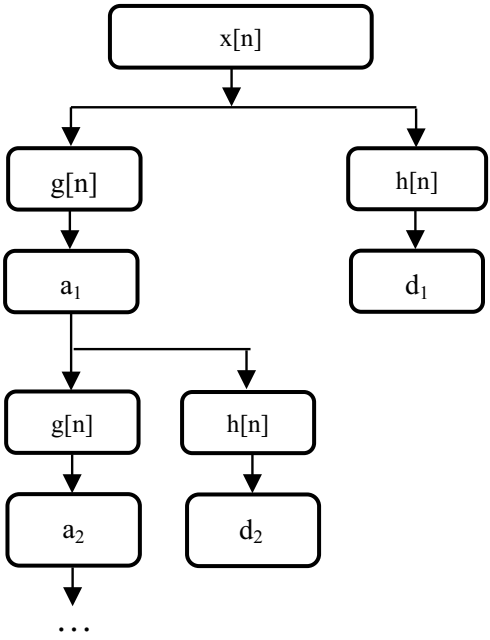


Fig. 1 DWT signal analysis [17]

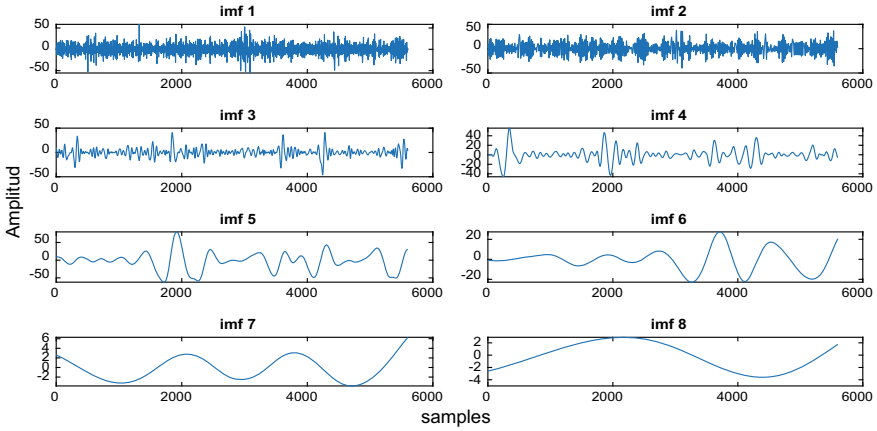


Fig. 2 IMFs component that are extracted from the EEG signals

effectiveness in similar applications and our prior research. These wavelet families possess desirable properties such as orthogonality, compact support, and good time–frequency localization, making them suitable for analyzing and processing EEG signals using the DWT. Their selection was guided by the goal of improving the accuracy and efficiency of the proposed method for EEG signal analysis.

Table 1 Wavelet types used in our research

Wavelets types	Wavelets
Daubechies	db1, db2, db3, db4, db5, db6, db7, db8, db9, db10, db11
Coiflets	coif1, coif2, coif3, coif4, coif5
Symlets	sym1, sym2, sym3, sym4, sym5, sym6, sym7, sym8, sym9, sym10, sym11

2.4 Blind Source Separation

A method called blind source separation (BSS) is used to decompose a combination of signals into its individual components. This is achieved by analyzing the statistical characteristics of the mixture, without any prior knowledge of the constituent signals or the mixing process. BSS approaches aim to identify unique statistical structures for each source signal, using statistical independence as a key characteristic. Higher-order statistics, sparsity, and non-Gaussianity are also frequently utilized. BSS can estimate the underlying sources by analyzing the mixture's statistical features. BSS is a powerful signal processing technique used in audio processing to separate mixed audio signals into individual sources, and it has many other applications, including in telecommunications, biological signal processing, and image and video processing.

$$X(P \times M) = A(S(N \times m)) \tag{6}$$

A represents the mixed model, X represents the mixed signals, and S is the source signal.

The most basic variation of the mixed model is the instant linear blend, where the observations are concurrently formed by combining the sources in a linear manner at every instance. In this case, a mixing matrix A with a size of (P, N) serves as the model for mixing.

$$X(P \times M) = A(P \times N)S(N \times m) \tag{7}$$

The BSS's objective is to identify an unmixing Matric B that can separate the input signal.

2.4.1 Independent Component Analysis

To resolve the issue of blind source separation (BSS), independent component analysis (ICA) is a highly effective approach that aims to separate blind sources [18, 19]. ICA works by discovering estimates of unknown sources through the assimilation of these sources to signals with maximal independence. Typically, this method involves two steps: whitening the mixed signals and utilizing higher-order statistics (HOS) to maximize independence.

HOS are a useful tool for achieving statistical independence between signals. By identifying patterns in the data that reflect the degree of independence between variables, HOS can help separate independent sources and reduce the impact of noise and outliers. Several approaches have been developed that use HOS, including maximum likelihood, contrast functions, and quadratic criteria. These approaches can be evaluated and compared using measures such as Kullback–Leibler divergence and mutual information. Overall, HOS provides a powerful framework for identifying and exploiting the complex statistical dependencies that exist in real-world data.

Infomax

Infomax techniques refer to learning algorithms that engage in an optimization procedure with the objective of enhancing entropy to its maximum extent. The underlying rationale is based on the concept that if the components within signal vector Y are self-contained, then the anticipation of combined entropy achieves its highest value [13]. The Infomax methodology is frequently implemented through a neural network architecture known as "feedforward network with lateral inhibition." The subsequent mathematical expression is employed for training the network [3]:

$$J(t+1) = J(t) + n(t)(I - ha(Y)Y^T)B(t) \tag{8}$$

The initial value of J is frequently a random matrix, where h is a function related to the distribution structure and $n(t)$ is a learning rate function.

2.4.2 Second-Order Statistics

Multivariate time series data can have hidden structures that can be found using second-order statistics approaches [8]. Second-order statistics refers to statistical measurements that may be derived from the covariance matrix of the observed data. These techniques are founded on the joint diagonalization of any quantity of random time-delayed covariance matrices calculated from X [20].

SOBI

The SOBI method's fundamental premise is to distinguish the sources in a combination using their temporal structure. It is predicted that there is no correlation between the sources and that each source has a unique time structure. The cross-correlation matrix of the combined signals is first calculated (Eq. 8).

$$Rx(\tau) = E\{x(t)x(t+\tau)^T\} \tag{9}$$

This matrix is then subjected to a process known as eigenvalue decomposition. This results in a collection of eigenvectors that serve as the foundation for the divided

sources. In a new coordinate system with uncorrelated sources, the combined signals are then projected using the eigenvectors.

SOBI-RO

The SOBI technique has been modified to perform better in the face of outliers or noise. This update is known as second-order blind identification with robust orthogonalization. It works by jointly diagonalizing a set of randomly selected time-delayed covariance matrices of the recorded signals. Before creating time-delayed covariance matrices for mixed signals, the following formula is a robust orthogonalization formulation:

$$\acute{x} = Ox \tag{10}$$

The next stage is to calculate the eigenvalue decomposition of the covariance matrix of the whitened data in order to determine the mixing matrix.

Amuse

The algorithm for multiple unknown signals extraction (AMUSE) is a widely used second-order statistics blind source separation (BSS) method that was introduced in 1991. The first step in the algorithm involves centering and whitening the mixed signal matrix to obtain $X_b(t)$.

$$X_b'(t) = BX \tag{11}$$

where B is the whitening matrix. Next, the time-lagged covariance matrix is decomposed into its eigenvectors and eigenvalues using an eigenvalue decomposition. Assuming V is the matrix of eigenvectors, the predicted mixing matrix J can be determined as:

$$J'(t) = B^{-1}V \tag{12}$$

This process is useful for estimating the individual sources from the mixed signals without prior knowledge of the sources [21].

2.5 Proposed Method

This study proposes two methods that combine EMD, DWT, and BSS techniques. The first method involves decomposing each EEG channel into a set of IMFs using the EMD algorithm (Fig. 2). Next, we apply the BSS method to each set of IMFs

to separate them into independent components (ICs) (Fig. 3). Finally, we use the Spearman correlation coefficient (SCC) to assess the degree of similarity between the ICs and the IMFs retrieved from the VEOG and the HEOG.

For the second method, we used the DWT to extract the approximation coefficient at a specific level from the IMF data (Fig. 4). We then applied the BSS method to these data to decompose them into ICs (Fig. 5). Similar to the first method, we used the SCC to compute the similarity between the created ICs and the approximation coefficient extracted from the HEOG's and VEOG's IMFs.

A statistical metric called the Spearman correlation coefficient evaluates the direction and strength of the relationship between two ranking variables. The value is

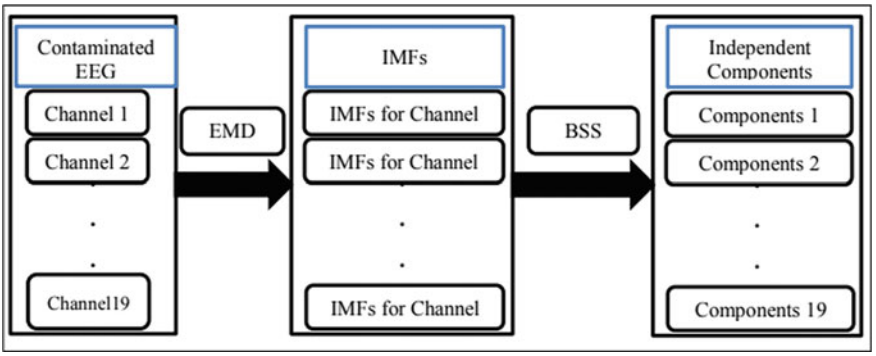


Fig. 3 First method's steps. The illustration includes two steps: (1) EEG decomposition using EMD into IMFs; (2) Application of the five BSS methods to the IMFs, and then computing the correlation coefficient between them and the IMFs extracted from the VEOG and the HEOG

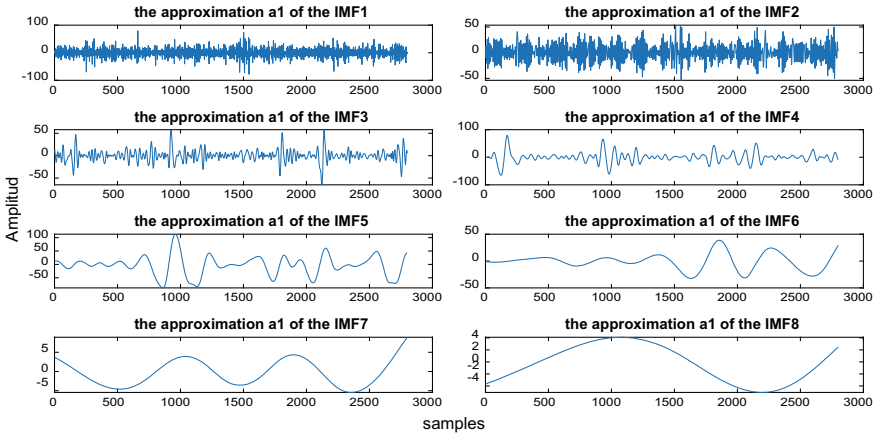


Fig. 4 Approximation coefficient a1 extracted from the EEG's IMFs using the wavelet symlet 1

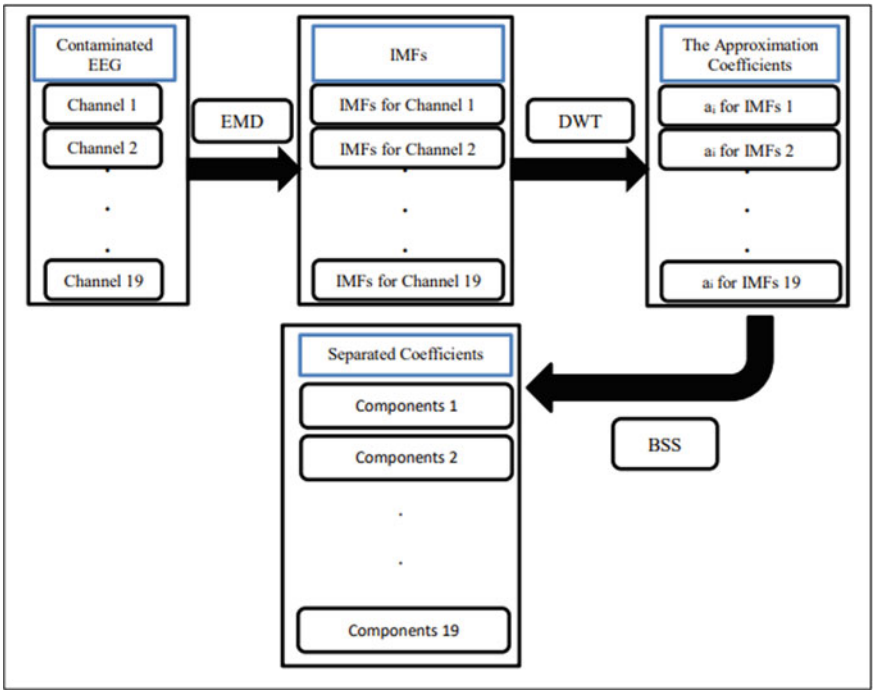


Fig. 5 Second method's step, in this method we add a block of DWT to extract the approximation coefficients from the IMFs and then we try to compute the correlation coefficient using these separated components and the approximation coefficient extracted from the VEOG's and the HEOG's IMFs

between − 1 and + 1. Whereas a number of − 1 denote a perfect negative correlation, a value of + 1 denotes a perfect positive correlation. No correlation exists between the variables, as shown by a value of 0.

The following formula may be used to calculate the Spearman correlation coefficient [22]:

$$\mathrm{SCC} = 1 - 6\sum \frac{r^2}{I\left(I^2 - 1\right)} \tag{13}$$

where r is the square of the difference in the two coordinates' rankings for each data pair, and I is the number of data pairs.

3 Results

In this study, we introduce two techniques that integrate BSS, DWT, and EMD algorithms to segregate the intrinsic mode functions (IMFs) obtained from individual EEG signals, along with the approximated coefficients derived from these IMFs. Our approach involves the assessment of BSS methods applied to both EEG data's IMFs and approximation coefficients, utilizing the previously mentioned SCC metric.

For the initial method, we gauge the SCC values between the extracted components and the IMFs sourced from VEOG and HEOG. Subsequently, we identify the mean values across 54 samples by opting for the larger SCC value.

In the second approach, we calculate the SCC between the isolated components and the approximation coefficients taken from HEOG and VEOG's IMFs, with the inclusion of the DWT block. The proposed technique is implemented using MATLAB 2019a in conjunction with EEGLAB, an interactive toolkit within MATLAB designed for processing EEG data.

EEG signal analysis can benefit from various techniques to extract meaningful information. EMD utilizes a data-driven approach to break down EEG signals into IMFs. These IMFs capture the different oscillatory modes at various scales present in the signal. Conversely, DWT offers a multi-resolution analysis by decomposing EEG signals into different frequency bands. This allows for the identification of specific frequency components associated with different cognitive states or neurological disorders. DWT provides a localized representation of the signal in both the time and frequency domains, aiding in the characterization and classification of EEG patterns. To address the issue of artifacts contaminating EEG signals, BSS techniques such as ICA are employed. These methods aim to separate EEG signals into their underlying source components, isolating and extracting various artifacts like eye blinks, muscle activities, and external interference. The removal of these unwanted components enhances the interpretability of EEG signals.

Combining EMD, DWT, and BSS techniques provides researchers with the complementary strengths of each method. EMD offers flexible and adaptive decomposition, DWT enables frequency-domain analysis and event localization, and BSS aids in signal separation and denoising. This integrated approach facilitates a comprehensive understanding of EEG signals, allowing for accurate interpretation of brain activity, detection of abnormalities, and investigation of cognitive processes. This framework has significant potential for advancing EEG signal processing and its applications in neuroscience, clinical diagnostics, and brain-computer interfaces.

The main objective of our work is to evaluate and assess the effectiveness of BSS methods when applied to the IMFs representing the underlying oscillatory modes of different scales in EEG signals. Specifically, we aim to test these methods by applying them to the approximation coefficients that represent the high-frequency components of the IMFs.

For the first method, the best SCC result is obtained using the ACSOBI-RO method, according to the measurement computed using the VEOG, which is equal to

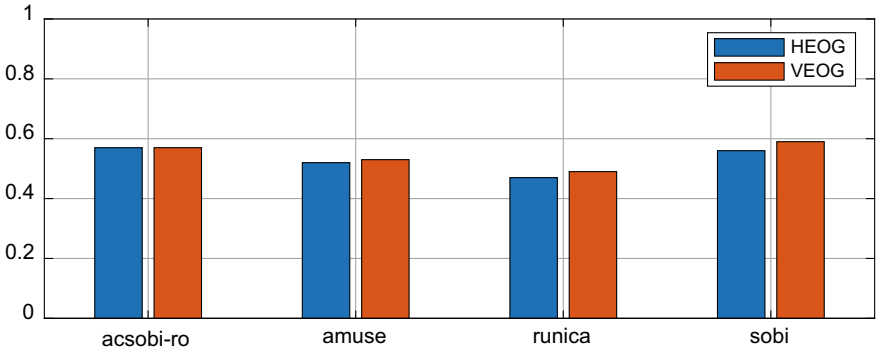


Fig. 6 Results of the first method (EMD-BSS)

0.57 and the higher value obtained using HEOG is given by the SOBI method which is equal to 0.59 (Fig. 6).

Figures 7, 8, 9 and 10 present the SCC results estimated using the HEOG data for each BSS method.

For the ACSOBI-RO method, the higher value of SCC is computed using the wavelet coif4 at level 3 which is equal to 0.59. On the other side, the SOBI method produced a higher value of SCC using the wavelet db1 at level 4 equal to 0.61. The best value for the RUNICA method is equal to 0.56, which is given using the wavelet db8 at level 4. And for the AMUSE method, the greatest value of SCC is 0.54 using sym3 at level4.

For the SCC values calculated using the VEOG signals, Figs. 11, 12, 13 and 14 present our results. The higher value of SCC for the ACSOBI-RO method is calculated using the wavelet coif4 at level 3, which is equivalent to 0.59. The SOBI technique, on the other hand, generated a higher SCC value of 0.60 using the wavelet db1 at level 4. The optimum result for the RUNICA method is 0.57, obtained using

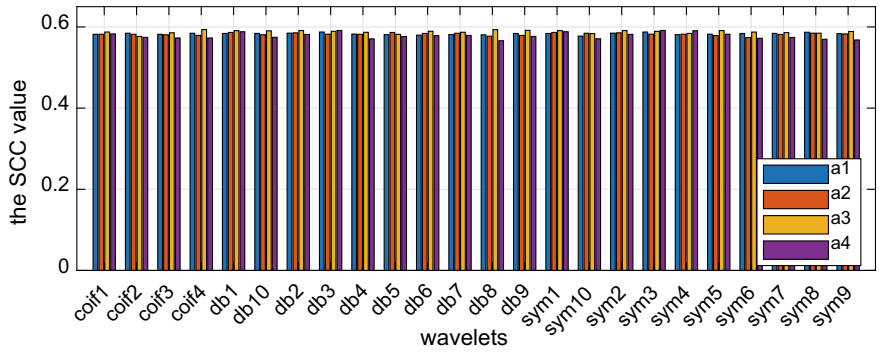


Fig. 7 SCC value calculated using the HEOG data and the separated components using the ACSOBIRO

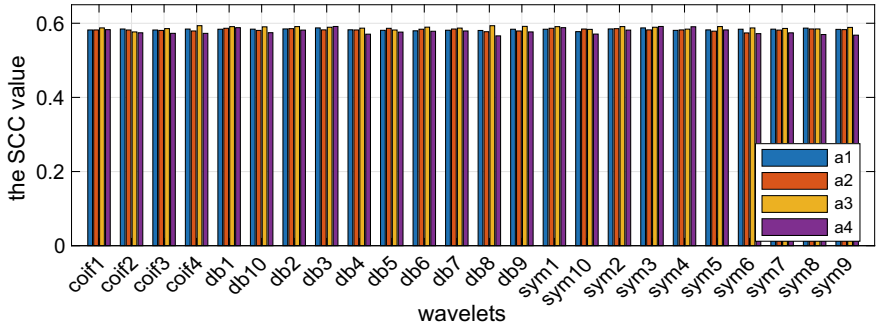


Fig. 8 SCC value derived from HEOG data and the separated components determined from SOBI

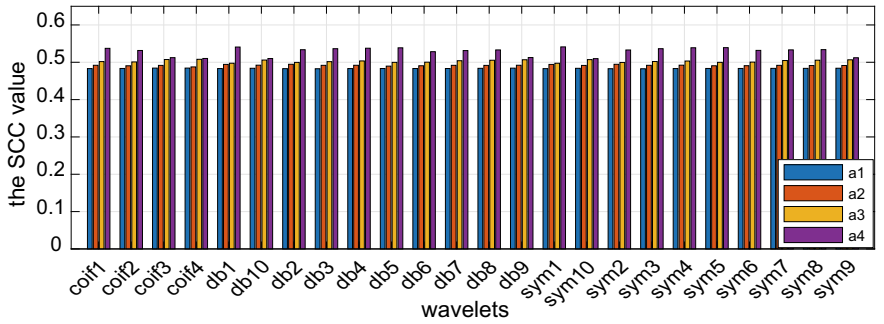


Fig. 9 SCC value obtained from HEOG data and the RUNICA separated components

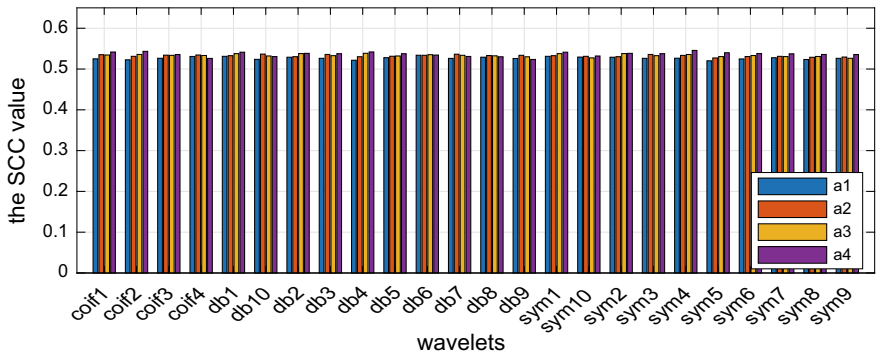


Fig. 10 SCC value obtained from HEOG data and the amuse separated components

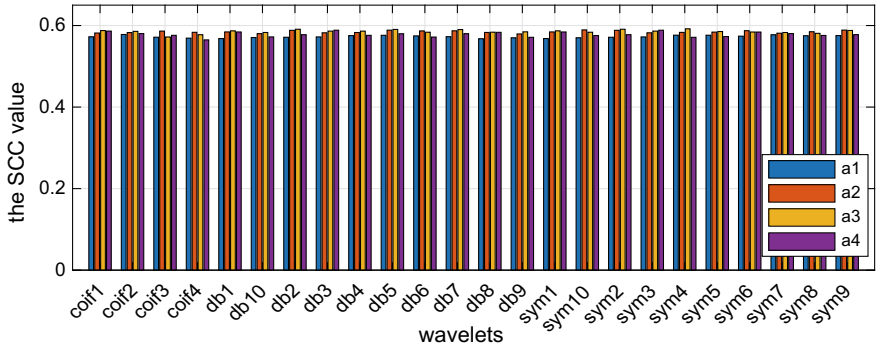


Fig. 11 SCC results obtained using the VEOG signals and the ACSOBIRO results

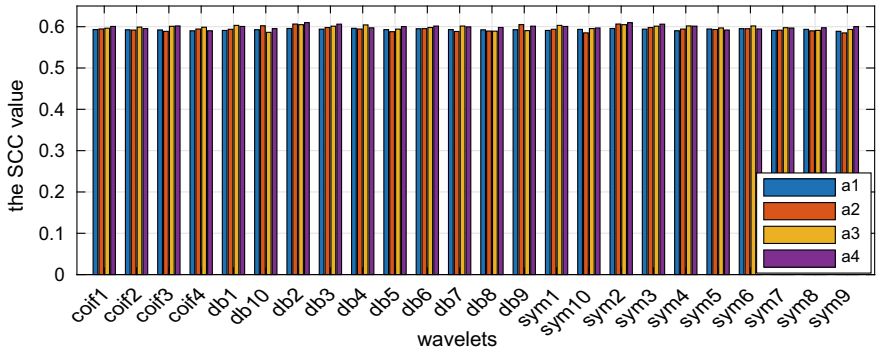


Fig. 12 SOBI results using the VEOG data

the wavelet db8 al level 4. The highest SCC number for the AMUSE method is 0.55 when using sym3 at level4.

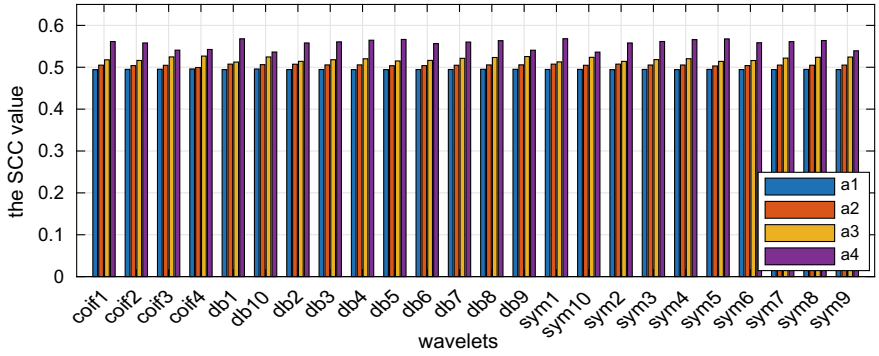


Fig. 13 SCC values calculated using the component extracted using RUNICA and the VEOG signals

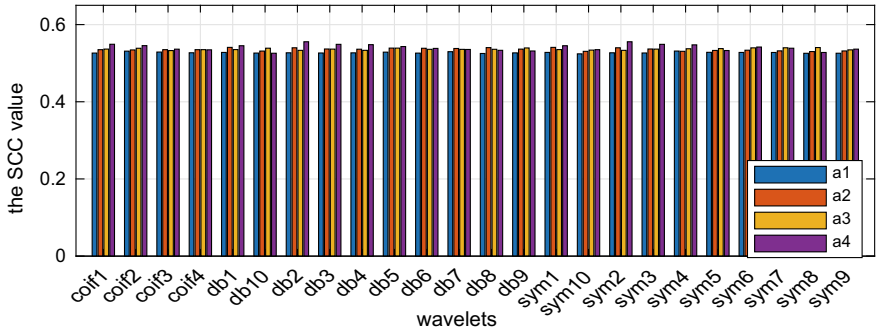


Fig. 14 SCC values derived from the amuse component and the VEOG signals

4 Conclusion

This study presented a novel method by combining the BSS technique with the EMD and DWT method. Further, the performance of the separation method is tested and evaluated after applying them to the data extracted from the EEG signals by using the EMD and the DWT methods.

The EMD method processes a non-stationary signal by decomposing them into IMFs, which are a representation of oscillatory components with different time scales. That makes the idea of separating these kinds of data interesting. As a result of the first approach, the best-performing BSS method is ACSOBI-RO with a value of 0.57. The DWT aims to decompose a signal into two types of coefficients, and one of these coefficients is the approximation coefficient which represents the low-frequency component of the related signal. In terms of this letter, we extracted the approximation coefficient from the EEG's IMFs, and then, we fed them into the

BSS methods, in order to test the separation approaches by applying them to these extracted data.

The best-performing algorithm based on this second method is SOBI, which gives the greatest value by applying the wavelet db1 at level 4, which is equal to 0.61 using the HEOG signals. And by applying the wavelet db2 at level 4, which is equal to 0.60 using the VEOG signal. As a conclusion of the second method, the DWT step has successfully increased the SCC result.

Acknowledgements This paper and the study it supports would not have been possible without the incredible help of my supervisors, Mm. M.MIYARA, Mr. T. Drissi Belhousine, and Mr. B. Nsiri. Their enthusiasm, knowledge, and careful attention to detail have inspired me and kept my work on schedule.

References

1. Mannan MM, Kamran MA, Jeong MY (2018) Identification and removal of physiological artifacts from electroencephalogram signals: a review. IEEE Access 6:30630–30652. https://doi.org/10.1109/ACCESS.2018.2842082
2. Mowla MR, Ng S-C, Zilany MSA et al (2015) Artifacts-matched blind source separation and wavelet transform for multichannel EEG denoising. Biomed Signal Process Control 22:111–118
3. Sahonero-Alvarez G, Calderon H (2017) A comparison of SOBI, FastICA, JADE and Infomax algorithms. In: Proceedings of the 8th international multi-conference on complexity, informatics and cybernetics, 17–22
4. Safie SI, Rahim R (2019) Quality assessment on muscle locations for speech representation. Indonesian J Electr Eng Comput Sci 17:957–967. https://doi.org/10.11591/ijeecs.v17.i2.pp957-967
5. Jamil Z, Jamil A, Majid M (2021) Artifact removal from EEG signals recorded in a non-restricted environment. Biocybern Biomed Eng 41:503–515.https://doi.org/10.1016/j.bbe.2021.03.009
6. Zhao M, Bonassi G, Guarnieri R et al (2021) A multi-step blind source separation approach for the attenuation of artifacts in mobile high-density electroencephalography data. J Neural Eng 18(6):066041
7. Chen X, Xu X, Liu A et al (2017) The use of multivariate EMD and CCA for denoising muscle artifacts from few-channel EEG recordings. IEEE Trans Instrum Measur 67:359–370.https://doi.org/10.1109/TIM.2017.2759398
8. Paradeshi KP, Kolekar UD (2017) Removal of ocular artifacts from multichannel EEG signal using wavelet enhanced ICA. In: 2017 international conference on energy, communication, data analytics and soft computing (ICECDS). IEEE, 383–387. https://doi.org/10.1109/ICECDS.2017.8390150
9. Klados MA, Bamidis PD (2016) A semi-simulated EEG/EOG dataset for the comparison of EOG artifact rejection techniques. Data Brief 8.https://doi.org/10.1016/j.dib.2016.06.032
10. Kopsinis Y, Mclaughlin S (2009) Development of EMD-based denoising methods inspired by wavelet thresholding. IEEE Trans Signal Process 57:1351–1362.https://doi.org/10.1109/TSP.2009.2013885
11. Echeverria JC, Crowe JA, Woolfson MS et al (2001) Application of empirical mode decomposition to heart rate variability analysis. Med Biol Eng Comput 39:471–479.https://doi.org/10.1007/BF02345370

12. Echeverría JC, Crowe JA, Woolfson MS et al (2001) Application of empirical mode decomposition to heart rate variability analysis. Med Biol Eng Comput 39:471–479. https://doi.org/10.1007/BF02345370
13. Toulni Y, Belhoussine Drissi T, Nsiri B (2021) ECG signal diagnosis using discrete wavelet transform and K-nearest neighbor classifier. ACM Int Conf Proc Ser: 3457628. https://doi.org/10.1145/3454127.3457628
14. Toulni Y, Belhoussine Drissi T, Nsiri B (2021) Electrocardiogram signals classification using discrete wavelet transform and support vector machine classifier. IAES Int J Artif Intell 10:960–970.https://doi.org/10.11591/ijai.v10.i4.pp960-970
15. Drissi TB, Zayrit S, Nsiri B et al (2019) Diagnosis of Parkinson's disease based on wavelet transform and Mel frequency cepstral coefficients. Int J Adv Comput Sci Appl 10
16. Chaovalit P, Gangopadhyay A, Karabatis G et al (2011) Discrete wavelet transform-based time series analysis and mining. ACM Comput Surveys (CSUR) 43:1–37.https://doi.org/10.1145/1883612.1883613
17. Daqrouq, K, Hilal TA, Sherif M et al (2009) Speaker identification system using wavelet transform and neural network. In: International conference on advances in computational tools for engineering applications. IEEE, 559–564. https://doi.org/10.1109/ACTEA.2009.5227953
18. Noorbasha SK, Sudha GF (2021) Removal of EOG artifacts and separation of different cerebral activity components from single channel EEG—an efficient approach combining SSA–ICA with wavelet thresholding for BCI applications. Biomed Signal Process Control 63:102168. https://doi.org/10.1016/j.bspc.2020.102168
19. Xi J, Chicharo JF, Tsoi AC et al (2000) On the INFOMAX algorithm for blind signal separation. In: WCC 2000-ICSP 2000. 2000 5th international conference on signal processing proceedings. 16th world computer congress 2000. IEEE, 425–428. https://doi.org/10.1109/ICOSP.2000.894523
20. Langlois D, Chartier S, Gosselin D (2010) An introduction to independent component analysis: InfoMax and FastICA algorithms. Tutor Quantitative Methods Psychol 6:31–38
21. Tong L, Soon VC, Huang YF et al (1990) AMUSE: a new blind identification algorithm. In: IEEE international symposium on circuits and systems. IEEE, pp 1784–1787
22. Myers L, Sirois MJ (2004) Spearman correlation coefficients, differences between. Encyclopedia of statistical sciences, vol 12

Image Extraction Approaches for Density Count Measurement in Obstruction Renography Using Radiotracer 99mTc-DTPA



Pradnya N. Gokhale, Babasaheb R. Patil, and Sameer Joshi

Abstract This research pursued to define quantitative amounts of renal radiotracer's density inside the kidneys in terms of renal obstruction levels by applying image processing transforms FFT, wavelet, and Harr-like extraction and finding correlation among density measured by transforms and radioactive counts by gamma machines. In this retrospective study, we considered 140 renal scintigraphy scans from which 110 were detected with renal obstruction (hydronephrosis) disorder ranging from moderate to severe and 30 cases were found to be normal functioning kidneys. Of 110 cases, 64 cases were diagnosed with left kidney hydronephrosis (LK-HL) with an M/F ratio of 43/21, and 46 cases were diagnosed with right kidney hydronephrosis (RK-HR) with an M/F ratio of 23/23. The mean age of the selected case is 25.65 $\pm$ 24.58 years. Three image-enhanced transforms, namely FFT, wavelet, and Harr-like extraction, are applied to renal scans to detect the density of darkness inside the kidney. Correlation among density-measured count and scintigraphy-measured radioactive count by Spearman's correlation method. There was a strong positive correlation between FFT-measured density inside the kidney and dynamic renal scintigraphy using ^{99m}Tc-DTPA measured radioactive counts in the case of both left hydronephrosis kidney and right hydronephrosis kidneys ($p = 0.81$ and $p = 0.80$, respectively). Also found a moderately positive correlation between wavelet

Authors' Contributions The study was designed by Pradnya Gokhale. Material preparation and data collection were performed by Pradnya Gokhale and Prof. Dr. Babasaheb Patil. The data analysis was performed by Pradnya Gokhale. The first draft of the manuscript was written by Pradnya Gokhale, and all authors commented on previous versions of the manuscript. Prof. Dr. Babasaheb Patil played a crucial role in developing the methodology for the study, and Dr. Sameer Joshi provided insightful feedback on the analysis carried out. All authors read and approved the final manuscript.

P. N. Gokhale (✉)
Sardar Patel College of Engineering, University of Mumbai, Mumbai, Maharashtra, India
e-mail: pradnya.gokhale@spce.ac.in

B. R. Patil
VIMEET, University of Mumbai, Mumbai, Maharashtra, India

S. Joshi
Miraj Nuclear Medicine AND Molecular Imaging Center, Miraj, Maharashtra, India

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2023

S. Shakya et al. (eds.), *Fourth International Conference on Image Processing and Capsule Networks*, Lecture Notes in Networks and Systems 798,
https://doi.org/10.1007/978-981-99-7093-3_4

and Harr-like transform's measures and scintigraphy measures. In conclusion, this study showed that the application of the FFT object detection technique can be a better option to count the density of renal scan in terms of obstruction level as compared to Harr-like and wavelet transform methods, and these quantitative measures of this density can be considered to define the obstruction/hydronephrosis levels instead of reporting moderate level to severe levels of renal obstruction disorder.

Keywords Renal scintigraphy · Obstruction · Feature extraction · Radiotracer · Hydronephrosis · Correlation coefficient

1 Introduction

Renal scintigraphy images provide information regarding renal radiotracer excretion activity that is regularly categorized by ample noise and low contrast and resolution features while using ^{99m}Tc-DTPA, and it provides observations of the regions of interest in a very complicated way which finds to be difficult to analyze in terms of diagnosing the level of obstruction, i.e., hydronephrosis [1]. Before the scintigraphy procedure, patients were hydrated for 30–60 min. They were then positioned supine to a gamma camera, and the acquisition process was initiated. The radiopharmaceutical DTPA was then injected as a bolus dose of 5–10 mCi for DTPA scintigraphy, depending on the patient's age, gender, weight, medical history, and type of renal parameters. Clinical experts decide the appropriate radionuclide dose based on the patient's characteristics. This renal imaging technique thus provides information on peak time and half time of radiotracer's accumulation inside the kidney, glomerular filtration rate (GFR), and radioactive counts on the bases of the tracer's excretion time activity curve stimulated after drawing the regions of interest around the kidneys [2–4]. In renal scintigraphy, motion detection is restricted due to the high level of noise and fading of organ boundaries, and it represents inherent image characteristics. These restrictions can cause a reduction in the precision of motion detection in cases especially radiotracer's transit process through renal parenchyma to the cortex. Though this imaging provides a diagnosis of renal obstruction in terms of hydronephrosis, it fails to count the levels of renal obstruction in terms of quantity of figures which will directly help the nephrologist to understand the status of obstruction. This research study utilizes 110 renal scan samples which were processed using image edge extraction transforms to quantify the density count of radiotracer inside the kidney.

1.1 Characteristics of DTPA in Renal Imaging

1. **Glomerular filtration**: DTPA is mainly cleared by glomerular filtration which means it is filtered through the glomeruli in the kidneys and evacuated in the

urine. It provides information about the glomerular filtration rate and the overall function of the parenchyma.

2. **Static and dynamic Imaging**: For static and dynamic renal imaging, DTPA can be used, where in the static method images are acquired at specific time points after the injection and in the dynamic method sequential images are acquired over several minutes. In static mode, it allows for the evaluation of renal morphology like renal size, dimensions, and shape while in dynamic mode it provides evaluation of renal blood flow and tubular function [5].

2 Materials and Methods

A. Sample Selection Details

Patients aged less than 10 years and those aged more than 65 years were excluded from the study. Renal scans were selected only if they showed gradual tracer accumulation into the dilated pelvis without significant excretion after diuretic administration. The study did not include cases with horseshoe-shaped kidneys, renal transplants, donors, or single-functioning kidneys. As shown in Table 1, for this study, 64 cases of left kidney hydronephrosis (M/Fe: 43/21) and 46 cases of hydronephrosis (M/Fe: 23/23) were considered. The mean years of age of the selected case are 25.65 ± 24.58 years.

B. Renal Scintigraphy Image Processing

By using suitable color measure, a parametric appearance of each kidney can be acquired; permitting the pixel-by-pixel true mean transit time in the cortical zone to be read immediately and the mean transit time (MTT) image can be accommodated in renal image processing easily [6]. Estimation of absolute renal uptake of radiotracer and its evacuation rate through the kidney can be achieved by using iterative deconvolution and Rutland–Patlak methods in which reported matrix inversion deconvolution values were significantly higher than Rutland–Patlak values [5, 7].

Initial recognition of clinically substantial renal hydronephrosis affected by ureteropelvic obstruction was achieved by using automated signal analysis and a machine learning framework. But the value of the signal investigation depends extremely on the quality of the data obtained by skilled nuclear medicine physicians or technologists' accuracy level of earlier detection of obstructed cases effects [8].

Table 1 Sample selection data

Parameter	Male	Female	Total	Sample types
Left hydronephrosis (HL)	43	21	64	Renal hydronephrosis (renal obstruction) samples
Right hydronephrosis (HR)	23	23	46	
Normal kidney	15	15	30	
Total	81	59	140	

Renal scintigraphy scans are regularly categorized by much noise and low contrast and resolution while using ^{99m}Tc-DTPA, which observes the regions of interest very complicated and difficult to analyze in terms of diagnosing the level of obstruction, i.e., hydronephrosis [9]. Here, each digital image is a set of coordinate values "*nij*" which corresponds to the number of counts in a set of pixels identified by (*i*, *j*). These pixels are signified in matrix form by {(*i*, *j*, *nij*)}. Drawing the defined region of interest (ROI) around the high-density darkened area inside the kidneys and counting the activity on all the selected pixels for *P*-recorded images, not only allows the tracing over radioactivity period but also the evolution of the activity of the structure analogous to the selected pixels. Thus, access to a dynamic study of the functioning of the renal system is achieved. Data is then articulated in a matrix pattern by inserting the time parameter *tk* which designates the development of images: {(*i*, *j*, *nij*), *tk*} [10].

The study incorporates three image enhancement techniques, namely wavelet transforms, FFT transform, and Harr-like extraction, to analyze each sample obtained from renal scans. The objective is to quantify the darkness within the kidney region by measuring the density of the last frame of each scan, specifically the 30th image in the case of SIEMENS gamma scans.

These image enhancement techniques aim to improve the visibility and clarity of the kidney region, allowing for more accurate measurements of darkness. The density of the last frame is considered a representative measure of the darkness within the kidney. To clinically validate these results, they are compared with the amount measured by the Gamma machine's radioactive kidney counts. This comparison ensures that the quantification of darkness obtained through image enhancement techniques aligns with the measurements obtained directly from the Gamma machine, which is a reliable and established method.

Thus, in this study, as shown in Fig. 1, image processing steps are designed and followed to find the best of transform which will approximately match to the clinical report and will provide quantity in terms of density of the obstructed part of the kidney.

C. Significance of image enhancement transforms

In the study, the density counts obtained through the application of FFT, wavelet, and Harr-like extraction transforms indicate the presence of radioactive tracer inside the kidney. Normally, when the renal system is functioning properly, the tracer is

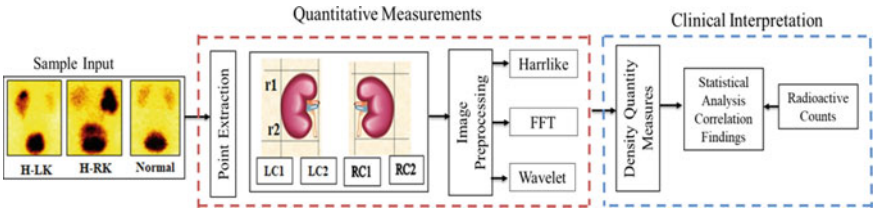


Fig. 1 Renal scintigraphy scan process

able to excrete out of the kidney. However, in the case of hydronephrosis or renal obstruction, the tracer is unable to pass through the obstructed area and remains inside the kidney.

To measure the density of darkness inside the kidney, image processing techniques are applied to enhance and extract the kidney structure from the renal scan images. The transforms, namely FFT, wavelet, and Harr-like extraction, are utilized to improve the visibility and clarity of the kidney region. By enhancing and extracting the kidney structure, the density of darkness, which represents the presence of the radioactive tracer, can be measured.

These image processing techniques provide a quantitative measure of the density of darkness inside the kidney, indicating the extent of renal obstruction or hydronephrosis.

Feature extraction is a process that involves extracting relevant information or characteristics from raw data, in this case, images. The main objective of feature extraction is to represent the raw image in a reduced form while preserving important information. By reducing the dimensionality of the data, feature extraction aims to simplify the representation of the image while retaining its discriminative properties.

One of the primary motivations for feature extraction is to reduce the number of false findings or irrelevant information. By extracting and focusing on the most relevant features, the likelihood of false detections or misleading information can be minimized.

In the context of image feature extraction, the goal is to transform the low-dimensional space of an image sample into a higher-dimensional space where meaningful features can be identified and utilized. These features serve as the fundamental characteristics that can be used to identify and distinguish one image from another [7].

Thus, the better features can entirely recognize the object (here in the case of darkness inside the kidney) which is required to be identified. In this study, three effective feature extraction methods, namely wavelet transform, FFT transform, and Harr-like extraction transform, are applied to renal scan to extract the structures so that the design of classifiers can be specific and more accurate to define the density of darkness [11].

The flowchart mentioned in Fig. 2 represents the steps involved in quantifying the density count of darkness. These steps likely include preprocessing the renal scan images, applying the feature extraction methods, and using the extracted features to train and evaluate classifiers. The flowchart provides a visual representation of the overall process followed in the study.

Renal scan frame of 30 images/sample is selected as an input image. As image sizing for every sample is different, all images were resized with matrix 330 × 1100 scale. Renal scan is passed through RGB to gray conversion to enhance the required pixels. Point extraction is performed on each of left kidney and right kidney of all cases. Object detection is achieved by applying the mentioned three image enhancement techniques. The dark area inside the kidney is counted in terms of the density of darkness.

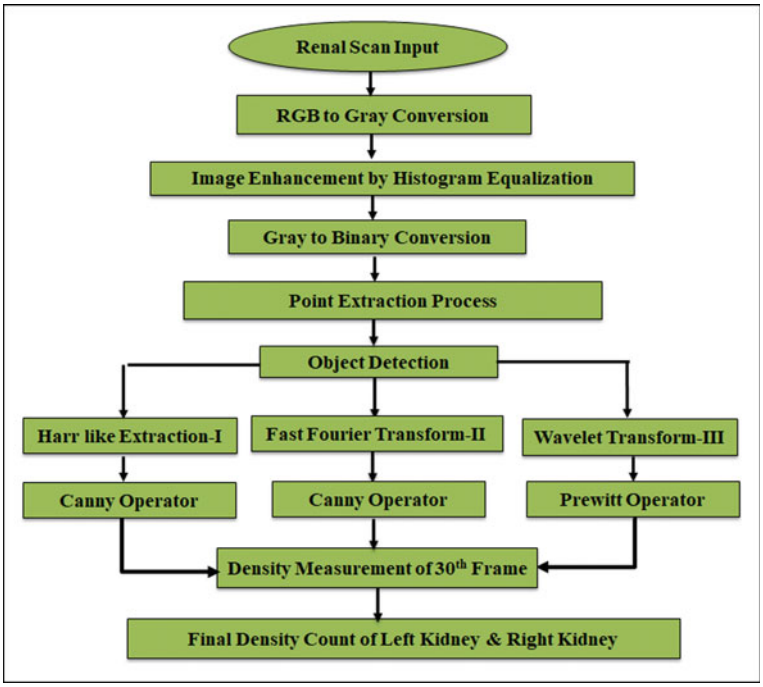


Fig. 2 Renal image processing flowchart

As shown in Fig. 3. for every kidney pair scan edge identification is done using Canny operator for Harr-like and FFT methods and Prewitt operator for wavelet transform by defining edges with row1, row 2 left kidney column LC1 and LC2 and right kidney column RC1, RC2.Three edge detection transforms, namely wavelet, FFT, and Harr-like extraction, were applied on every 30th last frame of every scan, and three density counts were identified using these transform techniques. The wavelet transforms are mathematical tools used for extracting information from images. These transforms provide both frequency and spatial localization of image features, making them useful for analyzing images in both the frequency and spatial domains.

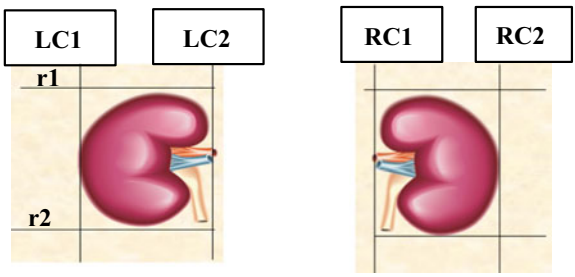


Fig. 3 Point extraction of kidney pair

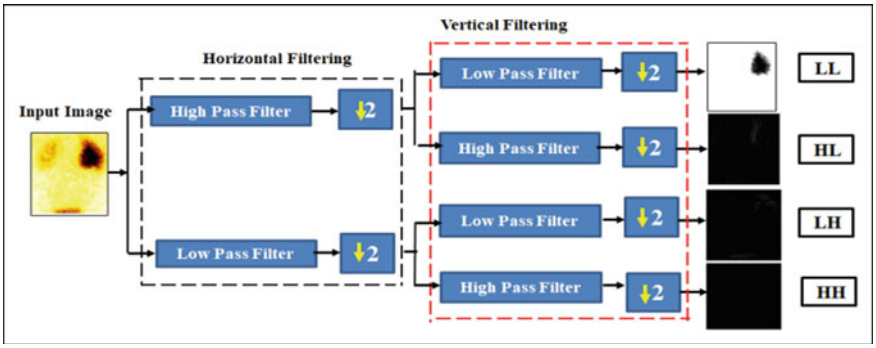


Fig. 4 Image extraction method using wavelet transform

The wavelet transform decomposes an image into different frequency components, allowing the extraction of fine details at different scales. It achieves this by using a set of wavelet functions that are scaled and shifted to analyze different parts of the image. The transform captures both low-frequency components, which represent global image features, and high-frequency components, which represent local image details.

One advantage of wavelet-based transforms is their ability to provide temporal resolution, meaning they can capture changes in frequency and spatial content over time. This temporal resolution makes wavelet transforms suitable for analyzing dynamic or time-varying images, as they can capture variations in both frequency and spatial information [12]. Figure 4 shows wavelets various filters and their concern results.

A Haar-like feature is defined by dividing a rectangular region of an image into sub-rectangles and computing the difference between the sum of pixel intensities in the white sub-rectangles and the sum of pixel intensities in the black sub-rectangles. These features are typically represented as adjacent black and white rectangles. The Haar-like features capture local intensity variations in an image and can be used to detect edges, lines, corners, and other simple patterns. They are computationally efficient to compute and are often used as a preliminary feature extraction step in object detection algorithms, such as the Viola-Jones algorithm for face detection.

By using Haar-like features, the algorithm can learn to distinguish between different patterns or objects based on the variations in intensity captured by these features. These features provide a simple yet effective way to represent and detect visual patterns in images.

Harr-like transform can be stated in matrix form which is expressed as:

$$T = HFH^{\tau}$$

Here, F is an "$N \times N$" image matrix, H is an "$N \times N$" Harr-like transformation matrix, and T is the subsequent "$N \times N$" transform [13].

The fast Fourier transform-low pass (FFT) is a mathematical field. In this study, FFT-HP is used for renal image filtering, reconstruction, and density count measurements. It is an important image processing tool that is used to decompose an image into its sine and cosine components.

While dealing with the derivation of the FFT it is expressed as:

$$F(u) = \sum_{X=0}^{M-1} f(x) W_M^{ux}$$

where, $u = 0, 1, 2, M - 1$, where

$$W_{M=e^{-2j\pi/M}}$$

And M is assumed to be in the form of $M = 2^n$ with "n" positive integer. The transformation output symbolizes the image in the frequency domain, and the input image is the spatial domain equivalent. In the frequency domain image, each point denotes a particular frequency contained in the spatial domain image [14].

The density count measured by these transforms was compared and analyzed to find one of transform performs better in terms of precise and approximate density count close to the radioactive count provided by the renal gamma scan.

3 Results

Sixty-four patients with left kidney hydronephrosis cases (M/Fe: 43/21) and forty-six patients with right kidney hydronephrosis cases (M/Fe: 23/23) were considered in this study. Mean and standard deviation of scintigraphy-measured radioactive count.

3.1 Mean and Standard Deviation of Transforms

The mean is a statistical quantity that signifies the average of a set of values. It is calculated by summing up all the values in the set and then dividing the sum by the total count of values. The mean provides a central value that represents the typical value of the dataset.

While standard deviation is a statistical measure that quantifies the amount of variation or dispersion in a set of numbers. It measures how much the individual data points deviate from the mean.

A higher standard deviation indicates that the data points are more spread out from the mean, while a lower standard deviation indicates that the data points are closer to the mean. Both the mean and standard deviation are important statistical

Table 2 Mean ± S.D. of radioactive counts of HL, HR, and normal cases by scintigraphy scan

Type of category		Mean value ± S.D.
DTPA HL	Left kidney	66.76 ± 8.25
	Right kidney	33.24 ± 8.25
DTPA HR	Left kidney	36.66 ± 10.44
	Right kidney	65.34 ± 10.44
Normal	Left kidney	48.43 ± 4.59
	Right kidney	51.57 ± 4.59

Table 3 Mean ± S.D. of transforms detected density counts of HL, HR, and normal cases

Type of category		FFT	Wavelet	Harr
		Mean ± S.D.	Mean ± S.D.	Mean ± S.D.
DTPA HL	LK	68.28 ± 23.11	65.94 ± 22.30	57.06 ± 18.94
	RK	28.59 ± 14.55	27.44 ± 13.76	25.59 ± 13.11
DTPA HR	LK	27.78 ± 18.50	26.87 ± 21.24	22.87 ± 18.50
	RK	70.22 ± 21.22	67.13 ± 20.43	63.43 ± 19.39
Normal	LK	19.13 ± 4.92	18.30 ± 4.60	15.63 ± 4.11
	RK	18.50 ± 4.60	17.53 ± 4.43	16.70 ± 4.28

measures that provide information about the central tendency and spread of a data set, respectively.

Tables 2 and 3 indicate the mean and the standard deviation values of transforms measured density counts including normal functioning kidney cases. As shown in Table 3, hydronephrosis case kidney's mean and standard deviation found to be higher than other two transforms' results.

3.2 Radioactive Counts Measurement

Renal scintigraphy counts measurement by gamma machine (Gate's method). The gamma machine records the number of gamma rays detected during a specified time period, typically in counts per minute (CPM). These counts are a measure of the radiotracer activity in the kidneys, reflecting the function and excretion of the radiotracer by the renal system.

It is expressed as:

$$\text{Uptake Kidney Counts} = \frac{(\text{Kidney Counts} - \text{Background Counts})}{e^{(-\mu \times \text{Kidney Depth})}}$$

where μ = Linear attenuation coefficient of radiotracer.

Right kidney depth = 13.3((Wt.)/(Ht.)) + 0.7.

Table 4 Comparison among mean ± S.D. of transforms detected density counts of HL, HR, and radioactive scintigraphy counts

Type of category		Mean ± S.D. of transforms detected density counts			Mean ± S.D. uptake counts
		FFT	Wavelet	Harr-like	
DTPA HL	LK	68.28 ± 23.11	65.94 ± 22.30	57.06 ± 18.94	67.40 ± 7.02
	RK	28.59 ± 14.55	27.44 ± 13.76	25.59 ± 13.11	32.60 ± 7.02
DTPA HR	LK	27.78 ± 18.50	26.87 ± 21.24	22.87 ± 18.50	34.66 ± 10.44
	RK	70.22 ± 21.22	63.13 ± 20.43	63.43 ± 19.39	68.34 ± 10.44

Left kidney depth = 13.2 ((Wt.)/(Ht.)) + 0.7.

Table 4 shows that calculated mean and standard values of all density count measures by three transforms along with radiotracer's uptake counts and we found that, density counts measured by FFT transform are approximately close to that of radioactive scintigraphy uptake counts measured by gamma camera. This identifies that the FFT transform offers more precise uptake count estimation in terms of mean value and variability.

3.3 Statistical Correlation Findings

Spearman's rank correlation coefficient, often denoted as ρ (rho), is a statistical measure that quantifies the strength and direction of the monotonic relationship between two ranked variables. It is based on the ranks of the observations rather than the actual values. It is expressed as:

$$\rho = 1 - \frac{6 \sum \mathrm{d}i^2}{n(n^2 - 1)}$$

where di = difference among two ranks of every observation, and n = number of observation

Spearman rank finding is a non-parametric test that is used to quantify the degree of relationship [15]. All statistical evaluations were achieved using the SPSS 0.20 version software.

Findings of Spearman's rank coefficient values help to understand the linear relationship among two continuous variables [15–17]. As shown in Tables 4 and 5, among three techniques FFT techniques were found to be having a strongly positive correlation with radioactive counts (ρ = 0.804:0.801) which indicates that the density count measured by FFT transform is precisely close to gamma machine results.

Table 5 indicates correlation count of the transform counts and scintigraphy counts in case of left kidney hydronephrosis, while Table 6 indicates coefficients in case of right kidney hydronephrosis. In both cases, FFT transform correlation was found to be strongly positive as compared to wavelets and Harr-like's transforms.

Table 5 Correlation of the transform counts and scintigraphy counts in case of LK-HL

Left kidney hydronephrosis		
Type of transforms	Radioactive counts by scintigraphy scan	Correlation interpretation
FFT	$\rho = 0.804$	Strongly positive
Wavelet	$\rho = 0.595$	Moderately positive
Harr-like	$\rho = 0.575$	Moderately positive

Table 6 Correlation of the transform counts and scintigraphy counts in case of RK-HR

Right kidney hydronephrosis		
Type of transforms	Radioactive counts by scintigraphy scan	Correlation interpretation
FFT	$\rho = 0.801$	Strongly positive
Wavelet	$\rho = 0.573$	Moderately positive
Harr-like	$\rho = 0.572$	Moderately positive

4 Discussion

The standard deviation is a summary expression of the differences of each observation from the mean and analysis of these observations indicates the location of the center of the data and the standard deviation is the spread in the data. Comparison among mean and standard deviation values derived from three different transforms and renal scintigraphy gamma machines radioactivity counts indicate that:

- In the case of DTPA-HL, density count values derived by FFT transform are more as compared with remaining transforms, as in the case of the left kidney (68.28 $\pm$ 23.11) is more than the right kidney (28.59 $\pm$ 14.55) and find to be nearer to the radioactive count of the left kidney (67.40 $\pm$ 7.02) and right kidney (32.60 $\pm$ 7.02), respectively, measured by the gamma scanning output.
- Similarly, in the case of DTPA-HR, density count values derived by FFT transform are more than the remaining transforms, as in the case of the right kidney (70.22 $\pm$ 21.22) than the left kidney (27.78 $\pm$ 18.50) and found to be closer to the radioactive count of the right kidney (68.34 $\pm$ 10.44) and left kidney (34.66 $\pm$ 10.44) respectively counted by the Gamma scanning output.
- In the case of the normal functioning kidneys, all three transform density counts are found to be approximately the same as of scintigraphy radioactive counts.

In this study, Spearman's rank value is used to find a correlation between the transform counts results and the scintigraphy count. We found that, in case of the left kidney and right kidney hydronephrosis, there is a strongly positive correlation between FFT's density count and scintigraphy.

radiotracer count, i.e., $p = 0.81$ and 0.80, respectively, while outputs detected from wavelet transform in case of left kidney and right kidney hydronephrosis found to be moderately positive, i.e., $p = 0.59$ and 0.57, respectively. Also, in the case of Harr-like

extraction transform results of the left kidney and the right kidney hydronephrosis were found to be moderately positive, i.e., $p = 0.57$ and 0.57, respectively.

4.1 Statistical Analysis for Clinical Validation

The accuracy and consistency of a specific method used in clinical studies are proven by the outcomes drawn from clinical validation. This process involves a comparison among the results obtained from the transformations with established clinical standards or reference methods. For clinical validation of these results, regression analysis statistical method is used to assess the relationship between different variables and make predictions or estimations based on this relationship. In this study, it is used to assess the correlation between the density count measured by image processing techniques (such as FFT, wavelet, or Harr-like) and the uptake count of radiotracers in the kidneys.

In relation to the clinical validation of these results, a regression model was used to establish the relationship between the density measurements of normal kidneys from hydronephrotic kidney pair and the density measurements of normal functioning kidney cases. From the graph to assess the goodness of fit of the regression model, the coefficient of determination (R^2) was calculated.

a. **Regression of Coefficient for DTPA_NORMAL Kidneys_HL Cases**

Figure 5 shows a scatter plot indicating the relationship among density counts of DTPA LK-hydronephrosis sample and normal functioning kidney samples measured by three transforms.

As shown in Table 7, the determination coefficient (R^2) value of 0.698 in the case of FFT count and normal kidney counts indicates that approximately 69.8% of the variation in the normal kidney's count can be explained by the FFT transform.

b. **Regression of coefficient for DTPA_NORMAL Kidneys_HR Cases**

Figure 6 shows scatter plot indicating the relationship among density counts of DTPA-RK hydronephrosis sample and normal functioning kidney samples measured by three transforms.

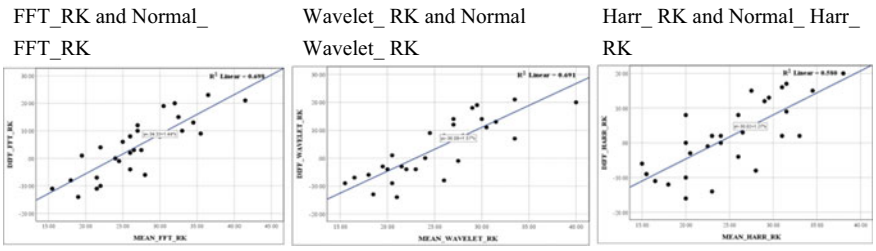


Fig. 5 Regression of coefficient for DTPA_NORMAL kidneys_HL cases

Table 7 Clinical validation analysis-DTPA-HL

Finding parameters	R^2 value	R^2 interpretation
FFT density count and normal kidney density count	0.698	Very strong correlation
Wavelet density count and normal kidney density count	0.691	Strong correlation
Harr-like density count and normal kidney density count	0.580	Strong correlation

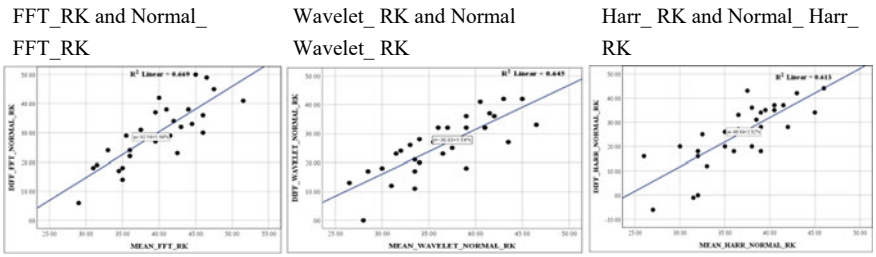


Fig. 6 Regression of coefficient for DTPA_NORMAL kidneys_HR cases

Table 8 Clinical validation analysis-DTPA-HR

Finding parameters	R^2 value	R^2 interpretation
FFT density count and normal kidney density count	0.669	Very strong
Wavelet density count and normal kidney density count	0.654	Strongly
Harr-like density count and normal kidney density count	0.613	Strongly

The R^2 value as shown in Table 8, i.e., coefficient of determination 0.669 shows that approximately 66.9% of the variation in the normal kidney's count can be explained by the FFT transform. This suggests a very strong relationship between the FFT transform's left kidney's density output of DTPA-HR case and the normal left kidney's density count.

5 Conclusion

This study aimed to overcome the limitations of using renal scintigraphy alone for quantitatively assessing the level of hydronephrosis (renal obstruction). By processing the renal obstruction scans using image extraction and enhancement transforms (FFT, wavelet, and Harr-like), you were able to measure the density of radiotracer inside the kidney.

It was found that the density counts obtained from the FFT transform were closely related to the radiotracer radioactive counts measured by the scintigraphy machine. This suggests that the FFT's density count can serve as a supportive measure for

quantitatively diagnosing the level of hydronephrosis, rather than relying on subjective levels such as “moderate” or “severe.” The direct proportionality between the FFT-measured density counts and radiotracer counts indicates that higher density counts correspond to a greater presence of radiotracer inside the kidney, which can be indicative of a higher level of renal obstruction.

By utilizing the FFT transform and its density count as a quantitative measure, it can be considered as an additional parameter for clinicians to evaluate and report the stage of hydronephrosis in a more objective manner. This can potentially enhance the accuracy and precision of diagnosing and assessing renal disorders, allowing for more targeted and effective treatment decisions.

Conflict of Interest Pradnya Gokhale and Prof. Dr. Babasaheb Patil declare that they have no competing interests.

Ethics Approval NA.

Funding NA.

Acknowledgements We want to thank Dr. Alok Pawasker (SSGCH) and Dr. Sameer Joshi (N.M. & MIC) for their suggestions and assistance in finding data sources through various gazettes of hospital resources.

References

1. Volterrani D, Daniele G, Mazzarri S, Guidoccio F, Dardano A, Giusti L, Del Prato S et al (2019) Annual congress of the European Association of Nuclear Medicine 12–16, 2019 Barcelona, Spain. Eur J Nucl Med Mol Imaging 46:1–952
2. Blaufox MD, De Palma D, Taylor A, Szabo Z, Prigent A, Samal M et al (2018) The SNMMI and EANM practice guideline for renal scintigraphy in adults. European Eur J Nucl Med Mol Imaging 45:2218–2228
3. Hans F, Suzanne D, Ariane B, Greet L, Anne R, Iris Van den H, Campbell M (2009) Dynamic renal imaging in obstructive renal pathology a technologist’s guide, ENAM. University Medical Centre Nijmege
4. Andrew T, Taylor DC, Brandon DP, Donald Blaufox M, Emmanuel D, Belkis E, Sandra FG, Andrew JWH (2018) SNMMI procedure standard/EANM practice guideline for diuretic renal scintigraphy in adults with suspected upper urinary tract obstruction 1.0. Seminars Nucl Med 48(4):377–390
5. Piyamas S, Kobchai D, Surapun, Y (2015) The estimation of GFR and ERPF using adaptive edge based active contour for the segmentation of structures in dynamic renal scintigraphy. Int J Innov Comput Inf Control 11(1)
6. Andre AD, Amy P, Hamphrey R (2008) Ham pixel-by-pixel mean transit time without deconvolution. Nuclear Med Commun 29(4):345–348
7. Issa IA (2010) A comparison between the values of renal parenchymal mean transit time by applying two methods, matrix inversion deconvolution and Rutland-Patlak plot. World Appl Sci J 8(10):1211–1219
8. Emily SB, Antonio RP, Elijah B, Pooneh R, Tabrizi RD, Sussman BMS, Eglal S, Massoud M, Hans GP, Marius G (2018) Early detection of ureteropelvic junction obstruction using signal analysis and machine learning: a dynamic solution to a dynamic problem. J Urol 199:847–852

9. Volterrani D, Daniele G, Mazzarri S, Guidoccio F, Dardano A, Giusti L, Del P et al (2019) Annual congress of the European Association of Nuclear Medicine 12–16, 2019 Barcelona, Spain. Eur J Nucl Med Mol Imaging 46:1–952
10. Madsena CJ, Mollerc ML, Zerahnb B, Fynboa C, Jensen JJ (2013) Determination of kidney function with 99mTc-DTPA renography using a dual-head camera. Nucl Med Commun 34(4):322–327
11. Jianhua L, Yanling S (2011) Image feature extraction method based on shape characteristics and its application in medical image analysis. CCIS 224:172–178
12. Van Fleet PJ. Discrete wavelet transformations, 2nd edn.
13. Gonzalez Rafael C. Digital image processing, 3rd edn. Pearson's International, 320–532
14. Oran Brigham E (1988) The fast Fourier transform and its applications. Prentice-Hall, Englewood Cliffs, New Jersey
15. Mukak M (2012) Statistics corner: a guide to appropriate use of correlation coefficient in medical research. Malawi Med J 24(3):69–71
16. Gokhale PN, Patil BR (2022) Correlation between renal tracer parameters derived for recognition of renal obstruction with ^{99m}Tc-DTPA scintigraphy. In: 2022 10th international conference on emerging trends in engineering and technology-signal and information processing (ICETET-SIP-22), 1–6
17. May P, Ehrlich HC, Steinke T (2006) ZIB structure prediction pipeline: composing a complex biological workflow through web services. In: Nagel WE, Walter WV, Lehner W (eds) Euro-Par 2006. LNCS, vol 4128. Springer, Heidelberg, pp 1148–1158

Deep Short-Term Long Memory Technique for Respiratory Lung Disease Prediction



B. Dhiyanesh, Y. Baby Kalpana, S. Rajkumar, P. Saraswathi, R. Radha, and S. Suresh

Abstract Recently, respiratory sampling has been of great help in diagnosing health disorders and diseases by providing valuable information for various medical conditions. Several methods can be used to diagnose respiratory infections, such as collecting blood or sputum samples. It has the potential to predict the risk of respiratory disease-related exacerbations. It is also an essential inflammatory biomarker with which medical professionals can prescribe imaging equipment for respiratory diseases and collect basic patient status information. However, current testing methods can be challenging and time-consuming to diagnose respiratory diseases. Although these techniques are accurate, they increase the risk of contamination and lead to complications when long analysis times are required. To address this issue, the deep short-term long memory method for respiratory lung disease prediction. First, we collected a dataset of respiratory disease sounds, a key indicator of respiratory lung health and disease, from the Kaggle repository. Next, a pre-processing step is performed to ensure a smooth and low-noise signal in the respiratory lung images using *Z*-score normalization. Next, feature extraction can be performed using the local binary Gabor filter (LBGF) to efficiently detect local spatial patterns and grayscale variation in an image. Finally, the deep short-term long memory (DSTLM)

B. Dhiyanesh (✉)
CSE, Dr. N.G.P. Institute of Technology, Coimbatore, India
e-mail: dhiyanu87@gmail.com

Y. Baby Kalpana
CSE, P.A. College of Engineering and Technology, Pollachi, India

S. Rajkumar
Sona College of Technology, Salem, India
e-mail: rajkumar.s@sonatech.ac.in

P. Saraswathi
IT, Velammal College of Engineering and Technology, Madurai, India

R. Radha
EEE, Study World College of Engineering, Coimbatore, India

S. Suresh
CSE, KPR Institute of Engineering and Technology, Coimbatore, India

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2023

S. Shakya et al. (eds.), *Fourth International Conference on Image Processing and Capsule Networks*, Lecture Notes in Networks and Systems 798,
https://doi.org/10.1007/978-981-99-7093-3_5

method can classify airway diseases of step obstruction and non-obstructive localization. The proposed simulation result achieves high efficiency in respiratory disease prediction. In addition, respiratory diseases can be diagnosed by several performance measures such as sensitivity, specificity, accuracy, and $F1$-score.

Keywords Respiratory · Lung disease · Deep learning · DSTLM · Z-score normalization · LBGF · Classification · Accuracy · $F1$-score

1 Introduction

Respiratory diseases are a significant contributor to both illness and death rates. Then, they represent a considerable burden and cause changes globally. Chronic respiratory diseases significantly impact patients worldwide, with 7.4% of the global population affected and 7.0% of all deaths attributed to these conditions [1]. The healthcare sector is a unique and indispensable industry compared to other industries. The profession that demands the highest quality of diagnosis, treatment, and service is the most crucial area where people spend their money. One of the primary factors contributing to the accelerating impact of diseases on human health is the effects of environmental changes, climate change, lifestyle, and other related risks.

The images and sounds produced by different medical devices may be limited due to their subjective nature, complexity, and level of intelligence. Medical image and audio analysis can improve the accuracy of predictive modelling and diagnose diseases immediately. Due to the shortage of trained personnel, healthcare workers welcome technological aids to help handle more patients. The impact of respiratory diseases is increasing, and apart from significant health disorders like cancer and diabetes, it also threatens social life [2, 3].

However, respiratory diseases pose a significant and persistently high burden on global health systems. Diagnosing this condition can be challenging, particularly in remote or resource-constrained settings where the required diagnostic tools may not be available. Respiratory illnesses such as colds and flu can cause mild discomfort and incapacitation. On the other hand, conditions like pneumonia, tuberculosis, and lung cancer pose a serious threat to one's life as they lead to severe respiratory diseases [4].

Instead, millions of people suffer from pneumonia. They affect millions of people and threaten healthcare systems. Globally, respiratory disease significantly contributes to mortality and disability [5].

In this section, we first collected a dataset of respiratory sounds, a key indicator of respiratory health and disease, from the Kaggle dataset. Then, we perform preprocessing steps using wavelet smoothing and Z-score normalization to ensure a smooth and low-noise signal in respiratory lung disease images. Next, we use local binary Gabor filter (LBGF) models to extract features from respiratory lung disease. Finally, obstructive and non-obstructive respiratory diseases can be classified based on airflow limitations using the proposed DSTLM methods. Thus, the proposed

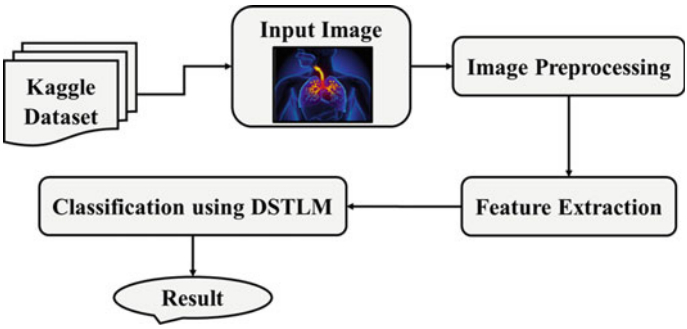


Fig. 1 Basic architecture of respiratory lung disease

method can improve the performance of pulmonary respiratory disease images and increase their application accuracy.

This section explores their applications through the basic structure of respiratory diseases as constructed in Fig. 1. In these, we first collected data from the Kaggle dataset and provided input images. Then, we pre-processed the respiratory disease images and extracted their features. Finally, we can find and obtain the accuracy of obstructive and non-obstructive artefact limits for respiratory disease using the proposed method.

2 Related Work

According to the author, asthma and COPD can be prevented, diagnosed, and treated using decision-support systems in clinical situations. These disorders can be diagnosed by identifying critical factors by conducting empirical breath studies on a representative sample [6].

The author discusses the effectiveness of three DT methods—packing, AdaBoost, and random forest—in predicting the LOS for children with respiratory diseases. The forms can also be used to monitor and treat patients based on data collected from a dataset of 11,206 records from hospital information structures. The capacity expansion technique can be adopted and analysed after pre-processing and computational transformation [7].

The author suggests that Wi-Fi CSI data can be used to track and extract human breathing motion. Through demonstrations, COTS models can be used to implement end-to-end systems composed of hardware platforms, control software, data acquisition algorithms, and processing algorithms [8]. The author proposed that a DL model can be developed and used to evaluate sleep states and respiratory disease events. The AHI score and subsequent calculations can only use the pulse oximetry data set to handle these [9].

The author proposed that the Mel-spectrogram-based deep CNN-RNN model can be applied for breath sound classification. The patient data that has been defined can be utilized to create personalized classification models for screening individuals for respiratory illnesses and identifying any irregularities [10].

The author proposed that new techniques such as OST and deep ResNets can be used to identify breaths, bursts, and natural sounds. In addition, the provided OST raw breath sounds can be processed, and Resnet's OST spectrogram can be rescaled [11]. The author proposed that a DL-based approach could be used in the NLSD to research long-term mortality. A binary classification model can be employed using a 3D ResNet-centric neural network to classify non-lung cancer mortality (cardiovascular and respiratory mortality) [12].

The author proposes using three classification techniques to implement a hybrid CNN model. Among them, the classification technique uses FC layers to classify CXR images and can collect weights with high classification accuracy with a model trained over multiple epochs [13]. The author proposed that non-contact auscultation could handle a DL-based model for lung disease classification. Two pulmonary specialists in respiratory medicine recorded the typical sounds of healthy breathing and five diverse sounds indicative of lung disease [14].

The author proposed that the DL auscultation analysis framework can be used to classify respiratory cycle abnormalities and diagnose disease from respiratory recordings. The framework converts the input speech into a spectrogram representation and starts with front-end feature extraction [15]. The author presents a low-cost PoC method for quantifying active NE in patient sputum using LFA [16]. The author proposed that a lightweight CNN framework can be implemented to utilize map-based composite volumetric lung sound features and classify respiratory diseases based on individual respiratory cycles. This feature offers the capabilities of EMD and CWT [17].

The authors proposed that breath signature analysis from RGB infrared sensors could be used to screen mask wearers' health status for non-contact techniques. In addition, respiratory data collection for mask wearers is functionally saturated in technology [18]. The author aims to obtain the length of the time-regression of the relationship between exposure and response. A novel method for detecting infectious respiratory diseases from human axial sweat samples using an electronic nose (e-nose) is being developed [19].

The author proposed that these can be used to derive regression lengths in exposure–response relationships. Two conventions are commonly used when exploring the connection between disclosure and response: linear and nonlinear. Both confounder- and confounder-free models can be implemented based on these assumptions [20]. The author proposed a model for class 2 and 5 COPD classification using a framework for learning representations of Mel-spectral slices. The method suggested involves pre-processing and data augmentation, creating Mel-spectrogram segments from lung sounds, and refining the pre-existing YAMNet [21].

The author notes that technological advances in USE lung diagnostic studies can be reviewed in detail. It presents USE's rationale and practical strategies and provides information on its various uses in evaluating lung disease [22]. The author suggests

exploring the significance and worth of utilizing the FOT scheme when evaluating lung cancer. Different levels of heterogeneity in lung tissue can be recognized through specialized equipment and related methods [23].

2.1 Problem of Statement

1. The risks of contamination and complications with traditional respiratory disease prediction techniques
2. The use of the deep short-term long memory method for improved accuracy in respiratory disease prediction
3. Details about the collection and pre-processing of the dataset used in this study from the Kaggle repository
4. A closer look at feature extraction using the local binary Gabor filter (LBGF) for detecting spatial patterns and grayscale variation in images.
5. Discussion on performance measures such as sensitivity, specificity, and accuracy.

3 Proposed Methodology

In this section, we first used the Kaggle dataset to collect information on pulmonary respiratory disease. We collected data from the Exasens dataset of breath sounds, a key indicator of health and disease in respiratory diseases. Then, wavelet smoothing and *Z*-score normalization methods can be used to pre-process respiratory disease images to smooth the images and ensure low signal-to-noise. Next, we use LBGF models to extract the features of these images. These methods can detect an image's local spatial patterns and grayscale variation efficiently. Lastly, the proposed E-LSTM method can detect obstructive and non-obstructive respiratory disease artefacts. Furthermore, based on DL methods, the DSTLM proposed method can recover the efficacy and accuracy of respiratory lung disease applications.

As shown in Fig. 2, the architecture diagram for respiratory disease using the proposed method is clearly explained in this section. First, we collect the Exasens dataset of breath sounds from the Kaggle dataset to collect respiratory disease data.

Then, pre-processing can be done using wavelet smoothing and *Z*-score normalization methods. Next, LBGF methods can be applied to feature extraction to efficiently detect spatial patterns and grey-scale variation in an image efficiently. Then, the proposed DSTLM method can detect the airflow limitations of obstructive and non-obstructive respiratory diseases to improve efficiency and increase accuracy.

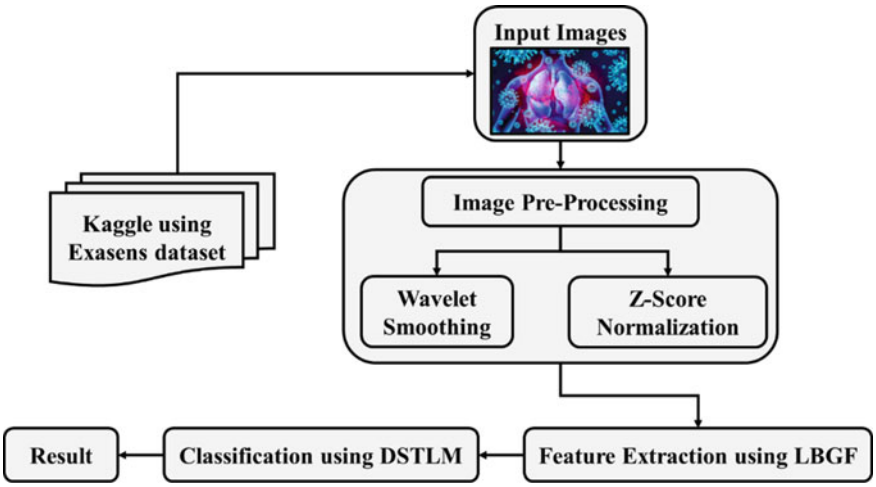


Fig. 2 Proposed architecture diagram (CMSFS-ARNN)

3.1 Dataset Collection

In this section, we collect data for pulmonary respiratory disease from the Kaggle dataset. Additionally, this data has introduced a novel dataset for classifying four respiratory conditions: COPD, asthma, infectious diseases, and healthy controls (HC). The Exasense dataset can collect demographic data from four saliva sampling partners as part of a collaborative project at the Borsdell Research Center at the Northern Biomaterials Bank.

3.2 Image Pre-processing

This section deals with the pre-processing steps after collecting the pulmonary respiratory disease information from the Kaggle dataset. Then, respiratory disease images can be smoothed and used to ensure low signal-to-noise by pre-processing lung diseases using wavelet smoothing and *Z*-score normalization methods on respiratory disease images. These methods may be disturbed by acoustic noise from ambient noise, ambient speech, electronic interference or stethoscope drift, lung sound recordings, and other electronically recorded biosignals. Therefore, it is necessary to ensure the signal is smoothed and pre-processed before the feature extraction process. Following two processes for image smoothing and normalization,

a. Wavelet smoothing
b. *Z*-score normalization.

3.2.1 Wavelet Smoothing

In this respect, wavelet transforms (WT) are widely used to smooth transient signals. Then, instead of using the Fourier transform (FT) to decompose groups of waves, these can be decomposed instead of the signal's complex sinusoids. By converting the time and frequency domains of a signal, useful information can be extracted or noise can be removed. Next, a fundamental frequency can be used as the simulation unit to preserve the original data and decompose a signal into higher and lower frequencies using WT.

This section estimates the signal translated into WT by multiplying the background noise (Eq. 1).

$$\varphi_{uv}(s) = |u|^{-1/2}\varphi\left(\frac{s-v}{u}\right) \tag{1}$$

This category calculates signal smoothing as a time-domain and frequency-domain transformation (Eq. 2).

$$A_m = w_m + z_m \tag{2}$$

Equation 3 is a continuous wavelet transform that computes various frequencies across different sizes and positions.

$$s(u,v) = \frac{1}{\sqrt{u}}\int_{-\infty}^{\infty} A(s)\psi^*\frac{(s-v)}{u}cs \tag{3}$$

Equation 4 is the evaluated discrete wavelet transform that uses finite wavelets at different magnitudes and levels.

$$s_{nm} = \int_{-\infty}^{\infty} A(s)\varphi_{nm}(s)cs \tag{4}$$

Calculate the infinite sequence using high-pass and low-pass filters (Eqs. 5 and 6).

$$\overline{H}_{rs} = \sum_{o=0}^{or-1} \overline{q}_{r,o}A_{s-o} \tag{5}$$

$$\overline{G}_{rs} = \sum_{o=0}^{or-1} \overline{e}_{ro}A_{s-o} \tag{6}$$

Let us assume an s—time instance, u—dilation function, v—translation function, $(\varphi(s))$ wavelet translate and dilate signal, m and n—signal noise, w—real signal, z—noised signal, r—level of the decomposition A_{s-o} infinite sequence, $\overline{H}_{rs}$ wavelet coefficient, $\overline{G}_{rs}$ rs—scaling coefficient, and $\overline{q}_{r,o}$ and $\overline{e}_{ro}$ high-, and low-pass filter. In this area, the signal can be estimated by multiplying the background noise. Then, the signal smoothing can be calculated as a time-domain and frequency-domain transform using continuous wavelet and discrete wavelet transforms.

3.2.2 *Z*-Score Normalization

In this category, it is necessary to ensure that each signal is normalized by its *Z*-score after the above signal pre-processing step. *Z*-score normalization increases the dynamic range between the corresponding values in the time-domain signal. In other words, significant trends in the signal do not necessarily dominate small movements.

This model Eq. 7 calculates the normalization for raw score converting.

$$G = \frac{G - \mu}{\sigma} \tag{7}$$

Equation 8 gives the standard deviation of the score distribution and can be calculated.

$$\sigma = \sqrt{\frac{\sum_{j=1}^{m} \left(g_j - \overline{g}\right)^2}{m}} \tag{8}$$

Estimate the normalized values according to the attribute's mean and standard deviation (Eq. 9).

$$A_j = \frac{B_j - \mathrm{AVG}(X)}{X} \tag{9}$$

Let us assume the g-raw score and μ—signal exhibits a mean value, σ—standard deviation, m—total number, j—score, $\overline{g}$—average score, X-attribute, A_j—value of the attribute, and B_j—normalized value. To calculate the standard deviation of the distribution, we normalize the raw scores as shown in the equation. A normalized value is also estimated based on the attribute's mean and standard deviation.

3.3 Local Binary Gabor Filter

This section uses important features extracted from images to solve specific problems. In addition, these methods can efficiently detect local spatial patterns and grey-scale variation in pulmonary respiratory dog images. Features can be particular

structures in an image, such as points, edges, colours, sizes, shapes, and objects. Logically, image type affects feature quality. Although these new features may differ from the original ones, they might exhibit a more vital ability to distinguish in various locations than the current site. Feature extraction aims to enhance the effectiveness of machine learning algorithms for object recognition by generating more valuable features, including through methods such as developing binary Gabor filters. The LBGF represents the size relationship between a central pixel and its neighbouring pixels in a neat form. The LBGF transform is part of the windowed Fourier transform, which uses the LBGF function to extract features related to various sizes and orientations in the frequency domain.

This category, Eq. 10, estimates the response of a spatially-based 2D Gabor filter.

$$A_j(a, b; \lambda, \varphi, \sigma_a, \sigma_b) = \frac{1}{2\pi\sigma_a, \sigma_b} \exp\left\{-\frac{1}{2}\left[\frac{p_1^2}{\sigma_a^2} + \frac{p_2^2}{\sigma_b^2}\right]\right\} \tag{10}$$

Compute the image transformation using the two-dimensional Gabor function expressed in Eqs. (11) and (12).

$$Z(a, b) = \iint \delta J(\alpha, \beta) H(a - \alpha, b - \beta) c\alpha c\beta \tag{11}$$

$$H_{\lambda,\theta,\varphi,\sigma b^{a,b}} = \exp\left[\frac{\left(a^2 + \gamma^2 + b^2\right)}{2\sigma^2}\right] \cos\left[2\pi\frac{a}{\lambda} + \varphi\right] \tag{12}$$

In this section, the imaginary part of the Gabor energy filter can reflect the gray level changes in different directions. (Eq. 13).

$$E(a, b, \theta) = \sqrt{((E^* g)(a, b, \theta, 0))^2 ((E^* g)(a, b, \theta, \pi))^2} \tag{13}$$

Compute the LBP histogram generated from the smallest values of all micro-patterns (Eq. 14).

$$X_j(s) = \begin{cases} 1, \ j(s) < J\left(s + \Delta r_j\right), \\ 0, \ \text{otherwise} \end{cases} \quad (j = 1, \ldots, M_m) \tag{14}$$

Each LBP is transformed into a vector to calculate co-occurrences efficiently in this category (Eq. 15).

$$E_j(s) = \delta_{jl}(y(s)) \tag{15}$$

This category indicates the number of adjacent LBP pixel sizes (Eq. 16).

$$G(a) = \sum_{\delta=J} E(s) E(s + x)^t \tag{16}$$

Let us assume a and b—function, $Z(a, b)$—Gabor residuals, (α, β)—convolution of the image, E—Gabor function frequency of filter centre, δ—set of image points, θ-theta, and filter direction, α, β—alpha and beta, λ—lambda, γ—Gama, 0 or π—phase offset, M_m—no of the neighbor pixel, $y(s)$—label, $G(a)$—element, a—displacement vector. In this section, the size relationship between a central pixel and its neighbouring pixels can be represented in an elegant form, and the LPGF method can be used to extract features related to different sizes and orientations in the wave domain.

3.4 Deep Short-Term Long Memory (DSTLM)

In this section, the proposed DSTLM method can be used to classify obstructive and non-obstructive respiratory diseases to determine the accuracy of pulmonary respiratory illness. A technique utilizing DSTLM has been introduced that can accurately classify pulmonary diseases while requiring minimal training time. During a network's operation, temporary information is managed by cells, while in and outgoing information is controlled by gates. The proposed DSTLM method networks can process data in forward (one-way) or forward and reverse order. In signal analysis, the latter is the preferred option and is crucial for timely learning. A DSTLM network can help predict lesions better by presenting clinical relationships between lesions.

Algorithm
Input: Number of adjacent pixel size G
Output: Prediction result
Start
Step1: Calculate the output of the core function cell

$$G_w = E_w G_{w-1} x_w G_w \quad (17)$$

Step 2: Calculate the gate entered in the result central cell

$$D_w = Q_w \tan D(G_w) \quad (18)$$

Step 3: The functions of the hidden units selected to operate in the network can be calculated based on the sigmoid function

$$x_w = \sigma(z_{Ux} U_w + z_{Dx} D_{w-1} + z_{Gx} G_{w-1} + a_x) \quad (19)$$

$$E_w = \sigma(z_{UE} U_w + z_{DE} D_{w-1} + z_{GE} G_{w-1} + a_E) \quad (20)$$

$$Q_w = \sigma\left(z_{UQ} U_w + z_{DQ} D_{w-1} + z_{GQ} G_{w-1} + a_Q\right) \quad (21)$$

Step 4: Calculate the limitations of the output's forward and backward direction artifacts

$$v_w = z_{\overrightarrow{Dv}} \overrightarrow{D^M} + z_{\overleftarrow{Dv}} \overleftarrow{D^M} + a_v \quad (22)$$

Return v_w
Stop

Let us assume G—functional cell, w—time, E_w—target gate activation, x_w—input gate activation, G_w—input to the central compartment, σ—sigmoid function, Q_w—output gate activation, z—weight, z_U—input gate weight, z_D—hidden

to hidden weight, z_G—peephole weight, v_w—bidirectional output. In this category, DSTLM processes pre- and post-respiratory disease information using bilateral artifact thresholding of the production.

4 Analyses and Discussions of Experimental Results

This section enables Python to manipulate the Jupyter toolbox to detect airflow limitations in pulmonary respiratory diseases, enhance image quality, and achieve accurate results. For more information, see the project's Website at https://archive.ics.uci.edu/ml/datasets/Exasens.

Table 1 mentions the data set name, number, training, test time, and languages using their values and categories to achieve the accuracy of respiratory diseases.

4.1 Evaluation Matrix

In this section, we can evaluate the enactment of the developed model using several metrics for analysing classification confusion matrices. Based on the training/classification scheme, a confusion matrix array can be generated after each iteration. Calculating all evaluation metrics is as simple as counting the number of them in the confusion matrix. By identifying true positives, true negatives, false positives, and false negatives, accurate measurements of accuracy, sensitivity, specificity, precision, and $F1$-estimate can be obtained.

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{23}$$

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} \tag{24}$$

Table 1 Parameter model

Types	Values
Dataset name	Kaggle
No. of dataset	400
Training	180
Testing	75
Language	Python
Tool	Jupyter
Dataset name	Kaggle

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{25}$$

$$F1\text{-Score} = \frac{2\text{TP}}{2\text{TP} + \text{FP} + \text{FN}} \tag{26}$$

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{27}$$

In this section, the performance of pulmonary respiratory disease images can be evaluated using sensitivity analysis to determine their accuracy, as introduced in Fig. 3. Then dealing with the FOT, CNN, and EMD methods presented in the literature review, their accuracy is as low as 49%. Sensitivity analysis of the DSTLM performance of the proposed method with the previously submitted form achieves high accuracy.

Figure 4 suggests that a dataset of pulmonary respiratory disease images was obtained from Kaggle and analysed for specific performance to determine their accuracy. Detailed performance analysis of CNN, FOT, and EMD methods has achieved lower accuracy. Comparing these three methods with the proposed DSTLM method, their accuracy is increased to 70%.

In this section, they can be implemented using algorithmic analysis to determine the accuracy of respiratory disease images, as shown in Fig. 5. Then, the accuracy performance of the proposed DSTLM method increased to 77% over analysis. Then, the three methods EMD, CNN, and FOT, presented in the literature analysis, have 59% lower performance accuracy than the proposed method.

In Fig. 6, $F1$-score performance is used to determine the accuracy of the proposed method for detecting respiratory diseases. A comparison of the EMD, FOT, and CNN

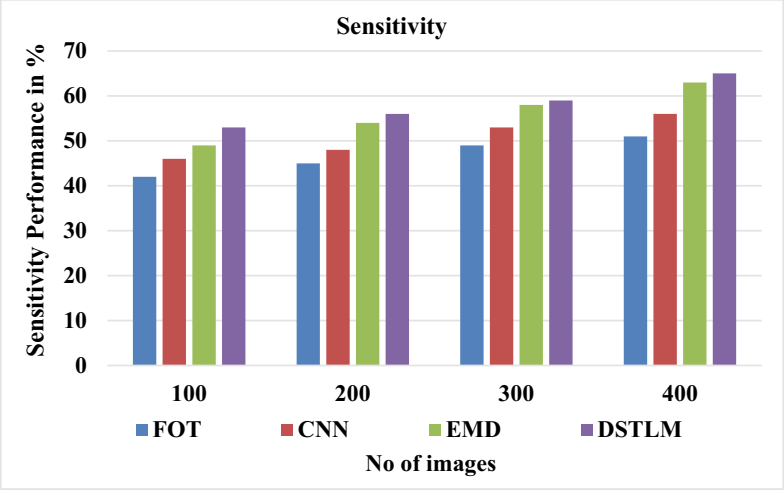


Fig. 3 Analysis of the sensitivity performance

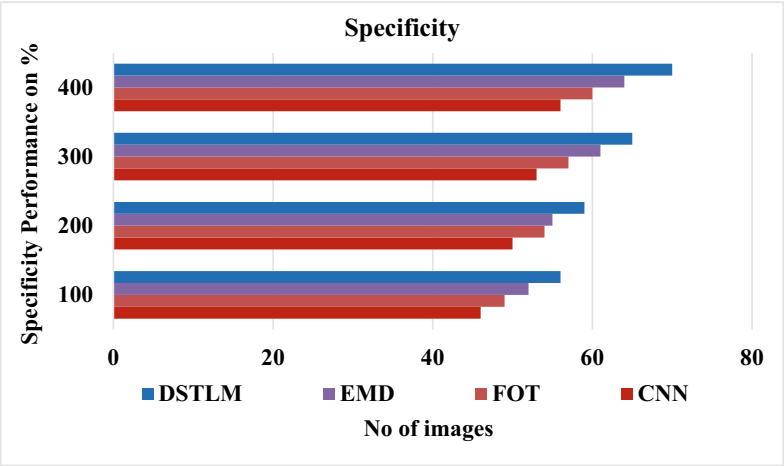


Fig. 4 Analysis of the specificity performance

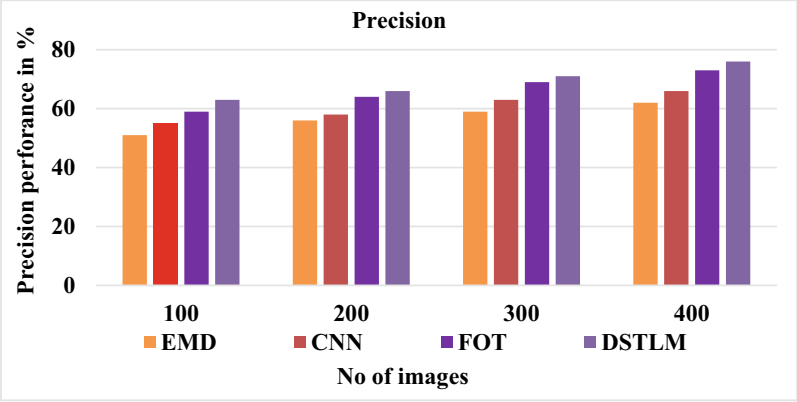


Fig. 5 Performance of the precision

models presented in the literature was done before determining the performance of the proposed method. It is necessary to increase the accuracy of their performance value. Then, the $F1$-score value of the proposed DSTLM method with these three methods increased to 83% with high accuracy.

In Fig. 7, image accuracy is improved by analysing the proposed method and the literature method presented. Thus, the proposed DSTLM and the presented literature methods provide a systematic classification and identify them with high accuracy. The classification accuracy of the DSTLM method is increased to 90% compared to the other three methods.

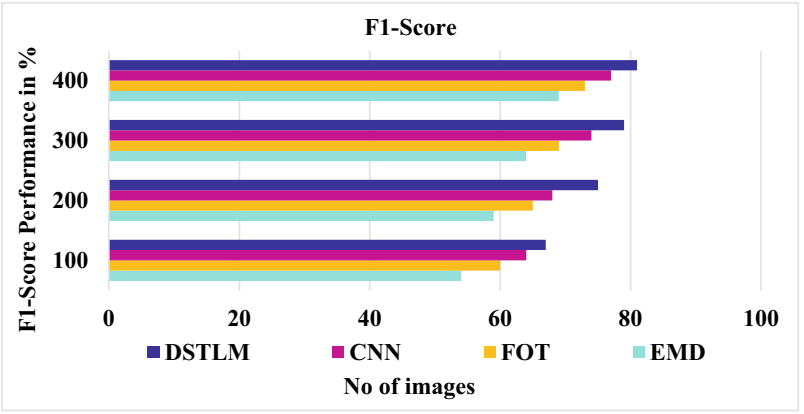


Fig. 6 Analysis of the *F*-score

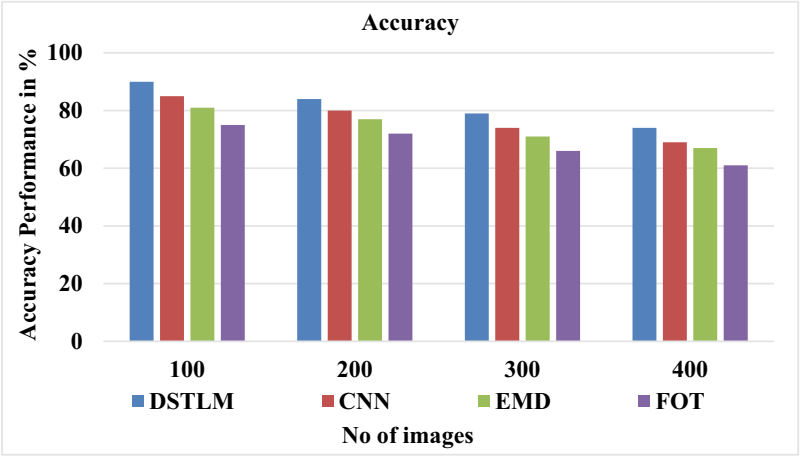


Fig. 7 Performance of the accuracy

5 Conclusion

This section uses a DL model based on DSTLM to classify pulmonary respiratory diseases. Then, based on this finding, a state-of-the-art DL-assisted lung and respiratory diagnosis taxonomy can be developed by summarizing the key concepts and features of current applications of DL in pulmonary respiratory disease diagnosis and their characteristics. This model differs from methods used to diagnose lung and respiratory diseases. Using LBGF feature extraction and wavelet smoothing with *z*-score normalization, only lung-specific features with appropriate image enhancement were considered. We proposed the DL model DSTLM to improve classification

performance with minimum computation. We can use the proposed method to analyse the data using sensitivity, specificity, precision, recall, and accuracy measurements. The model was used to classify lung sounds into several pulmonary diseases with an average accuracy of 90%. This improves the dependability of the model being suggested. The performance metrics of classification models are impressive, but there is room for improvement by adjusting the pre-processing and training techniques. Future directions are indicated, and there are new techniques being developed that could potentially improve performance and expand the use of lung respiratory disease applications.

References

1. Wang C, Ma J, Zhang S et al (2022) Development and validation of an abnormality-derived deep-learning diagnostic system for major respiratory diseases. NPJ Digit Med 5:124
2. Bharati S, Podder P, Rubaiyat Hossain Mondal M (2020) Hybrid deep learning for detecting lung diseases from X-ray images. Inform Med Unlock 20:100391. ISSN 2352-9148
3. Srivastava A, Jain S, Miranda R, Patil S, Pandya S, Kotecha K (2021) Deep learning based respiratory sound analysis for detection of chronic obstructive pulmonary disease. PeerJ Comput Sci 11(7):369
4. Jasmine Pemeena M, Priyadarsini KK (2023) Lung diseases detection using various deep learning algorithms. 2023:3563696
5. Sakthivel S (2015) Secure data storage auditing service using third party auditor in cloud computing. Int J Appl Eng Res 10(37)
6. Spathis D, Vlamos P (2019) Diagnosing asthma and chronic obstructive pulmonary disease with machine learning. Health Inf J 25(3):811–827
7. Ma F, Yu L, Ye L, Yao DD, Zhuang W (2020) Length-of-stay prediction for pediatric patients with respiratory diseases using decision tree methods. IEEE J Biomed Health Inform 24(9):2651–2662
8. Mosleh S, Coder JB, Scully CG, Forsyth K, Kalaa MOA (2022) Monitoring respiratory motion with Wi-Fi CSI: characterizing performance and the BreatheSmart algorithm. IEEE Access 10:131932–131951
9. Kiruthiga G et al (2022) Effective DDoS attack detection using deep generative radial neural network in the cloud environment. In: 2022 7th international conference on communication and electronics systems (ICCES), pp 675–681
10. Acharya J, Basu A (2020) Deep neural network for respiratory sound classification in wearable devices enabled by patient-specific model tuning. IEEE Trans Biomed Circuits Syst 14(3):535–544
11. Chen H, Yuan X, Pei Z, Li M, Li J (2019) Triple-classification of respiratory sounds using optimized S-transform and deep residual networks. IEEE Access 7:32845–32852
12. Karthick K et al (2023) Iterative dichotomiser posteriori method-based service attack detection in cloud computing. Comput Syst Sci Eng 44(2):1099–1107
13. Yaseliani M, Hamadani AZ, Maghsoodi AI, Mosavi A (2022) Pneumonia detection proposing a hybrid deep convolutional neural network based on two parallel visual geometry group architectures and machine learning classifiers. IEEE Access 10:62110–62128
14. Choi Y, Choi H, Lee H, Lee S, Lee H (2022) Lightweight skip connections with efficient feature stacking for respiratory sound classification. IEEE Access 10:53027–53042
15. Pham L, Phan H, Palaniappan R, Mertins A, McLoughlin I (2021) CNN-MoE based framework for classification of respiratory anomalies and lung disease detection. IEEE J Biomed Health Inform 25(8):2938–2947

16. Naveenkumar E et al (2022) Detection of lung ultrasound covid-19 disease patients based convolution multifacet analytics using deep learning. In: 2022 second international conference on artificial intelligence and smart energy (ICAIS), pp 185–190
17. Shuvo SB, Ali SN, Swapnil SI, Hasan T, Bhuiyan MIH (2021) A lightweight CNN model for detecting respiratory diseases from lung auscultation sounds using EMD-CWT-based hybrid scalogram. IEEE J Biomed Health Inform 25(7):2595–2603
18. Jiang Z et al (2020) Detection of respiratory infections using RGB-infrared sensors on portable device. IEEE Sensors J 20(22):13674–13681
19. Naveenkumar E et al (2022) Lung Ultrasound COVID-19 detection using deep feature recursive neural network. Intelligent sustainable systems. Lecture notes in networks and systems, vol 458
20. Lu J, Bu P, Xia X, Yao L, Zhang Z, Tan Y (2020) A new deep learning algorithm for detecting the lag effect of fine particles on hospital emergency visits for respiratory diseases. IEEE Access 8:145593–145600
21. Roy A, Satija U (2023) A novel melspectrogram snippet representation learning framework for severity detection of chronic obstructive pulmonary diseases. IEEE Trans Instrum Meas 72:1–11
22. Zhou B, Yang X, Zhang X, Curran WJ, Liu T (2020) Ultrasound elastography for lung disease assessment. IEEE Trans Ultrason Ferroelectr Freq Control 67(11):2249–2257
23. Ghita M, Billiet C, Copot D, Verellen D, Ionescu CM (2023) Parameterisation of respiratory impedance in lung cancer patients from forced oscillation lung function test. IEEE Trans Biomed Eng 70(5):1587–1598

Utilizing Satellite Imagery for Flood Monitoring in Urban Regions



Priyanka Sakpal, Shashank Bhosagi, Kaveri Pawar, Prathamesh Patil, and Pratham Ghatkar

Abstract According to the report, billions of people are affected by floods as it is a natural hazard caused by heavy rainfall or glacier melting. Floods occur when there is an overflow of water bodies like rivers, lakes, or oceans, in which there is a heavy flow of water that enters cities and villages causing lots of damage to homes and businesses. To decrease the impact of floods and decrease the loss of life and property, it is important to use the technology available to us to detect the flood-affected areas. But before using the technologies, extraction of water bodies from remote sensing areas is an important task that is mostly done by using a deep learning network for extraction of water bodies. In this paper, we have surveyed the previous research papers which talk about various techniques to predict the condition of water bodies from satellite-captured images and further use it to detect floods. In this paper, we have written the survey table about the accuracy of the techniques used in the paper and a summary of that paper.

Keywords Flood monitoring · Synthetic aperture radar · Normalized difference water index and Sentinel · Convolutional neural network · Dence local feature compression · VGG-Net · Dense-Net

1 Introduction

In recent years, floods have become increasingly frequent and devastating, causing significant economic and humanitarian crises around the world. Effective monitoring and timely response to flood events are critical to mitigating their impact and protecting vulnerable communities. One promising approach in flood monitoring is

P. Sakpal (✉) · S. Bhosagi · K. Pawar · P. Patil · P. Ghatkar
JSPM's Rajarshi Shahu College of Engineering, Pune, Maharashtra 411033, India
e-mail: pssakpal_comp@jspmrscoe.edu.in

S. Bhosagi
e-mail: srbhosagi@jspmrscoe.edu.in

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2023

S. Shakya et al. (eds.), *Fourth International Conference on Image Processing and Capsule Networks*, Lecture Notes in Networks and Systems 798,
https://doi.org/10.1007/978-981-99-7093-3_6

the utilization of satellite images, which offer valuable data for understanding and predicting flood dynamics over large areas.

Satellite imagery provides a comprehensive and synoptic view of the Earth's surface, enabling the detection and mapping of flood-affected regions. By leveraging the advancements in remote sensing technology, flood monitoring using satellite images has witnessed remarkable progress. This survey paper aims to provide a comprehensive overview of the state-of-the-art techniques and methodologies employed in flood monitoring using satellite images.

The objective of this survey is threefold:

First, to review the fundamental principles of remote sensing and satellite image analysis as they pertain to flood monitoring. This includes an examination of different satellite sensors, such as optical, synthetic aperture radar (SAR), and multispectral sensors, and their capabilities in capturing flood-related information. Furthermore, we will explore various image pre-processing techniques, such as radiometric and geometric corrections, and image enhancement methods specific to flood monitoring.

Second, to investigate the different approaches for flood detection and mapping using satellite images. This section will delve into the algorithms and methodologies employed for flood detection, including pixel-based, object-based, and machine learning techniques. Additionally, we will explore the challenges and limitations associated with flood detection, such as cloud cover, image resolution, and image classification accuracy.

Third, to highlight the applications and case studies where flood monitoring using satellite images have been successfully utilized. We will discuss how satellite-based flood monitoring contributes to flood risk assessment, early warning systems, and post-disaster management. Moreover, we will explore the integration of satellite data with other data sources, such as hydrological models and geographic information systems (GIS), to enhance flood monitoring and response capabilities.

By examining the existing literature and advancements in the field, this survey paper aims to provide a comprehensive understanding of flood monitoring using satellite images. The knowledge and insights gained from this survey will not only benefit researchers and practitioners but also contribute to the development of more accurate and efficient flood monitoring systems in the future. Ultimately, the effective utilization of satellite imagery can play a pivotal role in minimizing the impact of floods and safeguarding communities worldwide. Flood causes lots of damage to human life and the economy, according to the report published by the Centre for Research on the Epidemiology of Disasters, flood affected about 1 billion people all over the world from 2001 to 2010, and overall damage is 142 billion dollars. Google Earth Engine provides data for flood mapping which saves time in gathering information and accelerates image processing. High-resolution satellite photos are used to verify the effectiveness of the applied strategy.

Many techniques can help us in the detection of floods by using satellite images so that we can efficiently classify flooded and non-flooded areas. Satellites provide a cost-effective and rapid resource for mapping and monitoring the scope of ongoing flood occurrences in real time. Since the 1970s, numerous studies on the mapping

and monitoring of floods using evidence from various triggers aboard different satellites have been published in the scientific literature. “Landsat”, “TRMM”, “GPM”, “Terra”, “Aqua”, and “GRACE” are a few of the satellites frequently used for flood monitoring and forecasting. The delay between data gathering and picture scattering to the user on the base is short for “RADARSAT” and “ENVISAT.”

2 Related Work

In the year 2009, research conducted by Martinis et al. [1] developed an approach based on fuzzy set theory to automatically map flooded areas from multispectral S2 MSI pictures in this work. It derives its evidence scores from two band combinations: (1) Spectral index and (2) Element of hue, saturation, and value color transformation. The results prove that performance is better than earlier single-input feature segmentation algorithms, with more consistent and strong accuracy.

In the year 2013, research conducted by Giustarini et al. [2] hybrid approach integrating backscatter thresholding, region growth, and change detection is detailed in this paper as a method for obtaining flood extent from very high-resolution urban SAR images in an automated and repeatable manner. It is based on a statistical distribution of open-water backscatter values derived from flood images. The proposed methodology was shown to be 81.7% accurate.

Mason et al. [3] 2014 conducted research on the double scattering approach, utilized in this paper to detect flooded metropolitan areas using SAR pictures. To assess double scattering durability, the method uses a SAR image and a high-resolution LADAR map of the metropolitan region. This strategy is 100% accurate in flooded areas and 91% accurate in non-flooded areas.

Verwaeren et al. [4] 2015 researched the speckle uncertainty in a flood mapping approach and the bootstrap method which are employed in this paper. Using an image segmentation method, different speckle patterns were found, and speckle was added to the segment mean backscatter value to create fake flood images. F-values of 0.50 and 0.65 indicate the accuracy of two test instances.

In the year 2015, Kersten et al. [5] studied how to detect the flood crisis in real time, a Terra SAR-X-based flood service is paired with a completely automated processing chain in this research. SAR data pre-processing, computation, and change of overall additional data and unsupervised classification utilizing fuzzy logic-based algorithms are all part of the processing chain. The proposed methodology was tested in two locations: Thailand and Germany, yielding 98.5% flood and 71.9% non-flood areas, and 95.37% flood and 83.52% non-flood areas, respectively.

In the year 2016, Luca et al. [6] created a prototype system and offered a mapping algorithm for real-time flood conditions. To identify the flood, this system was evaluated utilizing COSMO-SkyMed and Sentinel-1 data. The precision was calculated to be 98%.

Renaud et al. [7] in the year 2016 investigated research in which they used the data of synthetic aperture radar and they have also used a probabilistic flood mapping

approach. A Bayesian technique was applied in this case. The likelihood of a given pixel was calculated by using the readings taken from a histogram of backscatter.

Chapi et al. [8] 2017 proposed two models for mapping flood susceptibility, bagging-LMT—a combination of bagging ensemble and logistic model tree. Here the bagging-LMT model outperforms the logistic model tree, logistic regression, Bayesian logistic regression, and random forest models. An accuracy of 99.3% was achieved for the training dataset and an accuracy of 95.5% on the dataset for testing the bagging-LMT model.

In the year 2018, Notti et al. [9] present a semi-automated mapping of flood based on the available open-source satellite images. Various techniques are used to detect flooded areas utilizing MODIS, Proba-V, Landsat, and Sentinel-2, multispectral satellite data. To detect places that have been flooded, semi-automatic extraction is used. After that, utilizing auxiliary data web-based data and data such as the digital elevation model by using this map of the flood is manually purified.

Wang et al. [10] in the year 2018 studied the SRFIM-MSI approach using the normalized difference water index to extract spectral information from the multispectral band. Two Landsat 8 OLI multispectral data were used to assess the proposed technique, yielding an overall accuracy of 96.81%.

Murphy et al. [11] 2018 conducted research on deep convolutional and recurrent neural networks for the semantic segmentation of UAV imagery. More correct, expressive, and meaningful division maps were obtained using an end-to-end integration network and also used the combination of RNN and CNN networks.

In the year 2018, Bioresita et al. [12] investigated a completely automated chain of processing for rapid flood and surface water mapping. The main goal is to see if Sentinel-1 data would be used for flood detection and surface water mapping, as well as to develop an automated surface water extraction chain processing system. The accuracy level reached was 98.9%.

Martinis et al. [13] in the year 2018 study on a system for detecting changes automatically based on data from Sentinel-1 SAR. A Jensen–Shannon divergence-based index is used to choose the reference picture from a pool of possible image candidates. Second, using log-ratio data, saliency detection is utilized to determine the prior probability of impacted and unaffected classes.

In the year 2019, Li et al. [14] researched the multitemporal TerraSAR-X data to look at the contribution of the intensity of SAR and interferometric coherence in flood detection in urban areas. In this structure, the multitemporal intensity is essential.

Rudner et al. [15] 2019 proposed the rapid segmentation approach for flooded areas by fuzzy multiresolution, multisensory, and multitemporal satellite images by using multitemporal satellite images. It is very easy to differentiate between flooded areas and other sections. It is fundamental to the larger problem of automated maps from satellite imagery. These maps are useful for tasks like disaster management or locating eligible rooftops for solar panels.

Nallapareddy et al. [16] in the year 2019 studied thresholding and supervised classifications methods to classify flooding areas. The utilization of estimating area

data that is considered typical of each rock type to be categorized is referred to as supervised classification.

In the year 2019, Shen et al [17] studied near-real-time flooding maps created using satellite information. Here, NRT system called "Radar Produced Inundation Diary (RAPID)" involved four steps of classification based on multi-threshold and compensation, statistics, machine learning correction, and morphological processing.

Han et al. [18] 2019 researched on "unsupervised detection" of flood by using the "synthetic aperture radar" system for the data. The large data faces the problems like automatic thresholding as some data shows unimodal. For multi-scale chessboard segmentation-based tiles, there are statistical concepts such as bimodal Gaussian distribution and non-parametric histogram-based method.

Huang et al. [19] 2020 investigated how SAR is used to penetrate clouds and heavy rainfall. Sentinel-1 SAR image-set are used to cover the location and backscatter classification with the assistance of Sentinel-2 images using a supervised classifier. By using the same process, SAR pictures categorized the pre-flood and flooding duration to understand "change detection" by uniting them.

In the year 2020, DeVries et al. [20] studied SAR sensors which shows the essential data for flood disaster planning. As "Sentinel-1" SAR satellite data is freely available [GEE]. The authors used very high-resolution optical imagery with an accuracy of $89.8 \pm 2.8\%$ (95%).

Bonafilia et al. [21] in the year 2020 researched "Sen-1" floods 11used to justify the full "convolutional neural network" to parts the flood water. Differentiate the permanent flood and total surface water here using the 4 data divisions learners from training as "FCNN" as this dataset consists of 4831 512 * 512 chips coves 120,406 km^2 so it divides into four parts (1) 446 chips cover water surface from the event of flood (2) 814 chips for the permanent surface of water (3) 4385 chips of water surface classifies the "Sen-1" pictures from the flooding conditions.

In the year 2020, Li et al. [22] conducted research on dynamic water bodies. So to detect such water bodies, old dataset gives some limitations, but the advanced cloud-based "Google Earth Engine" platform with freely available data from "Sentinel-1" is very much useful.

Jain et al. [23] in the year 2020 studied on automatic identification of water from satellite by old methods whose accuracy is not that perfect, so to avoid a deep convolutional neutral network with the addition of water index-based solutions found. Here by combining the normalized difference water index with the blue/NIR spectral index, they found shallow water.

In the year 2020, Ross et al. [24] research on SAR used to find flooded areas. Data provided by the SAR is very effective for flood detection. First, here divide the region into two categories for example (a) monsoon region and (b) not rainy season. Then other steps of the algorithm work.

Hashemi-Beni et al. [25] in the year 2021 researched deep learning-based approaches used in flood detection with the help of a "convolutional-neutral-network" from the optical images. Using a digital electoral model, the region-expanding approach assesses the degree of flooding beneath vegetation that is difficult to observe from an imaginative perspective.

Meysam et al. [26] in the year 2021 studied flood events monitored by "Sentinel-1" images. The Otsu threshold method is used to separate flooded areas. Accuracy is more than 90%.

In the year 2021, Li et al. [27] conducted research on deep learning which used remote sensing images for the separation of water bodies from other areas. So the new network is used which is a dense local feature compression network. The authors created a new dataset called "GaoFen-2". Authors give good results by using "Sentinel-2 ZY-3 images" comparing old methods, and the accuracy of that network is above 90%.

3 Major Techniques Used

3.1 Ordered Weighted Averaging

Ordered weighted averaging is an aggregation technique between the logical "or" and "and" which allow for a combination of multiple inputs using weighted averages. Ordered weighted averaging is a step to assign weights to the model updates based on their performance. It is generally used in computational intelligence which is work as an optimized algorithm to train neural networks refers as the federated averaging algorithm. Federated averaging is an optimization algorithm in the field of federated learning, specifically designed for training neural networks. By performing local training on the dataset at the central server extracting features and identifying flood-related patterns. The model updates, in the form of gradients, are then aggregated using federated averaging. This process ensures that the shared model incorporates the collective knowledge and insights from the distributed devices. It contains three major components:

- Central Server: It coordinates the federated learning process. Here initialization of the model is done; it is used to manage the overall training process.
- Client Devices: These may be any smart device like IoT devices. It performs the model training locally using its data.
- Communication Protocol: Rules for communication between the central server and client devices are defined here. It ensures secure communication with data privacy about the model updates.

By using both techniques collaboratively, flood monitoring systems can benefit from the collective intelligence of the distributed satellite devices which may be any smart device like IoT devices while accounting for the varying performance levels or reliability of each device. This collaboration will create a more accurate and comprehensive model for flood detection and prediction.

The following steps are used which are as follows.

a. For each input attribute from estimating data, define data partitioning.

b. Define and choose the ordered weighting average that will be used in the soft integration. And apply federated averaging phases, such as model initialization, local training, model update, and communication and aggregation.
c. Soft-integration builds on pixels to find out the synthetic general evidence degree.
d. Segmentation of synthetic overall verification for creating maps of flooding areas based on the decision-maker mindset.

3.2 Spectral Indices

Spectral indices, mostly used in multispectral remote sensing areas, is a mathematical function that is the combination of pixel values from two or more than two spectral bands. It is applied on more than two reflectance bands of the image. It is based on the spectral properties of the object of interest like identifying the properties of the earth's surface.

The steps to create a map of flooding areas using spectral indices:

- Gather satellite data: Gather the data for flooded areas. To spot the changes in land cover and water presence, it is ideal to use photos taken before and after the flood event.
- Pre-process the imagery: To adjust atmospheric conditions, geometric distortions, and radiometric variances, pre-process the satellite images.
- Calculate spectral indices: The methods like normalized difference water index (NDWI), normalized difference vegetation index (NDVI), and modified normalized difference water index (MNDWI) are used to calculate the spectral indices.
- Thresholding and classification: After calculating the spectral indices, we should classify the flooded areas using the appropriate threshold values.
- Create a flooded area map: Construct a binary map labeling flooded areas as "1" and non-flooded areas as "0" to create the flooded area ap by applying the threshold values to the spectral index images.
- Post-processing and visualization: Refine the binary map by using morphological operations to eliminate artifacts and smooth the borders such as erosion and noise reduction of an image. Finally, for visualization and interpretation, place the map of the flooded area over a base map.

3.3 Region Growing

It is an image segmentation approach. It is also a pixel-based picture segmentation approach that necessitates seed point selection which is particularly used in image processing to recognize the features of an area of interest. A region grows from this seed point and segments the image. Figure 1 is the image that is used to depict the region's growth.

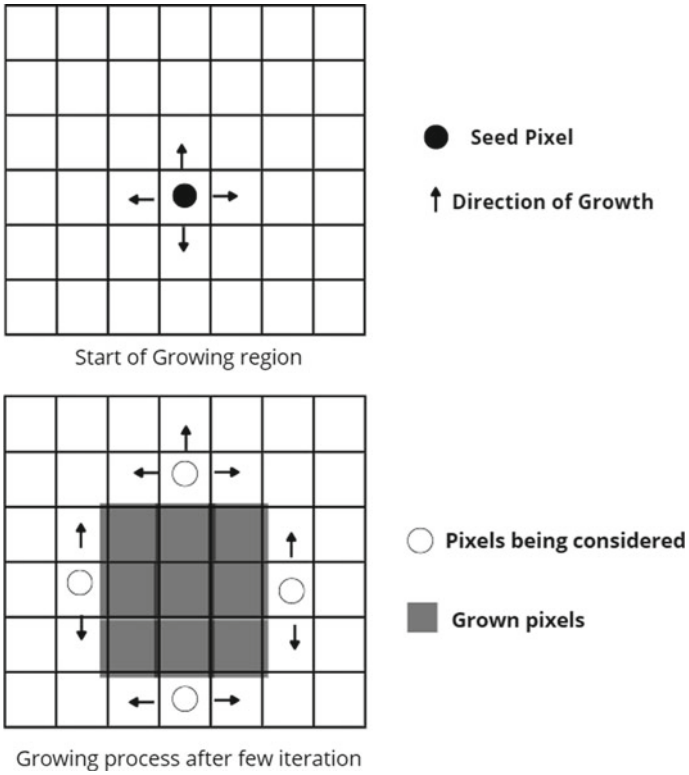


Fig. 1 Region growing [28]

In this algorithm first need to select the seed pixel of the selected region. After that pixel algorithm checks the neighboring pixel and decides whether in the region the pixel should be added or not by examining if those pixels meet certain define criteria or not like pixels with the same intensity or the color values of the pixel. And if those pixels meet that criterion, then by adding those neighbor pixel to the seed pixel the region starts to grow. It can be executed by the plain threshold approach.

3.4 Double Scattering

There are two approaches are used to find the double scattering in the urban flooded areas. The first approach used a single SAR which included flooded areas images. Most of the urban areas in the Terra SAR-X pictures are unflooded, and those areas are called aerial photos which may be flooded or maybe not. SAR images are much more easily available than aerial photos. Even only using synthetic aperture radar images unflooded areas would also detect.

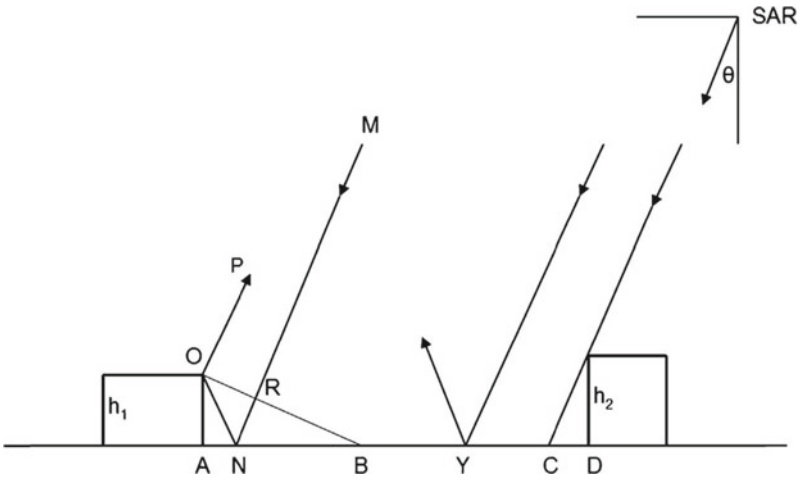


Fig. 2 Double scattering [3]

And the second approach for double scattering used the concept of multiple-scattering theory as detecting the change between the flooded and the unflooded pictures which is used for much complex and heterogeneous media (Fig. 2).

3.5 Bootstrap Method

It is a replacement sampling resampling strategy for estimating statistics. It can be used to compute the mean and standard deviation of the sample data. It is a metric used in applied machine learning to assess how well models perform when they are generating predictions on data that was not part of the training set.

The steps used are as follows:

- Step-1: Pick several bootstrap trials and obtain the sample data of the region of interest.
- Step-2: Choose a select size of test.
- Step-3: Choose several samples from the original sample of the interested region with equal size using the chosen measurement.
- Step-4: Evaluate the sample statistic.
- Step-5: Assess the sample statistics mean using the calculated data and use those distributed samples to calculate errors or execute the hypothesis test on the earliest sample.

3.6 Fuzzy Logic-Based Post Classification

For post-classification fuzzy logic-based algorithm was used with initial labeling. It is a useful method for combining clear data sources by considering their uncertainties to improve the accuracy of classification. Here, first image will convert into the fuzzy set, and according to that set classified result gives the membership value which is nothing but represents the degree of pixels to a different class. By merging this degree with specific information refining of data will be done. And fuzzy logic is used where data is undefined to increase the reliability of the expected result.

The fuzzy threshold helps to determine the degree of similarity which is also called membership value. And it converts images into binary regions.

The fuzzy thresholds are defined as [5],

$$x_{1[h]} = \mu_{h(\text{water})}$$
$$x_{2[h]} = \mu_{h(\text{water})} + f_\sigma * \sigma_{h(\text{water})}$$

where $x_{1[h]}$ and $x_{2[h]}$ are variables, $\mu_{h(\text{water})}$ = mean of the elevation of initially derived water objects and $\sigma_{h(\text{water})}$ = standard deviation of derived water objects.

The factor f_σ is defined by [5]:

$$f_\sigma = \sigma_{h(\text{water})} + 2.5$$

This feature was included to lessen the impact of the digital elevation model in areas of low topography.

3.7 Probabilistic Flood Mapping

The procedure of flood-plain mapping is essential for a number of processes in a probabilistic manner. These procedures include

1. Creating models for flood overflow.
2. Using data from earlier floods, carry out a sensitive inspection of the model.

At each *i*th cell, a weighted total of each Monte Carlo simulation is used to create the probabilistic map.

The flooded state C_j is, therefore [29],

$$C_j = \frac{\sum_k L_k W_{ki}}{\sum_k L_k}$$

where C_j = weighted average flood state for *j*th cell, W_{ki} = a moist cell, and L = a probability weight allocated to each *k*th simulation result and calculated using [29].

$$L_j = \frac{F_k - \min(F_k)}{\max(F_k) - \min(F_k)}$$

F fit metric that shows how well a replicate flood extent detects the flood extent and
$\max(F)$ and $\min(F)$ minimum and maximum measures of fit metric.

3.8 Normalized Difference Vegetation Index

The NDVI is nothing but an unlimited index that measures the thickness of green on a segment of land by describing the distinction in plant cover reflectance linking visible and near-infrared light.

It is calculated as [9],

$$\text{NDVI} = \left(\frac{\text{NIR} - \text{RED}}{\text{NIR} + \text{RED}}\right)$$

where

NIR reflection in the near-infrared spectrum and
RED reflection in the red range of the spectrum.

3.9 Modified Normalized Difference Water Index

The modified normalized difference water index utilizes green and SWIR bands to amplify open-water features. It is used to lower the characteristics of built-up areas connected to open water in other indexes. This is calculated using the below formula [9].

$$\text{MNDWI} = \left(\frac{\text{Green} - \text{SWIR1}}{\text{Green} + \text{SWIR1}}\right)$$

where

Green green bands pixel values.
SWIR Short-wave infrared pixel values and SWIR is defined as band11(1620 nm).

3.10 Normalized Difference Water Index (NDWI)

The normalized difference water makes use of open-water features in a satellite image, and it is nothing but the index that is used to identify the scope of water

bodies in remote areas and also used to extract the area of land which is covered by water. NDWI index successfully calculates the moisture content. The GREE-NIR mixture is used to enumerate the NDWI, allowing for the identification of minute changes in the water content of water bodies [19].

$$\text{NDWI} = \left(\frac{\text{Green} - \text{NIR}}{\text{Green} + \text{NIR}}\right)$$

where

NIR near-infrared band and
Green green band.

Due to the specific properties of water and the high wavelength of NIR, it penetrates through water easily. So, a combination of near-infrared bands absorbing light causes darker in the NIR band which is high as compared to other bands. Due to those properties, NDWI is used to detect water bodies from the images of remote sensor areas.

3.11 CNN (Convolutional Neural Network)

A CNN which is a deep learning network refers as a convolutional neural network. A CNN takes input as an image and according to weights assigned to each feature in the image differentiates one from the other. It is used for classification purposes for many visual applications like images and pictures. CNN has a multilayer perceptron Fig. 3, which means that all neurons present in one layer are connected to all neurons present in the next layer. A neural network contains minimum input of three layers, a hidden network, and an output. In the input layer, the neurons take the pixel values as input, then adjust the values of input using the activation function. Rectified linear unit (ReLU) is a mostly used activation function due to its simplicity, and it simply set all negative values to zero and keeps positive values unchanged. Using this activation value, network's expressive power enhances. After input it has a hidden network that can contain any number of layers, and then, it has the output layer which has the result.

First, the image is taken, and with the help of a filter or kernel, the feature vector of the image is gathered and then used to generate feature maps. Then those feature maps are fed into a fully connected layer for classification, and then, output is generated.

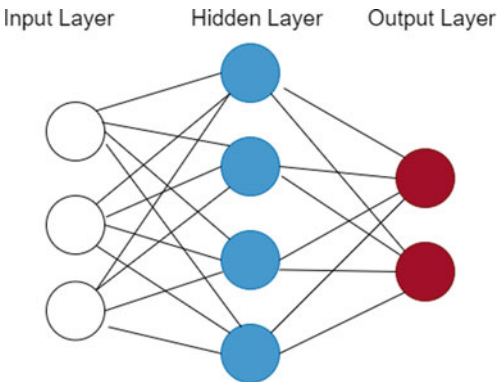


Fig. 3 Multilayer perceptron

4 Literature Survey

See Table 1.

5 Observation on Literature Survey

After the survey of 27 research papers, our paper summarizes techniques for image segmentation. In recent years, researchers have started using "machine learning algorithms" and "neural networks" for segmentation like "CNN", "FCNN", and "RCNN". Some researchers also mixed multiple algorithms to get more accurate results, e.g., "backscatter thresholding", "region growing", and "change detection". We also got some information about the free resource that can be used to get the free satellite images.

Machine learning algorithms and neural networks, such as convolutional neural network (CNN), fully convolutional neural network (FCNN), and region-based convolutional neural network (RCNN), have indeed gained popularity for image segmentation tasks due to their ability to learn complex patterns and features.

Combining multiple algorithms is a common approach to improve segmentation accuracy. Techniques like backscatter thresholding, region growing, and change detection can complement the capabilities of machine learning algorithms and neural networks. These methods often leverage specific characteristics of the satellite images or exploit contextual information to refine the segmentation results.

Here, we use image segmentation methods like Region Growing to create an initial estimate of the flooded areas, Convolutional Neural Network to improve the particular patterns and features connected to flooding and Dense-Net (Densely Connected Neural Network) use to design exact border localization, and the addition of contextual data, improving the segmentation outcomes by precisely defining the flooded areas within the satellite images.

Table 1 Literature survey table

References	Performance matrix	Summary
[1]	*F*-score > 0.90 *F*-score decreased to less than 0.75 in heterogeneous conditions *F*-score is a statistical measure used in binary classification Formula: *F*-score = 2 * (precision * recall)/ (precision + recall) The *F*-score ranges from 0 to 1, where a value of 1 indicates perfect precision and recall, while a value of 0 indicates poor performance	This paper studied the single-polarized high-resolution SAR satellite data. With a focus on SAR data, the paper presents an automatic flood detection system that combines histogram thresholding with segmentation-based categorization
[2]	Total good i.e., the accuracy by which the methods could differentiate between flooded and non-flooded regions on SAR image Method M1 = 81.6% Method M2a = 81.8% Method M2b = 81.7%	Synthetic aperture radar sensors are helpful to represent build-up environments where the flood risk is high. This paper has introduced M2b methods which is an advanced version of M2a. Method M1 focuses on SAR images and extracts the water pixels from the image. Method M2a includes the non-flooded image as a reference to detect the water and enhance the performance of the algorithm
[3]	Flooded double scattering detects 100% accuracy Unflooded with 91% accuracy	Here double scattering method is used in urban areas. Using the SAR image data and high-resolution LiDAR height map of the metropolitan region, this approach scattering intensities. The LiDAR data estimates the scattering curves in the images
[4]	*F*-value = 0.50 to 0.65 Where *F*-value is the ratio of two variances	In this paper, two strategies are used which are a non-parametric bootstrap method and another method is flood mapping procedure

(continued)

Table 1 (continued)

References	Performance matrix	Summary
[5]	Thailand Flood accuracy = 98.55% Non-flood accuracy = 71.99% Germany Flood accuracy = 95.37% Non-flood accuracy = 83.52%	Terra-SAR-X SAR using high resolution for real-time flood mapping or detection fully automatic processing chain was provided
[6]	Overall accuracy = 98%	This paper has proposed the algorithm known as flood mapping algorithm that can get SAR data from satellites that are based on the flood forecast. In SAR using dark objects classifier separates the non-flooded pixel and flooded pixel from images
[7]	By applying this method to four images of two rivers, we found error values were ranging from 0.04 to 0.23 It indicates the good performance of the proposed system	The probabilities flood mapping approach is employed in this paper, which is based on SAR data

(continued)

Table 1 (continued)

References	Performance matrix	Summary
[8]	The performance for estimating the dataset of the new hybrid bagging-LMT model was accuracy = 99.3%, kappa coefficient 0.949, which is highest among the other methods The representation for evidence dataset of the new hybrid bagging-LMT model was accuracy = 95.5%, kappa coefficient of 0.863, which is highest among the other methods The kappa coefficient (also known as Cohen's Kappa) is a statistical measure used to assess the agreement between two raters or observers Formula Kappa = (Po − Pe)/(1 − Pe) Where • Po is the observed proportion of agreement • Pe is the proportion of agreement expected by chance	This paper has proposed a "bagging-LMT" model for mapping flood vulnerability. It was validated using data from the Haraz watershed in Iran. LMT, logistic regression, and random forest are compared with the proposed model against soft computing models
[9]	Geomorphic-based flood maps are 95% similar to official flood maps available	This paper has proposed a semi-automatic method for flood mapping using freely available images and software. Multispectral data obtained by MODIS, Landsat, Senitel-2, SAR by Sentil-2, and Proba-V was used to detect flooded areas
[10]	Overall accuracy (traditional method DPSO) = 94.65% overall accuracy (proposed method SRFIM-MSI) = 96.81%	This paper has proposed a novel approach to the SRFIM by providing more spectral information to enhance flood mapping

(continued)

Table 1 (continued)

References	Performance matrix	Summary
[11]	In UAV dataset included the semantic segmentation of flooded areas with an accuracy of 96% and MioU of 92%	In this paper, two dense networks are employed which are CNN and RNN used for correctly segmenting out object boundaries and utilizing it and end learning. The network used for aerial image separation in UAV for flooded areas in Houston, TX
[12]	The overall accuracy of the algorithm when applied on England December 29, 2015 SAR data is 98.86%	Fully autonomous processing was employed in this paper. The procedure is tested on three separate places that have recently been flooded. It has an overall accuracy of above 98% with the value calculated by using *F*-measure that are between 0.64 and 0.92
[13]	A kappa value of 0.9238 was obtained by the flood crisis at Evros River, Greece A kappa value of 0.8682 was obtained by the flood events at Ouse River and Wharfe River in York, United Kingdom	In this presentation, quick flood mapping uses a two-step automated change detection chain based on Sentinel-1 SAR. A Jensen Shannon uses divergence-based index to choose a reference picture from a pool of probable image choices, and a log-ratio-based saliency detection is used
[14]	Overall Accuracy of supervised CNN $= 90.39 \pm 0.32\%$ A-SL CNN $= 92.82 \pm 0.10$ K statistic of supervised CNN $= 0.614 \pm 0.010\%$ A-SL CNN $= 0.686 \pm 0.004\%$	This study employed multitemporal "TerraSAR-X" data to detect urban floods using SAR intensity. Here, an active self-learning convolutional neural network framework was developed
[15]	Multi3Net has a 93.7% accuracy rate in segmenting flooded structures from non-flooded structures	They suggest a method for rapidly segmenting flooded buildings by combining multiresolution, multisensory, and multitemporal satellite images in a CNN in this paper. Using publicly available medium-resolution data, this program creates highly accurate segmentation maps of flooded buildings. Multi3Net was utilized to segment flooded structures by fusing multiresolution, multisensory, and multitemporal satellite images, and the results showed that full-fusion beat other fusion methods
[16]	–	This study explores thresholding and unsupervised classification approaches for locating areas in the Rapti and Ghaghara rivers in August 2017. Here, optical images and high-resolution multitemporal SAR were employed. It indicates that SAR data can be sufficient for flood water mapping and flood monitoring

(continued)

Table 1 (continued)

References	Performance matrix	Summary
[17]	The automated RAPID system exhibited producer and user accuracies of 93%, 77%, and 75%, respectively	We need real-time maps in case of a flood. Here, they developed an NRT system called Radar produced a diary for processing Sar data and producing reliable inundation maps
[18]	In VV polarization RGA-based LM thresholding gave the highest kappa coefficient of 0.91, with an overall accuracy of 98.82% In VH polarization TRS-based KI thresholding gave the highest kappa coefficient around 0.89, with an accuracy of 98.59%	This paper has introduced a multi-scale chessboard segmentation-based tiles selection method and a non-parametric thresholding algorithm for identifying water areas. Various methods like Otsu thresholding, KI thresholding, LM thresholding, TRS, and RGA were discussed
[19]	Overall accuracy = 80.95% kappa value = 74.1%	Change detection is an aspect, and thresholding selection is the most important part of flood mapping. Here they proposed a novel approach where there will be no need for thresholding selection. Instead of using change detection of thresholding, it uses land cover backscatter classification
[20]	Using high-resolution images they estimated an area-normalized accuracy of 89.8 ± 2.8% (95% c.i.), over Houston, Texas Thessaly, Greece, and Eastern Madagascar in January and March 2018, respectively, receive an overall rate of around 98.5% by comparing the results with the flood map derived by SAR	This paper has proposed an algorithm that takes all the Senitel-1 SAR data and other data required hosted on Google Earth Engine to map the flooding. This algorithm depends upon multitemporal SAR images to identify unexpected floods. The proposed algorithm eliminates the time-consuming data download and pre-processing

(continued)

Table 1 (continued)

References	Performance matrix	Summary
[21]	–	The Sen1 Flood11 dataset was utilized to estimate, verify, and test the neural network which is fully convolutional to segment flood water from the permanent water surface. The deep learning model for radar flood detection performs more accurately with flood water training labels rather than just permanent surface water
[22]	Overall classification accuracy of the HSWDC dataset was 93% and the kappa coefficient of 0.86	In this paper, they used the Google Earth Engine platform and Sentinel-1 imagery to map the surface of water bodies dynamically on a large scale with high spatial and temporal precision. Here uses threshold approach for SAR-based water mapping for the large-scale area with variant land cover types; samples across China are created
[23]	Their system gave an $F1$ score of around 96% and a kappa score of around 92% which good among the other systems	Using the Media Eval 2019 flood dataset, this research paper attempted to combine water index-like characteristics with a deep CNN-based solution for flood identification
[24]	–	In this paper, the approach is to process the SAR pictures of the same location from two different dates and establish flooded and non-flooded differentiation to provide a realistic estimate if the region is flood-prone or not

(continued)

Table 1 (continued)

References	Performance matrix	Summary
[25]	RG (Scenario-1 and no vegetation) = 70.5% RG (Scenario-1 and vegetation) = 81.7% RG (Scenario-1 overall accuracy) = 77.3% RG (Scenario-2 and no vegetation) = 90.7% RG (Scenario-2 and vegetation) = 92.1% RG (Scenario-2 overall accuracy) = 91.8% Integrated method (Scenario-1 and no vegetation) = 98.6% Integrated method (Scenario-1 and vegetation) = 81.7% Integrated method (Scenario-1 overall accuracy) = 88.4% Integrated method (Scenario-2 and no vegetation) = 98.6% Integrated method (Scenario-2 and vegetation) = 92.1% Integrated method (Scenario-2 overall accuracy) = 92.4%	This paper proposed an integrated approach for mapping the flood extent using the concept of FCN deep learning and RG. The digital elevation model was used to find the extent of flood beneath vegetation, and flood surfaces are extracted from high-resolution UAV imagery using the FCN model based on deep learning. Using an integrated strategy, floods can be detected in both visible and hidden places

(continued)

Table 1 (continued)

References	Performance matrix	Summary
[26]	The general reliability was higher than 90%, confirming the pertinence of the Otsu which is the automatic thresholding method in flood mapping	In this research paper, they used SAR images from Senital-1 and studied a flood-affected area in Iran. They have developed an algorithm called the Otsu thresholding algorithm, which will separate flood-affected areas from remaining land cover
[27]	Their overall accuracy was about 98.44% which is among the highest accuracy given by other methods, $F1$-score was about 95.39%, and IoU was 91.25% $F1$-score also known as F-score F-score is a statistical measure used in binary classification Formula $F\text{-score} = 2 * (\text{precision} * \text{recall})/(\text{precision} + \text{recall})$	Here using dense local feature compression network proposed a novel with the goal of automatically removing water bodies from various remote sensing photos

6 Proposed Architecture

The suggested model will direct the development of a system that will be utilized to minimize any tragedy from the flood by alerting authorities about the state of the water. The two core modules of suggested concepts for real-time flood monitoring systems utilizing satellite imagery are water detection and water body extraction. Our research indicates that CNN designs are the best option, but for real-time applications, CNN's DenseNet architecture is the greatest option. DenseNet is the best option for our system because of its user-friendliness and quicker, more precise results. The suggested model uses DenseNet, which is especially suited for image classification tasks, and it is the backbone for this proposed model. During training, the pre-processed images are fed as input to the DenseNet model, and using DenseNet, features are extracted very effectively throughout the network. DenseNet has dense blocks where every layer is connected to every other layer, helping with feature reuse and addressing the vanishing gradient problem.

7 Methodology

7.1 Training

Given train dataset with images and given a label for those images as below

$$\text{Dataset} = \{(x_1,\ y_1), (x_2,\ y_2), \ldots, (x_n,\ y_n)\}.$$

Pre-processing the image steps include:

- Step-1: Resize the images of the training dataset.
- Step-2: Normalization of images by dividing the pixel value.
- Step-3: To improve or increase the area cover by rotating and transforming the images, data augmentation techniques are mostly used.
- Step-4: Here subtract the mean value for each image.
- Step-5: Standardized the pixel of images.

Training Model

$X = \{x_1, x_2, \ldots, x_n\}$
//Set of images
$P = \{p_1, p_2, \ldots, p_n\}$
//Pre-processed images
for each x of p

$p =$ pre-processed images(x);

end for

Trained Model = CNN (P).

7.2 Testing

Taking the sample images after using this dataset as input for the trained model's final output will help to define the meaning of the given image.

8 Conclusion and Future Scope

In this work, we have studied various methods or strategies for software-based systems to detect floods from various types of satellite images. We have studied research papers from reputed journals. In Table 1, we have included the performance of the method used and a summary of the same. Figure 4 has a pie chart that shows the usage percentage of the image segmentation technique. The survey summarizes, we learned many new techniques of extracting the water bodies from the satellite images using image segmentation techniques and machine learning algorithms, and further use it for detection of the flood. The research papers which we referred to for this survey were published between 2009 and 2021. We discovered that there are not many applications like this for alert messages in urban areas, so we are creating one for flood monitoring. We will also try to improve the system's accuracy, which was previously described. In the existing system, processing the image takes a lot of time. However, we will also make improvements to the system's ability to analyze pictures more quickly by utilizing DenseNet architecture, a sort of CNN design (Fig. 5).

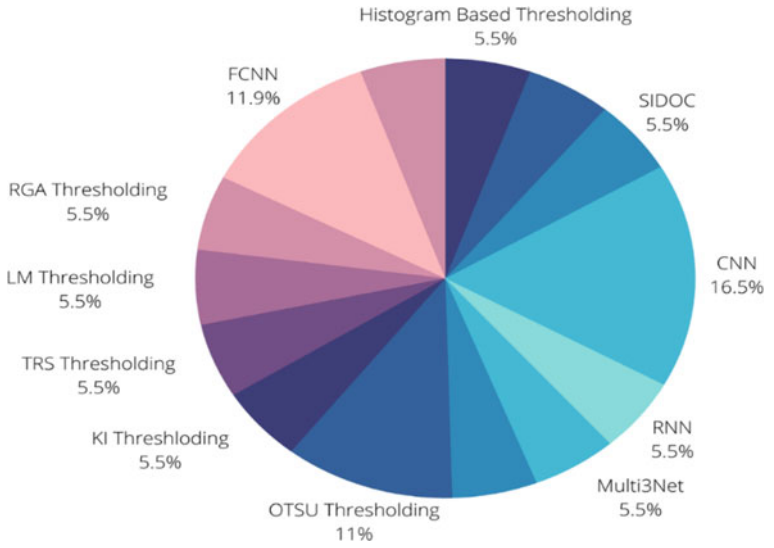


Fig. 4 Usage of various methods used to extract water from the satellite image

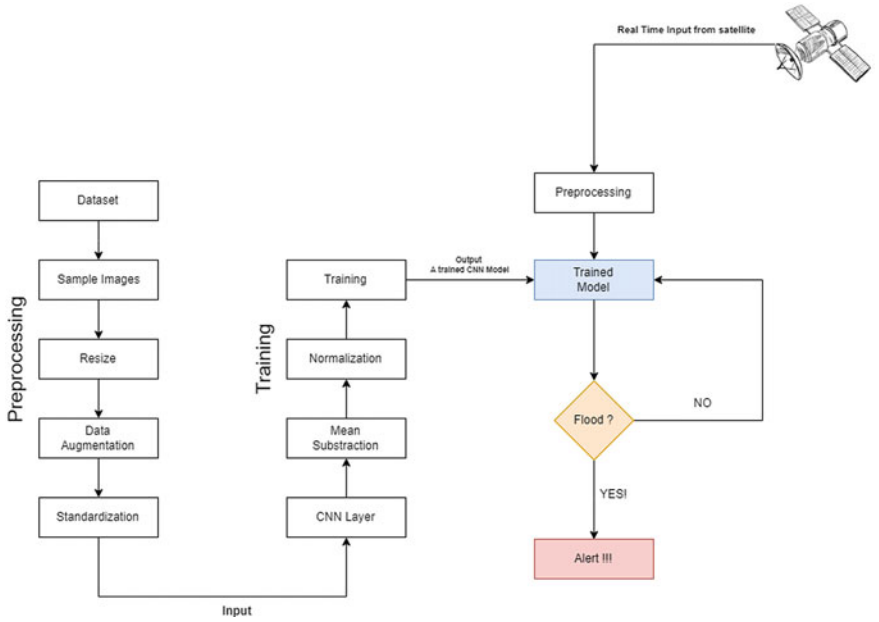


Fig. 5 Flowchart of system architecture

References

1. Martinis S, Twele A, Voigt S (2009) Towards operational near real-time flood detection using a split-based automatic thresholding procedure on high-resolution TerraSAR-X data. Nat Hazard 9:303–314. https://doi.org/10.5194/nhess-9-303-2009
2. Giustarini L, Hostache R, Matgen P, Schumann GJ, Bates PD, Mason DC (2013) A change detection approach to flood mapping in urban areas using TerraSAR-X. IEEE Trans Geosci Remote Sens 51(4):2417–2430. https://doi.org/10.1109/TGRS.2012.2210901
3. Mason DC, Giustarini L, Garcia-Pintado J, Cloke HL (2014) Detection of flooded urban areas in high-resolution Synthetic aperture radar images using double scattering. Int J Appl Earth Observ Geoinf 28:150–159. ISSN 1569-8432. https://doi.org/10.1016/j.jag.2013.12.002, https://www.sciencedirect.com/science/article/pii/S0303243413001700
4. Giustarini L, Vernieuwe H, Verwaeren J, Chini M, Hostache R, Matgen P, Verhoest N, De Baets B (2015) Accounting for image uncertainty in SAR-based flood mapping. Int J Appl Earth Observ Geoinf 34:70–77. https://doi.org/10.1016/j.jag.2014.06.017
5. Martinis S, Jens Kersten, André Twele, A fully automated TerraSAR-X based flood service. ISPRS J Photogrammetry Remote Sensing 104:203–212. ISSN 0924-2716. https://doi.org/10.1016/j.isprsjprs.2014.07.014, https://www.sciencedirect.com/science/article/pii/S0924271614001981
6. Boni G et al (2016) A prototype system for flood monitoring based on flood forecast combined With COSMO-SkyMed and Sentinel-1 data. IEEE J Sel Top Appl Earth Observ Remote Sens 9(6):2794–2805. https://doi.org/10.1109/JSTARS.2016.2514402
7. Giustarini L et al (2016) Probabilistic flood mapping using synthetic aperture radar data. IEEE Trans Geosci Remote Sens 54(12):6958–6969. https://doi.org/10.1109/TGRS.2016.2592951
8. Chapi K, Singh V, Shirzadi A, Shahabi H, Bui D, Pham B, Khosravi K (2017) A novel hybrid artificial intelligence approach for flood susceptibility assessment. Environ Model Softw 95:229–245. https://doi.org/10.1016/j.envsoft.2017.06.012
9. Notti D, Giordan D, Caló F, Pepe A, Zucca F, Galve JP (2018) Potential and limitations of open satellite data for flood mapping. Remote Sens 10:1673. https://doi.org/10.3390/rs10111673
10. Wang P, Zhang G, Leung H (2019) Improving super-resolution flood inundation mapping for multispectral remote sensing image by supplying more spectral information. IEEE Geosci Remote Sens Lett 16(5):771–775. https://doi.org/10.1109/LGRS.2018.2882516
11. Rahnemoonfar M, Murphy R, Miquel MV, Dobbs D, Adams A (2018) Flooded area detection from UAV images based on densely connected recurrent neural networks. In: IGARSS 2018—2018 IEEE international geoscience and remote sensing symposium, pp 1788–1791. https://doi.org/10.1109/IGARSS.2018.8517946
12. Bioresita F, Puissant A, Stumpf A, Malet J-P (2018) A method for automatic and rapid mapping of water surfaces from Sentinel-1 imagery. Remote Sensing 10:217. https://doi.org/10.3390/rs10020217
13. Li Y, Martinis S, Plank S, Ludwig R (2018) An automatic change detection approach for rapid flood mapping in Sentinel-1 SAR data. Int J Appl Earth Observ Geoinf 73:123–135. ISSN 0303-2434. https://doi.org/10.1016/j.jag.2018.05.023, https://www.sciencedirect.com/science/article/pii/S0303243418302782
14. Li Y, Martinis S, Wieland M (2019) Urban flood mapping with an active self-learning convolutional neural network based on TerraSAR-X intensity and interferometric coherence. ISPRS J Photogramm Remote Sens 152:178–191. https://doi.org/10.1016/j.isprsjprs.2019.04.014
15. Rudner T, Rußwurm M, Fil J, Pelich R, Bischke B, Kopackova-Strnadova V, Biliński P (2019) Multi3Net: segmenting flooded buildings via fusion of multiresolution, multisensor, and multitemporal satellite imagery. Proc AAAI Conf Artif Intell 33:702–709. https://doi.org/10.1609/aaai.v33i01.3301702
16. Nallapareddy A, Varadharajulu B (2019) Flood detection and flood mapping using multitemporal synthetic aperture radar and optical data. Egypt J Remote Sens Space Sci 23.https://doi.org/10.1016/j.ejrs.2019.01.001

17. Shen X, Anagnostou E, Allen G, Brakenridge R, Kettner A (2019) Near-real-time non-obstructed flood inundation mapping using synthetic aperture radar. Remote Sens Environ 221:302–315. https://doi.org/10.1016/j.rse.2018.11.008
18. Cao H, Zhang H, Wang C, Zhang B (2019) Operational flood detection using Sentinel-1 SAR data over large areas. Water 11:786. https://doi.org/10.3390/w11040786
19. Huang M, Jin S (2020) Rapid flood mapping and evaluation with a supervised classifier and change detection in Shouguang using Sentinel-1 SAR and Sentinel-2 optical data. Remote Sensing 12:2073. https://doi.org/10.3390/rs12132073
20. DeVries B, Huang C, Armston J, Huang W, Jones JW, Lang MW (2020) Rapid and robust monitoring of flood events using Sentinel-1 and Landsat data on the Google earth engine. Remote Sensing Environ 240:111664. ISSN 0034-4257. https://doi.org/10.1016/j.rse.2020.111664, https://www.sciencedirect.com/science/article/pii/S003442572030033X
21. Bonafilia D, Tellman B, Anderson T, Issenberg E (2020) Sen1Floods11: a georeferenced dataset to train and test deep learning flood algorithms for Sentinel-1. IEEE/CVF Conf Comput Vision Pattern Recogn Works (CVPRW) 2020:835–845. https://doi.org/10.1109/CVPRW50498.2020.00113
22. Li Y, Niu Z, Xu Z, Yan X (2020) Construction of high spatial-temporal water body dataset in China based on Sentinel-1 archives and GEE. Remote Sens 12:2413. https://doi.org/10.3390/rs12152413
23. Jain P, Schoen-Phelan B, Ross R (2020) Automatic flood detection in Sentinel-2 images using deep convolutional neural networks. https://doi.org/10.1145/3341105.3374023
24. Jardosh P, Kanvinde A, Dixit A, Dholay S (2020) Detection of flood prone areas by flood mapping of SAR imagery. Third Int Conf Smart Syst Inventive Technol (ICSSIT) 2020:814–819. https://doi.org/10.1109/ICSSIT48917.2020.9214089
25. Hashemi-Beni L, Gebrehiwot AA (2021) Flood extent mapping: an integrated method using deep learning and region growing using UAV optical data. IEEE J Sel Top Appl Earth Observ Remote Sens 14:2127–2135. https://doi.org/10.1109/JSTARS.2021.3051873
26. Moharrami M, Javanbakht M, Attarchi S (2021) Automatic flood detection using sentinel-1 images on the google earth engine. Environ Monit Assess 193:248. https://doi.org/10.1007/s10661-021-09037-7
27. Li M, Wu P, Wang B, Park H, Yang H, Wu Y (2021) A deep learning method of water body extraction from high-resolution remote sensing images with multisensors. IEEE J Sel Top Appl Earth Observ Remote Sens 14:3120–3132. https://doi.org/10.1109/JSTARS.2021.3060769
28. https://users.cs.cf.ac.uk/Dave.Marshall/Vision_lecture/node35.html
29. Di Baldassarre G, Schumann G, Bates PD, Freer JE, Beven KJ (2010) Flood-plain mapping: a critical discussion of deterministic and probabilistic approaches. Hydrol Sci J 55(3):364–376. https://doi.org/10.1080/02626661003683389
30. https://link.springer.com/article/10.1007/s13198-021-01152-5

Optimizing Permutations in Biclustering Algorithms



Aditya Shreeram, Tanmayee Samantaray, and Cota Navin Gupta

Abstract Data-driven methods used on structural magnetic resonance imaging (sMRI) have been successful in predicting disease subtypes within a population of subjects. Biclustering is a data-mining technique that can cluster rows and columns of a matrix to find submatrices corresponding to subtypes. One such method is N-BiC, which is an N-component biclustering algorithm. The limitation of this algorithm is that it is resource intensive, as it uses recursive Depth-First Search (DFS), which can be less efficient and time-consuming over iterative approaches. The biclusters enlisted in N-BiC are dependent on search order, and it requires a number of permutation operations to explore and merge the possible biclusters. In N-BiC, permutation refers to the order in which the components are sent to the modified DFS algorithm. The task of permutations in N-BiC is to find common subjects between components. For example, a permutation using three components (say 1, 2, and 3) relates to common subjects between components 1, 2, and 3. Similarly, yet another permutation 2, 3, and 1 relate to the same set of subjects between components 1, 2, and 3 in a different order, which is therefore redundant. Our proposed approach eliminates this repetitive process to obtain common subjects in a bicluster. Therefore, Modified N-BiC has optimized permutations and consumes reduced run time. It works on par with N-BiC and has reduced memory requirements. We also suggest that N-BiC algorithm does not find biclusters in the subject by voxel matrix as done by other biclustering studies but only in the decomposed loading matrix (subject by component). We also provide the necessary MATLAB codes for further research and application on other varied datasets.

Aditya Shreeram and Tanmayee Samantaray—Contributed equally.

A. Shreeram · T. Samantaray · C. N. Gupta (✉)
Neural Engineering Laboratory, Department of Biosciences and Bioengineering, Indian Institute of Technology Guwahati, Guwahati, Assam, India
e-mail: cngupta@iitg.ac.in

A. Shreeram
e-mail: sadityaaditya_s@iitg.ac.in

T. Samantaray
e-mail: tanma176106113@iitg.ac.in

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2023

S. Shakya et al. (eds.), *Fourth International Conference on Image Processing and Capsule Networks*, Lecture Notes in Networks and Systems 798,
https://doi.org/10.1007/978-981-99-7093-3_7

Keywords Structural magnetic resonance imaging · Depth first search · Biclustering · Data-mining algorithm · Source-based morphometry

1 Introduction

1.1 Literature Survey

Clustering is a popular unsupervised machine-learning technique that has been used to identify clusters within a dataset. Some of the applications of clustering include Image Segmentation and Recognition, Information retrieval, data mining, and so on [1, 2]. Even though it has many uses, clustering does not identify data points that may belong to several clusters over varying conditions (local patterns) [3]. For example, in a gene expression matrix, identifying the local patterns would lead to a greater understanding of genes that up-regulate or down-regulate together and the particular sample/condition in which such expression patterns occur. Biclustering is one such technique that can overcome this limitation of clustering. It is a data-mining technique that clusters rows and columns together. A bicluster refers to a two-dimensional sub-matrix within a larger matrix. They can be classified based on the type of values (constant, scaled variables, etc.), structure (row or column exclusivity and exhaustivity, etc.) or based on numerous evaluation measures like Mean Square Residue (MSR) or Scaled Mean Square Residue, etc., [4].

Biclustering also has applications in disease subtyping [4–9]. Other than biological data analysis, the technique has also been used in text mining [10, 11]. Some of the popular biclustering algorithms have been integrated into the toolbox BIDEAL, developed by [12]. However, it is a comparatively recent approach as applied to complex medical image data [13]. Some of the advancements concerning biclustering with neuroimaging datasets are Biclustered Independent Component Analysis (BICA) [14], N-component biclustering (N-BiC) [15], and Fuzzy non-negative matrix factorization (FNMF) [16]. In BICA, biclustering was performed on two components selected based on previous studies, whereas in N-BiC, multi-component biclustering was performed (in [14] and [15], sMRI data are used). In N-BiC, permutation refers to the order in which components are sent to the modified DFS algorithm. For example, a permutation using three components (say 1, 2, and 3) relates to common subjects between components 1, 2, and 3. Similarly, yet another permutation 2, 3, and 1 relate to the same set of subjects between components 1, 2, and 3 in a different order, which is therefore redundant. Similar FNMF was applied to diffusion tensor imaging (DTI) data, resulting in its decomposition to two different matrices with only positive entries. Overall, the number of studies in this domain is relatively less, and this warrants research for better and faster algorithms.

1.2 Aim of the Study

In this study, we attempt to improve upon N-BiC, which has been used for identifying the subtypes in a cohort of patients suffering from Schizophrenia [15]. Although the application of this algorithm onto neuroimaging datasets led to a better understanding of the disease, the method requires high resources due to the $n!$ permutation-based merging step. Our proposed algorithm, Modified N-BiC, which performs biclustering, includes usage of an iterative search method and performing comprehensive merging, thus removing the need for permutations. This optimization of the permutation count brings down the runtime and memory requirements significantly. To illustrate this, we consider two simulated datasets generated using custom MATLAB code and a publicly available multicenter sMRI dataset from Parkinson's Progression Marker's Initiative (PPMI) [17].

2 Materials and Methods

2.1 Datasets

Simulated Data and Bicluster Embedding. Two artificial datasets with embedded biclusters to mimic the subject by component matrix were created to compare the performance of N-BiC and Modified N-BiC. These were generated with a normal distribution in size (400 * 10) with the help of a MATLAB function *randn*. The matrix was normalized by using min_max_scaling to the range [0–1]. Once the data are generated, the biclusters were embedded in the data matrix (well separated to enable ease of interpretation) by artificially increasing the value of loading coefficients at particular locations. Dataset 1 has 10 columns, while Dataset 2 (Fig. 1) has 11 columns. Both datasets have three biclusters embedded with the following specifications (Table 1). The datasets have 10 and 11 columns, respectively, to demonstrate the effect of column count on performance.

Patient and Demographic Details. This is a publicly shared large multicenter [17] dataset from which 165 healthy controls (HC) and 216 Parkinson's disease (PD) patients were considered (Table 2). The studies were conducted following the Good Clinical Practice (GCP) principles and Declaration of Helsinki after approval was received from the regional ethics committees of the participating sites.

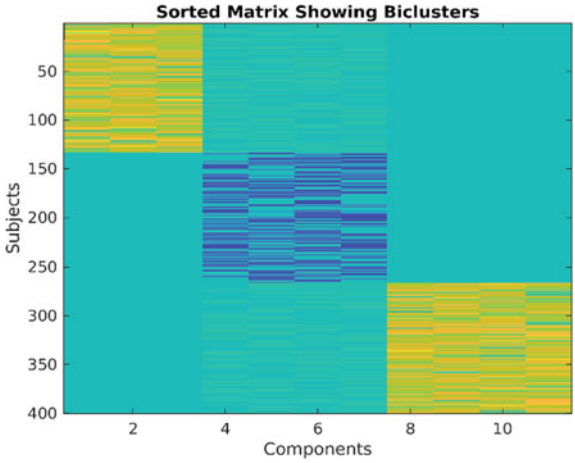


Fig. 1 Sorted eleven component datasets visualized using *imagesc* (MATLAB)

Table 1 Details of the simulated dataset

Dataset number	Bicluster number	Columns	Rows
1	1	1, 2, 3	1–133
	2	4, 5, 6	134–266
	3	7, 8, 9, 10	267–400
2	1	1, 2, 3	1–133
	2	4, 5, 6, 7	134–266
	3	8, 9, 10, 11	267–400

Table 2 Details of the patient demographics for PPMI participants

Measures	HC	PD
Count [total (male)]	165 (107)	216 (130)
Age in years (mean ± std.)	60.50 (11.03)	58.82 (7.39)
Age range	31–83	38–70

2.2 Device Specifications and Software

Following are the device specifications under which the experiments were run. CPU: I5-3470S, GPU: GT 710, System memory: 16 GB, Ram Specifications: DDR3 with 1600 MHz. All program codes were written using MATLAB 2022a software.

2.3 Structural Magnetic Resonance Imaging Data

The sMRI images were acquired using multiple scanner models of 1.5 or 3 T with scan acquisition parameters [18]. The sMRI scans were preprocessed according to the methodology followed in [19]. Computational anatomy toolbox, CAT12 [20], was used within Statistical Parametric Mapping (SPM 12) [21] for preprocessing in MATLAB R2021a [22]. The following steps were involved in the preprocessing:

1. Normalization and registration to standard 152 average T1 Montreal Neurological Institute (MNI) template.
2. Segmentation into different tissue types using unified segmentation algorithm [23].
3. Smoothing of gray matter (GM) images using full-width at half maximum (FWHM) Gaussian kernel of 8 mm.

The preprocessed images were regressed with age, gender, site, scanner, and field strength. These scans in the form of a 4D matrix (3D images over subjects) were flattened to a 2D matrix (subjects by voxel). Source-based morphometry (SBM) [24] was used on the subjects by voxel matrix, to obtain the loading coefficient matrix (subjects by components) and component matrix (components by voxel). The Group ICA of the fMRI toolbox (GIFT) was used to perform decomposition [25].

After SBM, a loading matrix of 381 * 30 was used. The number of components used in the current study was followed from previous subtyping studies [14, 15]. Of these 30 components, 10 and 11 number of components were considered separately to provide a balanced comparison between the two algorithms, N-BiC and Modified N-BiC. The N-BiC algorithm requires more than 42 GB to store 12! permutations of 12 components. In MATLAB, this takes up 12! * 12 * 8 bytes, where 8 is the number of bytes allocated by MATLAB to a single number of double data type. This gives around 42.8 GB to be exact.

2.4 Modified N-BiC Algorithm

Overview. A few steps of our algorithm are similar to the N-BiC algorithm [15] where the loading matrix is sorted according to the best methods explained therein (Step 1 of Fig. 2). The loading matrix, sorted using these methods, is further used to identify biclusters, by evaluating every combination of components at the given parameters. The loading coefficients of common subjects between pairs or groups of components are stored in a separate data structure (bicluster search) (Step 2 of Fig. 2). At the end of the search, the biclusters are sent to a local validator, where every bicluster is compared with $(N - 1)$ biclusters in the set. N is the number of biclusters enlisted and biclusters already compared are excluded. The biclusters are merged based on overlap criteria, and thus, final list contains unique biclusters for further analysis. The total number of such comparisons comes out to be $N * (N -$

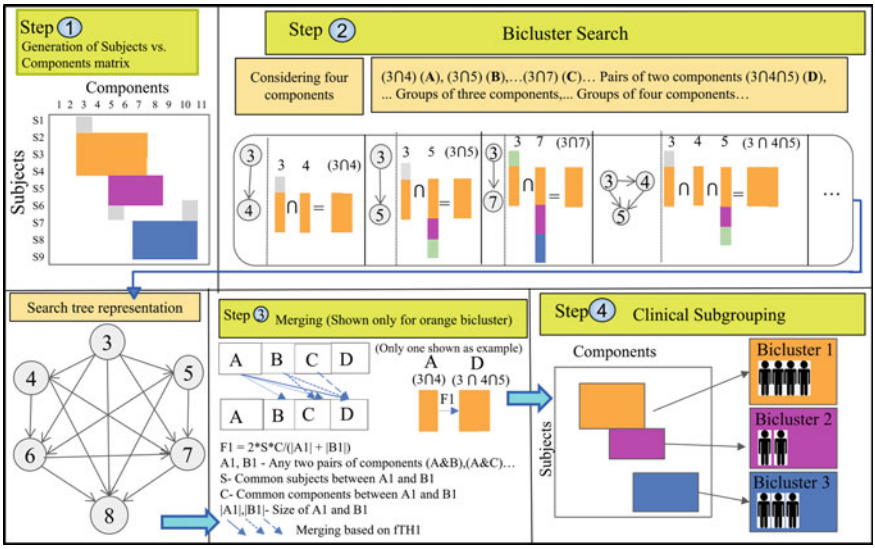


Fig. 2 Graphical abstract depicting methodology behind Modified N-BiC depicting bicluster search, merging, and subgrouping

1)/2. This validation process is shown in Step 3 of Fig. 2 (merging). After Step 3, required subgroups are identified from the bicluster list (Step 4 of Fig. 2).

Sorting. The main idea of sorting is to select only those subjects which show high expression for a particular component. For each component, the individual loading coefficients are compared with the mean loading coefficient and then retained or eliminated based on the threshold. More details on sorting methods could be found in Sect. 3.1 and [15].

Parameters and Search Bic function. The algorithm is a type of Depth-First Search (DFS), called iterative deepening DFS [26], which explores the intersection of all possible subsets of components. It combines the approaches of DFS and Breadth-First Search (BFS) [26] as it looks for the intersection between components at a particular depth before moving to a higher depth. For example, intersection for common subjects is performed between pairs of two components such as (1, 2), (1, 3), (1, 4), and so on, after which intersection between groups of three components (1, 2, 3), (1, 2, 4) happens and so on. This process continues till maximum depth is reached (total number of components) as shown in Step 2 of Fig. 2. The total number of such component pairs is given in Eq. (1).

$$\text{TotalPair(TP)} = \sum\nolimits_{k=2} \binom{n}{k}, \tag{1}$$

where k goes from 2 to n (n is the number of components in the matrix).

We used MINTERSECT [27], a MATLAB helper file, for performing sequential intersection. The parameters (NoS, K, O) referred to in [15] as (S, K, O) are used for our approach except for *fTH2*, which is the overlap criteria for merging over all the permutations. This is because in Modified N-BiC, the merging takes place with a single validator, requiring a single overlap parameter (*fTH1*). The overlap between two biclusters is determined by the $F1$ index or Dice index, which calculates the similarity between two biclusters [15]. The total number of operations (TO) in the worst-case scenario for Modified N-BiC is given by Eq. (2), while that for N-BiC is given by Eq. (3).

Let *Pair* represent an array of length TP (Eq. 1) containing all the pairs of components, m be the number of non-zero elements in a particular column, N be the total number of enlisted biclusters, M, P be the number of subjects, components, respectively, in bicluster 1, and Z, Q be the number of subjects, components, respectively, in bicluster 2 for pairwise comparison.

The total number of operations in our proposed approach, Modified N-BiC, is:

$$\mathrm{TO}_{\text{Modified - N - BiC}} = \sum_{i=1}^{\mathrm{TP}} \prod_{k=1}^{\text{length(Pair}(i))} m_k + \sum_{j=1}^{\frac{N*N-1}{2}} \left(M_j * Z_j + \left(P_j * Q_j\right)\right) \tag{2}$$

whereas, in N-BiC, the total number of operations is:

$$\mathrm{TO}_{\text{N - BiC}} = \sum_{a=1}^{\mathrm{NP}} \left[\sum_{i=1}^{\mathrm{TP}} \prod_{k=1}^{\text{length(Pair}(i))} m_k + \sum_{j=1}^{\mathrm{VN}} \left(M_j * Z_j + \left(P_j * Q_j\right)\right) \right] \tag{3}$$

where VN is $\frac{N*N-1}{2}$ when one bicluster is compared to $N-1$ remaining biclusters.

Here, NP is the number of permutations considered, VN is the variable number of biclusters, and since the bicluster merging is order dependent, merged bicluster pairs are removed. Here the first term in Eqs. (2) and (3) denotes the number of comparison operations involved in a component pair, which is sum of the product of m over all the pairs (TP) in a permutation. The second term in Eq. (2) relates to the intersection between two biclusters, which is the computed intersection of the common subjects/indices (M and Z) and common components (P and Q) summed over $N * (N-1)/2$. In Eq. (3), this relates to the number of operations in internal validation (internal summation across a variable number of biclusters) and the summation over all the permutation steps. To perform an exhaustive merging across all permutations, NP $= n!$.

2.5 Evaluation Measures

Mean Square Residue (MSR). The Mean square residue [4] is a measure reflecting the homogeneity of a bicluster. With MSR, it is possible to identify biclusters with similar values of loading coefficients, allowing one to better understand the dataset.

$$\mathrm{MSR}(B) = \frac{1}{|I| * |J|} \sum_{i=1}^{i=|\mathrm{I}|} \sum_{j=1}^{j=|\mathrm{J}|} (b_{ij} - b_{iJ} - b_{Ij} + b_{IJ})^2, \tag{4}$$

Here, B is a bicluster, and $|I|$, $|J|$, b_{ij}, b_{iJ}, b_{Ij}, b_{IJ} stand for bicluster row size, bicluster column size, element in the ith row and jth column, row means, column means, and total means, respectively.

F1 Index and Consensus Score (CS). The $F1$ index is calculated for each pair of extracted biclusters and original biclusters. Consensus score [28] denotes the consensus between the final list of biclusters (identified/extracted) and the original list of biclusters (ground truth).

Let us suppose there are three ground truth biclusters A, B, and C, and three extracted biclusters $A1$, $B1$, and $C1$. Suppose A has more subjects in common with $A1$, B in common with $B1$, and C in common with $C1$, according to $F1$ indexes. In this case, the $F1$ index is given by Eq. (5), and the consensus score is given by Eq. (6).

$$\text{Number of pairs} = 3\{(A, A1), (B, B1), (C, C1)\}, \tag{5}$$

$$\sum F1 \text{ indexes} = F1(A, A1) + F1(B, B1) + F1(C, C1)$$
$$\mathrm{CS} = \frac{\sum F1 \text{ indexes}}{\text{Number of pairs}} \tag{6}$$

2.5.1 Comparison with Bimax

We compared the performance of Modified N-BiC with an existing biclustering algorithm called Bimax [29], using the toolbox BIDEAL [12]. Bimax is a divide and conquer algorithm, which divides the whole binarized (1's and 0's) matrix into several partitions and searches for biclusters with only 1's. The algorithm was chosen for its ease of use and interpretation.

3 Results and Discussion

We compare the three algorithms namely, N-BiC, Modified N-BiC, and Bimax, using simulated data and the PPMI dataset. Positive and negative sorting methods as included within the N-BiC toolbox, were the default sorting method for all scenarios.

3.1 Simulated Dataset

The runtime and average MSR on simulated datasets are compared and shown in Table 3. Table 4 lists the biclusters identified by both algorithms on Dataset 1. The comparison of consensus scores of each algorithm over the simulated dataset is shown in Table 5. Table 6 represents biclusters enlisted by both algorithms on dataset 2. Due to space constraints, only the results at 100 permutations are shown in Tables 4 and 6. Parameter settings used for the following experiments: NoS = 125, $K = 2$, $O = 0.8$, where NoS—minimum number of subjects in a bicluster, K—minimum number of components in a bicluster, O—overlap between the biclusters. These parameters were set based on the rationale of the N-BiC algorithm.

Biclusters discovered by Bimax on Dataset 1. Parameters settings used for this experiments: min_rows (minimum rows): 125, min_cols (minimum columns): 2, NumBiClust (no. of biclusters): 20. Details of the biclusters found {Components, Number of Subjects} in Dataset 1: {(2, 3), 128}, {(1, 3), 127}, {(1, 2), 129}, {(1, 2, 3), 126}, {(7, 8), 129}, {(8, 9), 128}, {(7, 9), 131}, {(7, 8, 9), 127},{(7, 10), 131}, {(7, 8, 10), 127}, {(7, 9, 10), 129}, {(7, 8, 9, 10), 125}.

Biclusters discovered by Bimax on Dataset 2. Parameters settings used for this experiments: min_rows (minimum rows): 125, min_cols (minimum columns): 2, NumBiClust (no. of biclusters): 20. Details of the biclusters found by

Table 3 Performance of both algorithms on simulated datasets

Number of components	Number of permutations	N-BiC (RT)	Modified N-BiC (RT)	N-BiC (AvMSR)	Modified N-BiC (AvMSR)
10	1	0.66	0.82	1.04	0.94
10	2	0.74	–	1.07	–
10	5	0.82	–	1.06	–
10	100	3.00	–	1.06	–
11	1	6.61	0.98	0.86	1.01
11	2	6.50	–	0.89	–
11	5	6.68	–	0.90	–
11	100	9.03	–	0.91	–

RT runtime (seconds), *AvMSR* average MSR

Table 4 Biclusters enlisted by both algorithms on dataset 1

S. No.	N-BiC (number of components)	Modified N-BiC (number of components)	N-BiC (number of subjects)	Modified N-BiC (number of subjects)
1	[1, 2]	[1, 2]	125	129
2	[1, 3]	[1, 3]	126	127
3	[2, 3]	[2, 3]	126	128
4	[7–9]	[7–9]	125	127
5	[7, 8, 10]	[7, 8]	125	129
6	[7, 9, 10]	[7, 10]	126	131
7	[8–10]	[8, 9]	125	128
8	–	[8, 10]	–	127
9	–	[9, 10]	–	129
10	–	[1–3]	–	126
11	–	[7, 9]	–	131
12	–	[8, 9]	–	128

Table 5 Consensus score comparison on simulated datasets

Number of components	Number of permutations	N-BiC	Modified N-BiC
10	1	0.806	0.7396
10	2	0.793	–
10	5	0.798	–
10	100	0.802	–
11	1	0.706	0.743
11	2	0.691	–
11	5	0.692	–
11	100	0.682	–

Bimax{Components, Number of Subjects}: {(2, 3), 128}, {(1, 3), 127}, {(1, 2), 129}, {(1, 2, 3), 126}, {(8, 11), 127}, {(10, 11), 128}, {(8, 10), 127}, {(8, 9), 128}.

3.2 PPMI Dataset

Parameter settings used for performance comparison using the PPMI dataset are NoS $= 40$, $K = 2$, and $O = 0.8$. The PD-only loadings were considered for biclustering on components: (1:10) and (1:11), where NoS—minimum number of subjects in a bicluster, K—minimum number of components in a bicluster, O—overlap between the biclusters. A comparison of runtime and average MSR on the loading matrix is

Table 6 Biclusters enlisted by both algorithms on dataset 2

S. No.	N-BiC (number of components)	Modified N-BiC (number of components)	N-BiC (number of subjects)	Modified N-BiC (number of subjects)
1	[1, 2]	[1, 2]	126	129
2	[1, 3]	[1, 3]	126	127
3	[2, 3]	[2, 3]	126	128
4	[8, 9]	[8, 9]	125	128
5	[8, 10]	[8, 10]	125	127
6	[8, 11]	[8, 11]	125	127
7	[9, 10]	[9, 10]	125	129
8	[9, 11]	[9, 11]	126	129
9	[10, 11]	[10, 11]	126	128
10	–	[1–3]	–	126
11	–	[9–11]	–	126
12	–	[8–10]	–	125
13	–	[8, 9, 11]	–	125

shown in Table 7. These parameters were set based on the rationale of the N-BiC algorithm, although other configurations can be explored in future studies.

Table 7 Performance on PPMI dataset

Number of components	Number of permutations	N-BiC (RT)	Modified N-BiC (RT)	N-BiC (AvMSR) (* 10^8)	Modified N-BiC (AvMSR) (* 10^8)
10	1	0.61	0.15	3.58	3.15
10	2	0.64	–	3.71	–
10	5	0.67	–	3.33	–
10	100	2.11	–	3.15	–
11	1	6.51	0.19	4.09	3.38
11	2	6.44	–	3.62	–
11	5	6.58	–	3.63	–
11	100	8.46	–	3.38	–

RT runtime (s), *AvMSR* average MSR

3.3 Performance of Modified N-BiC on PPMI Dataset

Parameter settings used for bicluster search using the PPMI dataset: NoS = 55, K = 2, O = 0.8, where NoS—minimum number of subjects in a bicluster, K—minimum number of components in a bicluster, O—overlap between the biclusters. The PD-only loadings were considered for biclustering on all 30 components. Biclusters discovered: Components [3 and 16] with 56 subjects and components [5 and 16], [5 and 17], [25 and 30] with 55 subjects each. Modified N-BiC is capable of finding biclusters from a loading matrix with 30 components, under the same system configuration. Although we have executed the algorithm under various parameters of *NoS*, K, and *O*, the results have only been presented for one parameter setting as proof of execution. Biclusters common to four or more components were not generated under various parameters, so the algorithm was executed up to the intersection between three components. Modified N-BiC used 5 MB for Dataset 1 and 7.2 MB for Dataset 2 (both approximates). The memory usage was calculated using *whos* command in MATLAB. N-BiC on the other hand used 276.8 MB for Dataset 1 and 3.27 GB for Dataset 2 (just for allocation of the space for permutation possibilities), and it requires 42 GB of system memory for the same (12 components and more). At the same system configuration, N-BiC failed to run, but Modified N-BiC successfully discovered the biclusters efficiently. This indicates the feasibility of execution for Modified N-BiC and a higher number of valid biclusters returned by exhaustive search approaches in a reasonable time. It can be seen that Bimax performed on par with N-BiC as well as Modified N-BiC on Dataset 1 and Dataset 2. Thus, we suggest that Modified N-BiC can work comparably to existing algorithms, although more research on this aspect is required.

It is observed from Tables 3 and 7 that the run time increases for both N-BiC and Modified N-BiC with an increase in the number of components, though the increase in time for N- BiC is more than that for Modified N-BiC. This is because N-BiC is a backtracking algorithm that uses recursive DFS to enumerate search possibilities, and recursion can be less efficient in terms of time and space complexity over iterative approaches, especially with an increase in the number of operations [30].

For the N-BiC, the algorithm has to be run for a higher number of permutations (dependent on the dataset) to get stable results. This is because the arrangement/permutation plays a role in the enlistment of biclusters. Increasing the number of permutations increases the running time and has a varying influence on the average MSR due to a varying number of biclusters enlisted; every time the algorithm is executed. In the end, it can be seen from Tables 3, 4, 5, 6 and 7 that the Modified N-BiC has an on-par or similar performance with N-BiC.

4 Research Limitations/Implications

The biclusters enlisted by the algorithm are dependent on NoS, K, and O and hence these need to be tuned very properly. These values also affect the runtime of the algorithm (same as in N-BiC [15]). For example, if the overlap parameter (O) is set too high, then too many similar biclusters are listed, making the analysis tedious. In contrast, if the O value is set to be too low, an aggressive merging takes place, and too few biclusters are left for analysis.

5 Originality and Value

This study presents an improvement over N-BiC [15] in terms of reduction in memory requirements and run time while providing on-par performance with the same (bicluster enlistment). The improvement comes in the form of a switch to an iterative deepening DFS over the recursive DFS with backtracking in N-BiC. We have also optimized the merging step by comparison of a bicluster with other biclusters in a single permutation, which reduces the necessity for $n!$ permutations in the worst case down to one permutation. Hence, our approach seems to provide similar results at a lesser computational cost. The value of Modified N-BiC lies in its potential as an alternative biclustering algorithm that can be used for disease subtyping not only from sMRI data but for other varied applications like cancer and gene analysis.

6 Conclusion and Future Research Work

In this study, we suggest improvements for the N-BiC algorithm by using an iterative DFS approach and a comprehensive merging step, removing the need for permutations. The Modified N-BiC approach works on par with the N-BiC approach and uses fewer resources than the latter. We hope that the provided source code and datasets for download will enable further testing and validation of our work. This study also shows the feasibility of execution of exhaustive biclustering algorithms on devices with low specifications, especially for subtyping. This algorithm can still be improved upon, in the future by use of optimizations like dynamic programming approach. It can also be used on other datasets to obtain biologically relevant biclusters.

Code Availability The simulated data and codes used in the current study are available at the below GitHub link: https://github.com/NeuralLabIITGuwahati/Permutations_Biclustering.

Author Contributions CNG proposed the idea to AS and TS. AS did the entire work and had frequent discussions with TS and CNG. The first draft of the paper was written by AS and TS. Numerous discussions then between AS and CNG led to the final submitted manuscript.

Acknowledgements The authors are thankful to PPMI for providing the dataset. Aditya Shreeram was funded by Ministry of Education (MoE) scholarship by the Government of India (GOI). Tanmayee Samantary was funded by MoE doctoral scholarship by GOI. Cota Navin Gupta's time was supported by the Scheme for Promotion of Academic and Research Collaboration (SPARC Grant), Government of India, Project Code: P1073.

References

1. Saxena A, Prasad M, Gupta A, Bharill N, Patel OP, Tiwari A, Er MJ, Ding W, Lin CT (2017) A review of clustering techniques and developments. Neurocomputing 267:664–681
2. Adepu B, Gyani J, Narsimha G (2021) A novel multi-objective differential evolution algorithm for clustering data streams. In: Lecture notes in networks and systems, pp 67–78
3. Orzechowski P, Boryczko K, Moore JH (2019) Scalable biclustering—the future of big data exploration? GigaScience 8(7):giz078
4. Pontes B, Giráldez R, Aguilar-Ruiz JS (2015) Biclustering on expression data: a review. J Biomed Inform 57:163–180
5. Noronha MDM, Henriques R, Madeira SC, Zárate LE (2022) Impact of metrics on biclustering solution and quality: a review. Pattern Recogn 127:108612
6. Ramkumar M, Basker N, Pradeep D et al (2022) Healthcare biclustering-based prediction on Gene expression dataset. Biomed Res Int 2022:1–7
7. Wang YK, Print CG, Crampin EJ (2013) Biclustering reveals breast cancer tumour subgroups with common clinical features and improves prediction of disease recurrence. BMC Genomics 14(1):102
8. Samantaray T, Saini J, Gupta CN (2022) Subgrouping and structural brain connectivity of Parkinson's disease—past studies and future directions. Neurosci Inform 2:100100
9. Sun J, Bi J, Kranzler HR (2013) Multi-view biclustering for genotype-phenotype association studies of complex diseases. In: 2013 IEEE international conference on bioinformatics and biomedicine, 316–321
10. de Castro PA, de França FO, Ferreira HM, Von Zuben FJ (2007) Applying biclustering to text mining: an immune-inspired approach. Lecture notes in computer science, 83–94
11. Dhillon IS, Mallela S, Modha DS (2003) Information-theoretic co-clustering. In: Proceedings of the ninth ACM SIGKDD international conference on knowledge discovery and data mining
12. Verma NK, Sharma T, Dixit S, Agrawal P, Sengupta S, Singh V (2021) Bideal: a toolbox for bicluster analysis—generation, visualization and validation. SN Comput Sci 2(1):24
13. Castanho EN, Aidos H, Madeira SC (2022) Biclustering fMRI time series: a comparative study. BMC Bioinform 23(1):192
14. Gupta CN, Castro E, Rachkonda S et al (2017) Biclustered independent component analysis for complex biomarker and subtype identification from structural magnetic resonance images in Schizophrenia. Front Psychiatry 8
15. Rahaman MA, Mathalon D, Lee HJ et al (2020) N-BIC: a method for multi-component and symptom biclustering of structural MRI data: application to schizophrenia. IEEE Trans Biomed Eng 67(1):110–121
16. Arnedo J, Mamah D, Baranger DA et al (2015) Decomposition of brain diffusion imaging data uncovers latent schizophrenias with distinct patterns of white matter anisotropy. Neuroimage 120:43–54
17. Marek K et al (2018) The Parkinson's progression markers initiative—establishing a PD biomarker cohort. Ann Clin Transl Neurol 5(12):1460–1477
18. Parkinson's Progress Marker's Initiative Scanner Information. https://www.ppmiinfo.org/sites/default/files/docs/archives/PPMI2.0_MRI_TOM_Final_FullyExecuted_v2.0_20200807.pdf

19. Samantaray T, Saini J, Gupta CN (2022) Sparsity dependent metrics depict alteration of brain network connectivity in Parkinson's disease. In: 44th annual international conference of the IEEE engineering in medicine & biology society (EMBC), pp 698–701
20. Computational anatomy toolbox (CAT12). http://www.neuro.uni-jena.de/cat/
21. SPM 12. https://www.fil.ion.ucl.ac.uk/spm/software/spm12
22. MATLAB, Natick, Massachusetts. https://www.mathworks.com/products/matlab.html
23. Ashburner J, Friston KJ (2000) Voxel-based morphometry—the methods. Neuroimage 11(6):805–821
24. Xu L, Groth KM, Pearlson G, Schretlen DJ, Calhoun VD (2009) Source-based morphometry: the use of independent component analysis to identify gray matter differences with application to schizophrenia. Hum Brain Mapp 30:711–724
25. Group ICA of fMRI toolbox. https://github.com/trendscenter/gift
26. Korf RE (1985) Depth-first iterative-deepening: an optimal admissible tree search. Artif Intell 27:97–109
27. MINTERSECT—Multiple set intersection. https://in.mathworks.com/matlabcentral/fileexchange/6144-mintersect-multiple-set-intersection
28. Hochreiter S, Bodenhofer U, Heusel M et al (2010) Fabia: factor analysis for bicluster acquisition. Bioinformatics 26:1520–1527
29. Prelić A, Bleuler S, Zimmermann P, Wille A, Bühlmann P, Gruissem W, Hennig L, Thiele L, Zitzler E (2006) A systematic comparison and evaluation of biclustering methods for gene expression data. Bioinformatics 22:1122–1129
30. Sánchez MR (2018) Basic concepts of recursive programming. Introduction to recursive programming. CRC Press, Taylor & Francis Group, Boca Raton, p 26

Extracting Graphs from Plant Leaf Venations Using Image Processing



Ashlyn Kim D. Balangcod and Jaderick P. Pabico

Abstract Leaf venation is an important characteristic of plant species that can lead to the identification of a plant. The study of leaf venation is laborious if done by manual inspection. This study proposes a leaf venations' extraction process requiring minimal algorithms of image processing. Canny edge detection and Zhang-Suen thinning algorithm were used to extract the venation of the leaf images. Using ground truth tracing, the average values computed are 0.522 for correctness and 0.406 for completeness of the extracted venation. This study also presents the use of graph metrics to describe the leaf venations of plants. Using thirty images of five plant species, leaf venations up to the tertiary veins are compared using graph properties: size, volume, number of claws, number of wedges, and maximum degree. This study proposes these mentioned graph metrics as additional characteristics of leaf venations as results show a possible pattern for the five plant species.

Keywords Leaf venations · Image processing · Graph properties

1 Introduction

The morphological features of leaves are used as a basis for plant identification. Some flowering plants may even be identified by leaves alone. Some characteristics used for identification include the shape of the leaf, texture, margins of leaf blade, and leaf venations [1]. Leaf venations, or the positioning of the veins on the blade, are known to provide the mechanical support of the leaf. It contains the phloem and

A. K. D. Balangcod (✉)
Department of Mathematics and Computer Science, University of the Philippines Baguio, Baguio, Philippines
e-mail: adbalangcod@up.edu.ph

J. P. Pabico
Institute of Computer Science, University of the Philippines Los Baños, College, Laguna, Philippines
e-mail: jppabico@up.edu.ph

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2023

S. Shakya et al. (eds.), *Fourth International Conference on Image Processing and Capsule Networks*, Lecture Notes in Networks and Systems 798,
https://doi.org/10.1007/978-981-99-7093-3_8

xylem which transports the nutrients and water, respectively, needed for the plant's processes [2].

There had been studies on the other roles of plant venations aside from plant identification, such as tracing the historical lineage of genus Gunnera using leaf fossil records [3], understanding climatic conditions [2], and predicting season temperature and atmospheric carbon dioxide of an area [4].

Plants can be separated into two categories, and they are the monocots and the dicots. These two categories have different leaf venations. Generally, for monocots, leaves are parallel venations occurring side-by-side without intersecting. Dicots mostly have netted or reticulate leaf venations [1]. An extensive document to classify dicot leaves has been established by Hickey [5] and is still currently in use. An important aspect of leaf veins is the order of venation where the veins are classified according to their width relative to their point of origin. The primary vein of the first degree (1°) is the thickest vein. Second-degree (2°) veins are the next smaller veins and protrude from the primary veins. Higher-order venations such as tertiaries (3°), quarternaries (4°), and quinternaries (5°) are the next finer veins arising from their previous lower-order veins [5]. Aside from the order of venations, some quantitative measures for plant leaf venations are vein density, edges per area, number of areoles, number of nodes, and number of edges [6].

Prior to the computation of these metrics, leaves have to be extracted. There are different methods for extracting the venations, either manually or through image processing. Manual extraction requires chemical clearing of the leaf and staining which, in some cases, requires hours sometimes days to fully clear the leaves to obtain the veins. Another method is the visual inspection of high-resolution leaf photographs, which is another tedious and time-consuming process.

The approach proposed in this study is to use automatic extraction of the leaf images using algorithms of image processing, specifically the use of Canny edge detection and Zhang-Suen thinning algorithm, and a novel graph detection from the extracted venation image. The plants considered for this study are selected dicot plants found in the area.

2 Related Works

Some applications and algorithms have been developed using image processing to extract the leaf venations either automatically or with human intervention. Some of these are the Leaf Extraction Analysis Framework Graphical User Interface (LEAF GUI) by Price et al. [6], PhenoVein by Buhler et al. [7], Leaf Image Analysis Interface (LIMANI) by Dhondt et al. [8], and the Network Extraction from Images (NEFI) by Dirnberger et al. [9]. These approaches used different methods in the extraction of the venations using processes such as image segmentation, skeletonization, and computation of quantitative measures. Figure 1 summarizes the steps defined in LIMANI application. Figure 2 is the summary of the algorithm involved in the Leaf

GUI software, while Fig. 3 is the processes involved in the methodology of the phenoVEIN software.

The methodologies of LIMANI, LEAF GUI, and phenoVEIN produce quantitative data of leaf venations. LIMANI's results are data on lamina area, vascular vein length, length of free-ending veins, areas of areoles, and other data [8]. LEAF GUI has an output of the quantitative data of the vein network such as leaf area, leaf perimeter, edges per area, number of areoles, and areole statistics [6]. The measured

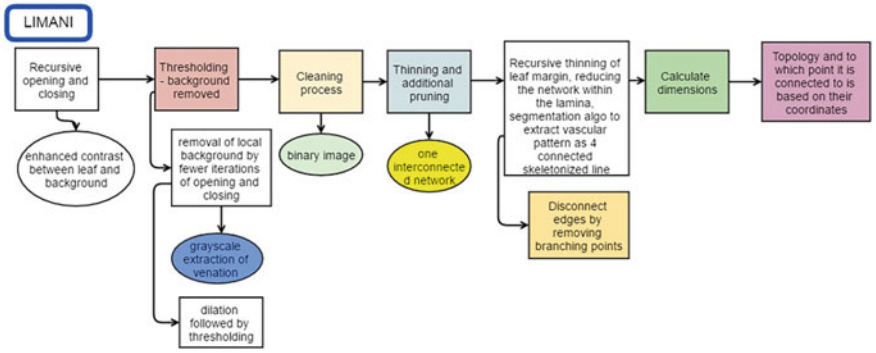


Fig. 1 Summarized methodology of LIMANI [8]

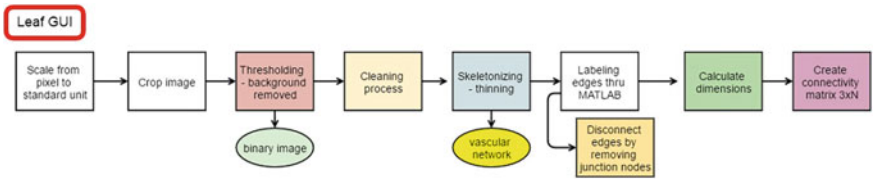


Fig. 2 Summarized methodology of LEAF GUI [6]

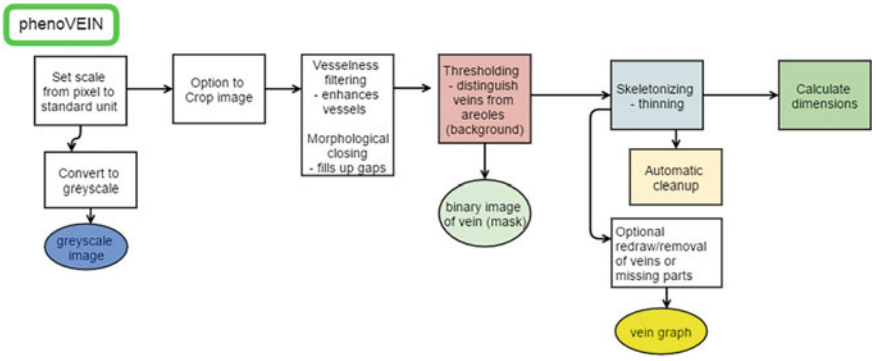


Fig. 3 Summarized methodology of phenoVEIN [7]

traits by PhenoVEIN include skeleton length, projected leaf area, number of veins, and number of areoles [7]. These characteristics are used to describe the plant leaf and venation as the measures used in plant taxonomy. For the goal of this study, which is to extract graphs from the venations, the applications mentioned above lack this step as well as computing metrics from the graphs.

Another study of similar work on extracting graphs from venations uses Canny edge detection, a midrib detection process, vein filling method, and the Zhang-Suen method for skeletonization [10].

The workflow in this study proposes a simplified process wherein, compared to the applications mentioned above, fewer steps are used to extract the venations of the leaves using image processing. This study also presents another metric for leaf venations which is the graph properties of undirected graphs. The venations extracted from the images are converted to a graph $G(V, E)$, where V is a finite set of n vertices $\{v_1, v_2, \ldots, v_n\}$ and E is the set of m edges $\{(v_i, v_j)|v_i, v_j \in V\}$. Metrics for undirected graphs are then computed as another method to describe leaf venations of different species of plants.

3 Methodology

3.1 Image Acquisition

A set of leaves were collected from five plant species selected according to their visibility of leaf venations. A plant taxonomist identified the samples and these plants are namely *Bougainvillea spectabilis* (BS), *Capsicum* sp. (CA), *Cassia* sp. (CS), *Durata erecta* (DE), and *Dahlia* sp. (DS). The leaves were photographed under ample lighting using a DLSR camera with a setting of ISO 200, focal length of 24 mm, and no flash. Using a normal photograph of a leaf under a bright light has proved to hardly capture the details of the venations. Instead, the light had to be supplied underneath the leaf and was achieved using a light pad where the leaves are placed revealing the veins more clearly. The proposed workflow of this study from image segmentation to the extraction of graph metrics is shown in Fig. 4.

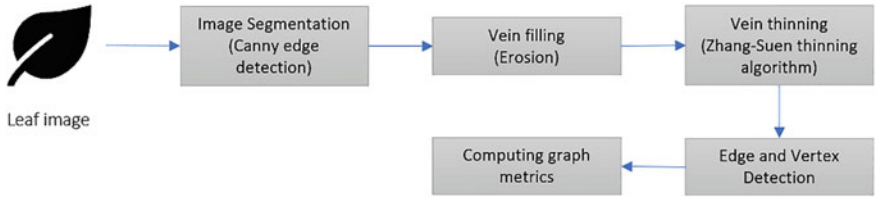


Fig. 4 Summary of the proposed simplified workflow from image segmentation to extraction of graph metrics

There were a total of 30 leaf photographs from five dicot plant species used in this study. Leaf images were selected depending on how well the contrast of the venations is compared to its background. Seven photos each of *Cassia* sp., *Durata erecta*, and *Dahlia* sp. plant species were chosen, while five photos were of *Bougainvillea spectabilis* and four photos were chosen from *Capsicum* sp. leaves. Since the venations of interest are only up to the third degree, the photographs were captured using a low resolution of 72 pixels per inch for faster processing of the algorithms.

3.2 Preprocessing

The raw photos of the leaves had a pixel dimension of 5184 × 3456 pixels (width × height) including largely a white background. The images were cropped to remove some parts of the background reducing the dimension to a width of 800 pixels to an approximate height of 500 pixels. To capture the overall graph, the whole leaf including the boundaries is included in the proposed methodology. Before the image segmentation process of Canny edge detection is applied, the background of the leaf images was first removed, after which the image is converted to a grayscale image.

3.3 Ground Truth (GT) Tracing

The veins of interest were traced manually using a pen tablet (XP-PEN) and an image-editing software (Adobe Photoshop 2023). Using a one-pixel size setting, the digital pen traces the veins of interest which are the primary veins, secondary veins, and tertiary veins. Tracing of one leaf took 30–60 min depending on the complexity of the venation. Another person aside from the tracer reviewed the output for possible editing. The digitized file is saved in a.jpeg format for comparison with the extracted image from the proposed workflow.

3.4 Vein Extraction and Graph Conversion

In the computational workflow for vein extraction, there are two major processes involved. First is the edge detection and second is the skeletonization process. This study utilizes the Canny edge detection algorithm [11] for the first process as it shows superior results best fit for this study [12–14]. Using the OpenCV library and the Java programming language, the photographs had to be converted to grayscale images before subjecting them to the Canny edge detection algorithm. The output image is the boundaries of the leaf and the venations. However, some of the leaf venations are too thick that instead of being detected as one vein, they are detected

as two parallel veins. In order to solidify the veins, a morphological operation called erosion was used. In erosion, the darker areas grow in size [15].

The second process is skeletonization where the solid veins including the boundaries of the leaf are converted to one-pixel width. In this step, Zhang-Suen's thinning algorithm [16] was utilized. Zhang-Suen is a thinning algorithm used mostly for character recognition in documents and is also known for its high performance, time-wise [17]. Specifically, this thinning algorithm aids in the clarification of curvature of scripts such as in Arabic texts [18] and Balinese scripts [19]. For this study, Zhang-Suen is used to clearly define the skeleton of the leaf venations to one-pixel size.

From the one-pixel width of the leaf venations including the boundaries, there are two objectives for the graph conversion method. The first is to determine which are vertices, or the intersection of the line segments with different slopes, and the second is to determine the edges which are connected to the vertices. In determining the vertices, each pixel and its immediate surrounding neighbors are compared to a set of 3×3 pixel-size patterns marked as vertex templates. If the pixels match any of the patterns, they will be added to the vertex list. Figure 5 shows the detected vertices marked by circles and saved in a text file. The edges are then determined by traversing from one vertex point following the connected lines to the vertex until it reaches another vertex point. The path is then stored as an edge including the pixels in its path. Metrics from graphs are then computed using the vertex and edge lists. Table 1 shows the graph properties used to compute the metrics of the extracted venations.

A limitation of the proposed workflow is that the threshold value of the Canny edge detection must be adjusted to include the needed venations from the extracted images. In this case, the threshold varies from 150 to 180. Also, the quality of the images is crucial such that if the input leaf images are blurry or the venations are not contrasting with its background, the workflow will not be able to extract the venations properly.

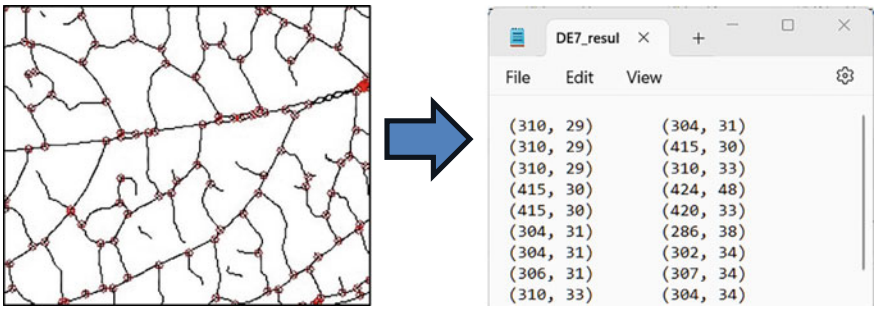


Fig. 5 Detected vertices are marked by circles. The vertices are saved in a text file as pixel points (x, y)

Table 1 Graph properties used as metrics of the converted undirected graphs of leaf venations [20]

Property	Description
Size	$n = \|V\|$
Volume	$m = \|E\|$
Number of wedges, 2-stars or 2-paths	$s = \sum u \in V \binom{d(u)}{2}$, where $d(u)$ is the degree of any vertex u
Number of claws or 3-stars	$z = \sum u \in V \binom{d(u)}{3}$, where $d(u)$ is the degree of any vertex u
Maximum degree	$d_{max} = max_{u \in V} d(u)$

4 Results

The proposed workflow in this study is able to identify the venations of leaves up to the veins of interest using the images used for this experiment. The veins of interest are the order of venations from first (1°) up to the third degree (3°). The parameters for the algorithm for Canny were adjusted accordingly to control the level of detail for the detection process. The range of threshold is from 150 to 180 using the default kernel in OpenCV. The image sizes were of 800 × ~ 500, width by height pixels with 72 ppi. Figure 6 shows the results of the different processes involved in the proposed workflow from Canny edge detection result, eroded image, and the Zhang-Suen thinning result. Venation extraction to vertex detection takes an average of 10 s per leaf running in an Intel Core i7, 16GB RAM 64-bit Windows 11.

4.1 Performance Analysis

The performance of the proposed workflow of venation image extraction can be measured by its correctness and completeness as defined by three criteria, the true positive (TP), false negative (FN), and false positive (FP) [21]. True positive is the pixels that are both in the extracted image and the basis image or ground truth. False positive is the pixels detected in the extracted image but is not in the ground truth image, while false negative is the pixels that are in the ground truth image but is not detected in the segmented image. Correctness (Cor) and completeness (Com) are defined as:

$$\text{Com} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{1}$$

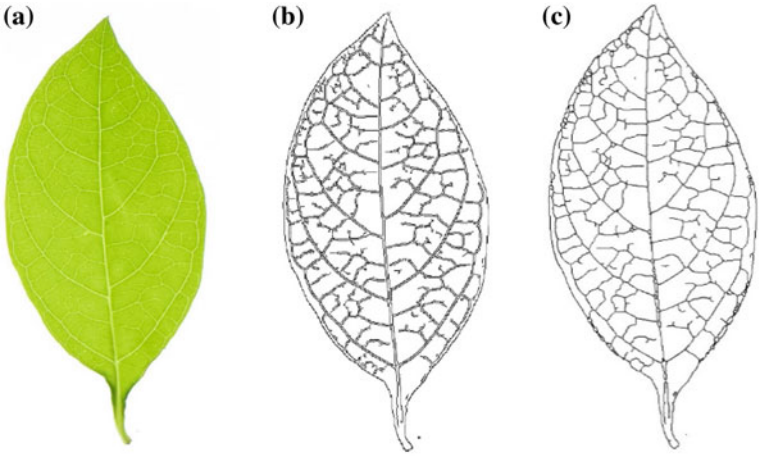


Fig. 6 **a** Photograph of leaf of *Durata erecta*, **b** output of the Canny edge detection, **c** image after the skeletonization using Zhang-Suen thinning algorithm

$$\text{Cor} = \frac{TP}{\text{TP} + \text{FP}}, \tag{2}$$

with values ranging from 0 to 1 and with 1 as the optimum value.

Using the metrics of correctness and completeness, results show that using the proposed workflow, the extracted images revealed a total average of 0.522 for correctness and 0.406 for completeness. The results are not as high given the resolution of the input images of only 800 × ~ 500. The aim of the algorithm was to extract the higher-order veins in order to convert the venations into an undirected graph which also includes the boundaries of the leaf. Given also that the lower-order veins were of the vein of interest, the relatively low completeness values are expected. Table 2 shows the results of the completeness and correctness of all leaf samples.

4.2 Graph Metrics Results

Using the formula in Table 1, the values of each metric for each leaf were grouped according to plant species. The result was tabulated and compared. The results show a difference in the computed graph metrics from the different plant leaf venations for each plant species. The graph properties that gave significant differences among the plant species are size (n), volume (m), number of wedges (s), and number of claws (z). The graph metric maximum degree was disregarded as all graphs revealed had a maximum degree of 5. Figures 7 and 8 show the result of the size (n) and volume (m) of graphs taken for each plant species. The results for the number of wedges (s) and number of claws (z) are also presented in the charts in Figs. 9 and 10.

Table 2 Correctness and completeness values of the extracted veins using the proposed workflow for 30 leaf images

Extracted vein image	Correctness	Completeness
BS1	0.513782016	0.433073515
BS2	0.432817885	0.416735799
BS3	0.51902464	0.405329242
BS4	0.548636149	0.406370157
BS5	0.533449879	0.411577396
CA1	0.550867744	0.416515837
CA2	0.532643312	0.405485264
CA3	0.515320031	0.418625251
CA4	0.53071195	0.414908363
CS1	0.575009092	0.395724267
CS2	0.576441763	0.402168918
CS3	0.596319468	0.413250127
CS4	0.572447272	0.396338916
CS5	0.577792286	0.422985717
CS6	0.580717949	0.416684384
CS7	0.570981996	0.411857453
DE1	0.593535011	0.336413653
DE2	0.581963822	0.370073656
DE3	0.588737154	0.362387042
DE4	0.575356265	0.366352883
DE5	0.564749897	0.379045811
DE6	0.587273128	0.384006253
DE7	0.601047469	0.39346225
DS1	0.555439011	0.436750404
DS2	0.537365964	0.418011243
DS3	0.531433765	0.44434346
DS4	0.510484066	0.446662933
DS5	0.515510593	0.427549902
DS6	0.536567181	0.40304408
DS7	0.553917208	0.433935463

There is a consistent pattern for the five plant species using the graph metrics of size (n), volume (m), number of wedges (s), and number of claws (z). DS has the highest range of values in all four metrics with its maximum value as the highest compared to other four plant species. CA has the smallest range in all metrics as well as the lowest values. It is also observed that the same pattern occurs in all metrics with CA getting the lowest values and DS having the highest values. The pattern is consistent when in terms of range of values in increasing order: CA, DE, CS, BS, and DS.

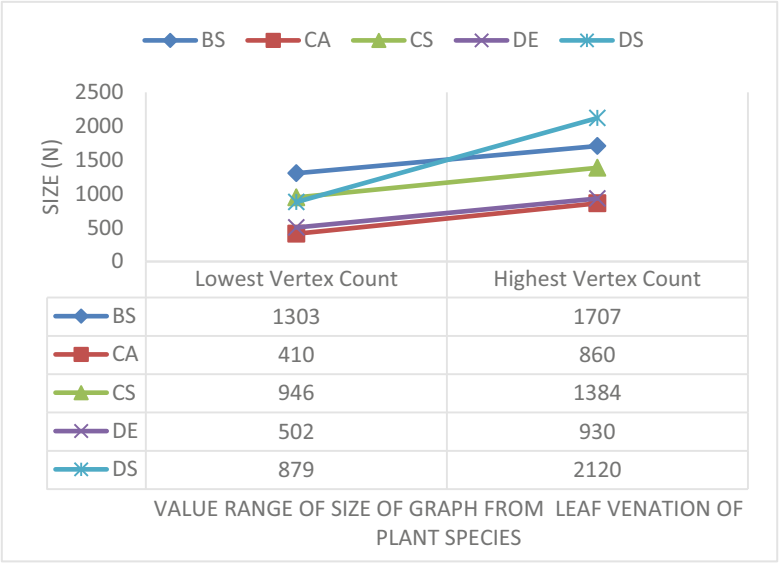


Fig. 7 Range of values of graph sizes (n) of leaves per plant species

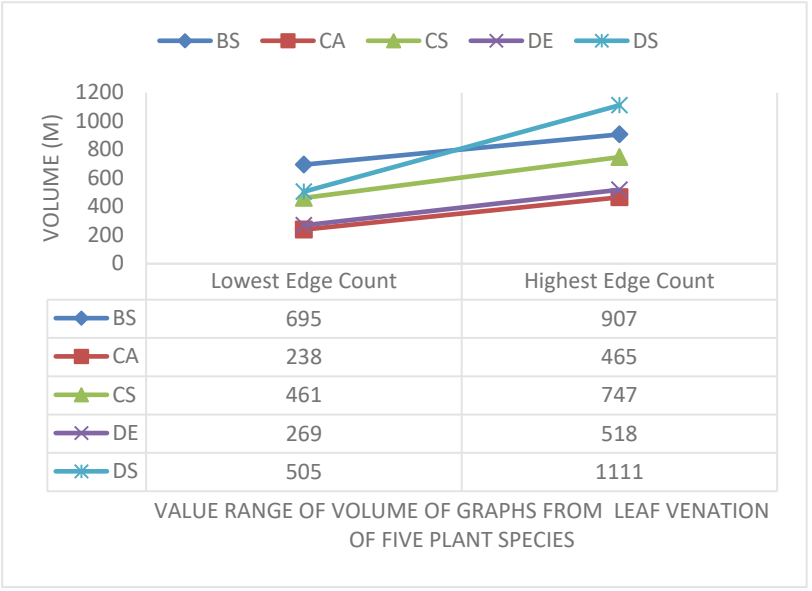


Fig. 8 Range of values of graph volume (m) of leaves per plant species

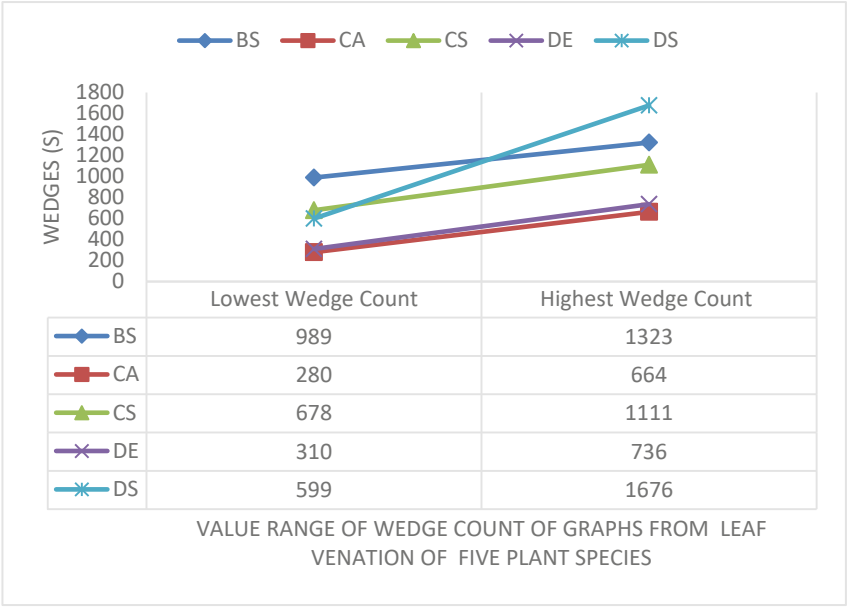


Fig. 9 Range of values of graph wedge count (s) of leaves per plant species

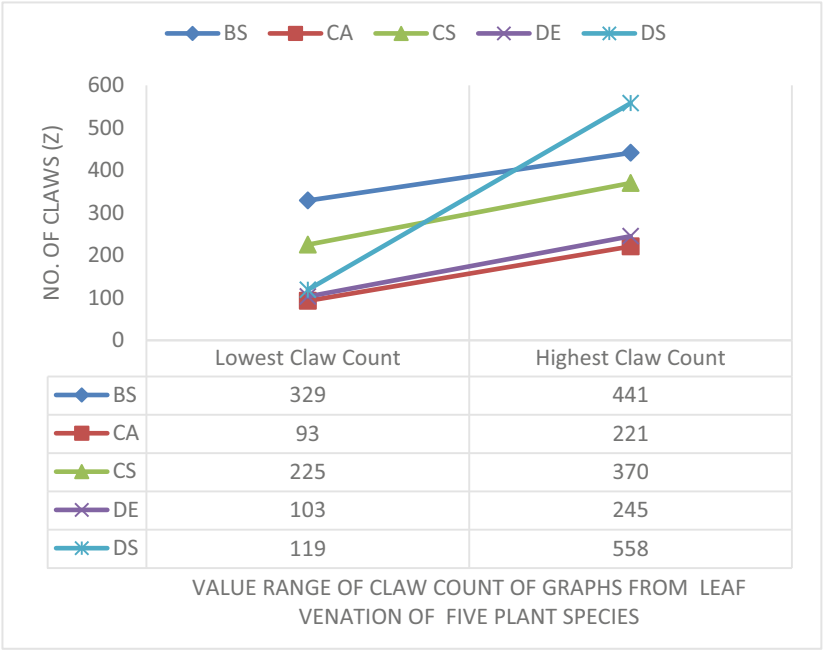


Fig. 10 Range of values of graph claw count (z) of leaves per plant species

5 Conclusion and Recommendation

The proposed workflow presented in this study to extract the venation of plants has lesser steps involved compared to the other known software, making the computation lesser and hence, faster. The use of the algorithms Canny edge detection and Zhang-Suen thinning algorithm is sufficient to segment the venations up to the venation of interest which is the third degree (3°) from the leaf images. The quality of the input photographs greatly affects the quality of the detection method such that the more distinct the venations are from the background, the better the performance of the proposed workflow. The undirected graph (G) was successfully extracted from the segmented venation images using the proposed workflow. Although the correctness and completeness of the extracted venations can still be improved, this study has shown that patterns can be revealed from the graph properties when converting the leaf venation to undirected graphs. These metrics can be added to the other leaf venations metrics for plant characterization. Other graph metrics can also be used as additional measures for pattern analysis. The proposed workflow can also be utilized in similar vein images where graphs are needed to be extracted. This study can also be improved by adding more sample datasets and with higher image resolutions. With more datasets, the graph metrics can be further analyzed using machine learning to identify patterns.

References

1. Jones Jr S, Luchsinger A (1979) Plant systematics. McGraw-Hill, pp 222–232
2. Sack L, Mckown A, Frole K, Scoffoni C, Rawls M, Havran J, Tran H, Tran T (2012) Developmentally based scaling of leaf venation architecture explains global ecological patterns. Nat Commun 3
3. Fuller DQ, Hickey LJ (2005) Systematics and leaf architecture of the Gunneraceae. Bot Rev 71:295–353
4. Blonder B, Enquist BJ (2014) Inferring climate from angiosperm leaf venation networks. New Phytol 204:116–126
5. Hickey LJ (1973) Classification of the architecture of dicotyledonous leaves. Am J Botany: 17–33
6. Price C, Symonova O, Mileyko Y, Hilley T, Weitz J (2011) Leaf extraction and analysis framework graphical user interface: segmenting and analyzing the structure of leaf veins and areoles. Plant Physiol 155
7. Bühler J, Rishmawi L, Pflugfelder D, Huber G, Scharr H, Hulskamp M, Koorneef M, Schurr U, Jahnke S (2015) PhenoVein—a tool for leaf vein segmentation and analysis. Plant Physiol: 2359–2370
8. Dhondt S, Van Haerenborgh D, Van Cauwenbergh C, Merks RM, Philips W, Beemster GT, Inze D (2012) Quantitative analysis of venation patterns of Arabidopsis leaves by supervised image analysis. Plant J 69:553–563
9. Dirnberger M, Kehl T, Neumann A (2015) Nefi: network extraction from images. Sci Rep 5
10. Balangcod AD, Pabico JP (2020) Automatic identification of selected dicot plant species using graph properties from leaf venations. In: 20th Philippine computing science congress
11. Canny JF (1986) A computation approach to edge detection. IEEE Trans Pattern Anal Mach Intell PAMI-8 6:679–698

12. Kaur S, Singh I (2016) Comparison between edge detection techniques. Int J Comput Appl 145:15–18
13. Acharjya P, Ritaban D, Dibyendu G (2012) Study and comparison of different edge detectors for image segmentation. Glob J Comput Sci Technol
14. Shin M, Goldgof DB, Bowyer K, Nikiforou S (2001) Comparison of edge detection algorithms using a structure from motion task. IEEE Trans Syst Man Cybern Part B (Cybern) 31:589–601
15. Said KAM, Jambek AB (2021) Analysis of image processing using morphological erosion and dilation. J Phys Conf Ser 2071(1)
16. Zhang TY, Suen C (1984) A fast parallel algorithm for thinning digital patterns. Commun ACM 27(3):236–239
17. Widiarti AR (2011) Comparing Hilditch, Rosenfeld, Zhang-Suen, and Nagendraprasad-Wang-Gupta thinning. Int J Comput Inf Eng 5(6):563–567
18. Saudagar AKJ, Mohammed HV (2016) OpenCV based implementation of Zhang-Suen thinning algorithm using Java for Arabic text recognition. Inf Syst Des Intell Appl Proc Third Int Conf India 3:265–271
19. Sudarma M, Sutramiani NP (2014) The thinning Zhang-Suen application method in the image of Balinese scripts on the Papyrus. Int J Comput Appl 91(1):9–13
20. Kunegis J (2013) KONECT — The Koblenz network collection. In: Proceedings of international web observatory workshop, 1343–1350
21. Heipke C, Mayer H, Wiedemann C, Jamet O (1997) Evaluation of automatic road extraction. Int Arch Photogramm Remote Sens 32(3 SECT 4W2):151–160

Multispectral Fusion of Multisensor Image Data Using PCNN for Performance Evaluation in Sensor Networks



S. Dharini and Sanjay Jain

Abstract This research study proposes a Multispectral Fusion of Multisensor Image Data using PCNN by exploring the image visual quality and the object detection process. Initially, the collected IR and VIS images are preprocessed using a sampling, filtering and resizing process. Then, the contrast enhancement is done on the preprocessed images to improve the visual quality of preprocessed image. While performing image fusion, utilizing the less block size image has lowered the functioning of fusion process; thus, image resizing is applied before employing the DCT technique. Here, the pixels of IR and VIS are further improved by PCNN that intentionally performs to increase the quality of the pixels by capturing the local information of fused images. Further, IDCT is also employed to decompose the fused images. Finally, the object presented in the image is detected and the metrics like SNR and PSNR are used to prove the efficiency of the fused image.

Keywords Object detection · Multimodal sensor images · Fusion process · PCNN · PSNR · SNR

1 Introduction

The technological innovations and developments provide numerous benefits in the form of digital applications. Specifically, in the field of object recognition, the complexities in detecting multiple objects are greatly reduced. However, the intrinsic and extrinsic characteristics make the recognition process more challenging. Further to overcome the challenges, existing research works are analyzed in detail. The major

S. Dharini (✉) · S. Jain
CMR Institute of Technology, Bengaluru, India
e-mail: dhariniraghav@gmail.com

S. Jain
e-mail: Principal@cmrit.ac.in

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2023

S. Shakya et al. (eds.), *Fourth International Conference on Image Processing and Capsule Networks*, Lecture Notes in Networks and Systems 798,
https://doi.org/10.1007/978-981-99-7093-3_9

characteristics of a mining system are efficiency, and it must be attained by considering the storage features, processing metrics, feature extraction procedures, feature indexing and pattern retrievals [1]. Based on these characteristics, various mining models are evolved in the past decade and are mainly categorized into function-based mining and information-based mining. Further, the function-based mining is widely analyzed by the research community as it provides the important features of data. In this category, numerous clustering and classification models are evolved to attain better image analysis results. Specifically, the classification models are further categorized as unsupervised and supervised models [2]. In the supervised classification process, existing labeled images are used to group the unlabeled images. In case of unsupervised classification process, depending on the content or feature of the images, different groups are created [3]. Various techniques are developed under unsupervised category and supervised categories infer the essential information from the image. Specifically, low-level images are used in the mining process so that the high-level objects can be extracted [4]. Generally, the information will be extracted by four levels, namely, pixel, semantic concepts, object and pattern levels [5]. The features of deep learning algorithms are significantly proved in various classification and pattern recognition applications. Various deep learning models are evolved as recognition models, which imply the increased utilization of human computer interactions. One of the significant aspects of human computer interaction is emotion-related data, which helps to build an effective communication module. Similarly, various image processing applications are evolved based on deep learning algorithms. Deep learning approaches and existing learning models extract information or feature from a single input image. However, to obtain detailed features to enhance the classification or recognition performance, fused images are widely preferred. The process of combining multiple images as single image is termed as image fusion. Rich information can be obtained from the fused image so that it assists all phases involved in image processing and analysis tasks like object detection, image classification, image segmentation and so on. It also helps for better interpretation of processed images that covers the real-time applications like agriculture, head pose tracking system, surveillance applications, weapon detection, etc. [6]. The performance of fused images relies on two factors, namely, image decomposition algorithm and choice of fusion rule. It includes many properties such as redundancy, directional selectivity, shift variance, lossless reconstruction and preservation of edges. The fusion rules must be designed in a way that it should merge the information without any loss and without introducing spectral and spatial distortion in fused images. This research work presents a detailed analysis of image fusion process and validates the efficiency of fusion in object recognition application. This research study is organized as follows. Section 2 presents the related work that describes the merits and demerits of the prior techniques. Section 3 provides the proposed work, and simulation results are presented in Sect. 4. Finally, Sect. 5 concludes the proposed research work.

2 Related Work

In order to gain a broad knowledge on the existing image fusion techniques, a brief literature analysis is presented in this section by considering the methodology and its features. Multiple sensors with multiband and multimode capabilities were studied, and the process of data selection, preprocessing, registration and the fusion methodology are employed in multiple sensory applications and also analyzed. In general, Visible (VIS) and Infrared (IR) images are combined in the fusion process to obtain rich information content. Object location and thermal radiations can also be captured in IR images [7], but it is difficult to obtain more details from IR images due to its low-resolution features. In the case of VIS image, significant details can be obtained with high resolution, but due to environmental factors, the VIS images will lose some features. Thus, IR images are combined with VIS images using image fusion process to obtain the low- and high-level features.

The conventional IR and VIS image modeling involves Sparse Representation (SR)-related methods and the Multiscale Transform (MST)-related methods. The MST-related methods presented the decomposing model to the source images with the multiscale representations with the fusion rule; further by applying inverse transform on fused multiscale representations, fused images are obtained. Some methods like wavelet transform, Pyramid transform, non-subsampled contourlet transform (Hu et al. 2017) and MST methods are employed in image fusion process. [7], have designed a convolution-based Sparse Representation model to decompose the image fusion methods. However, the presented model enhances the fused image accuracy wherein the quality of image is reduced after the fusion process. An activity level measurement is addressed by [8] to generate fusion rules by using deep convolutional neural networks. Here, pixel-based activity information was shared among the users using a weighted map approach [9]. Due to the discontinuity variable, the weighted approach could not match the shared information during the retrieval process. In [10], a deep learning framework was designed to explore the efficiency of content-based medical image retrieval processes. This model takes high computational time to retrieve the relevant content. In continuation with the above study, [11] presented a method to identify the defocused and focused pixels in image and performed pixel-level fusion by using convolutional neural network. Here, the edges of the pixels are not properly detected, which results in creating overlapping issues during the fusion process. Li et al. [12, 13], introduced a fusion layer in deep CNN, wherein each block is connected with the output of the other layers. Though the fusion layer has properly fused the images, the decomposition part of the fused image was not remarkable. Similar study was extended by [14] for engaging more on deep features and zero-phase components to develop a fusion process. Since the components have reduced the sensitivity of the image enhancement, it is not suitable for real-time applications. A multimodal medical image fusion using deep CNN and multiscale transformation-based feature process were studied by [15]. This has proved the efficiency rate of the fused images, however, the interpretation of fused images becomes a complex task.

3 Research Methodology

The proposed Pulse-Coupled Neural Network (PCNN) approach for multispectral fusion of multisensor data is presented in this section. Here, a simultaneous Infrared image and Visible images are employed to fuse the images via Discrete Cosine Transform (DCT)-Pulse-Coupled Neural Network (PCNN). The followings are the proposed steps:

3.1 Image Preprocessing

It is the initial step that describes the details of information collection and the validation of preprocessing techniques based on the collected data. A pixel sampling is performed on the IR and VIS images in which the pixel values are replaced with median value of the neighbor pixels. The medial value is obtained by sorting the pixel values. A window slides over the entire image pixels and replaces it with median pixel value. After sampling, Gaussian filter is applied to remove the noise and enable image smoothening. Generally, convolution operators are termed as Gaussian operators, which smoothens the image. The parameters of Gaussian filters are standard deviation σ and window dimension. If the standard deviation value is high, then it requires high smoothing process.

3.2 Image Enhancement

The quality of the preprocessed image is further enhanced by applying various techniques, namely, contrast enhancement, histogram equalization and adaptive histogram equalization. Initially, the adjustment of the images is done, and then, it is followed by an equalization process.

(a) **Histogram Equalization**

To improve the image contrast, histogram equalization is employed to alter the intensity values to provide contrasted image. Histogram equalization is an effective and convenient image enhancement process. It is the technique used to develop a linear cumulative histogram value of the given input images. Based on the obtained histogram values, the pixel values are redistributed according to the intensity range. The conventional histogram equalization technique is described below:

Let us assume an input image, $I(p, q)$ with an aggregate of n pixels, in which the gray level ranges from $\left[X_0, X_{N-1}\right]$. The probability density function $P(r||k)$ for the image level r_k is given as,

$$P(r_k) = \frac{n_k}{n}, \tag{1}$$

where

n_k count of the r_k in the image.
n aggregate number of pixels with $k = 0, \ldots (N - 1)$.

Finally, the histogram of the image is estimated by the arrangement of n_k against the r_k. Owing to it, the altered intensity is estimated by the cumulative functions which is given as,

$$C(r_k) = \sum_{i=0}^{k} P(r_i). \tag{2}$$

It helps to arrange the image under dynamic range, $\left[X_0, X_{N-1}\right]$ which is again elaborated as,

$$f(X) = X_0 + (X_{N-1} - X_0)C(X). \tag{3}$$

The above equation has reduced the unequal histograms that have changed the brightness of the image.

(b) **Adaptive Histogram Equalization (AHE)**

At last, a preprocessed image is fed into Adaptive Histogram Equalization (AHE), which assists for obtaining enhanced image quality. It is a spatial domain model that describes the uniform distribution of pixel intensity for better contrast enhancement. The procedure of AHE is as follows:

(a) Input: preprocessed image.
(b) Obtain the image histogram.
(c) Finding the local minima in the histogram.
(d) Image is divided based on local minima.
(e) Allocating specific gray levels to each divided histogram.

By doing so, the histogram partitioning prevents some parts of histogram being dominated by others.

3.3 Image Fusion

To fuse the IR and VIS images, Discrete Cosine Transform (DCT) is applied by combining the pixels in order to provide the fused image. Cosine transform functions combine the VIS and IR images' missing pixels in the fusion process. The cosine functions make use of the Mel frequency range of the pixels and transform it into spatial domain. Further to estimate coefficients, DFT is applied. The final output of DFT provides the Mel frequency cepstral coefficient (MFCC) which is given as follows:

$$C_n = \sum_{k-1}^{k} (\log D_k) \cos[m(k-1)\pi], \tag{4}$$

where $m = 0, 1 \ldots k - 1$; C_n represents the Mel frequency cepstral coefficient, and m indicates the number of coefficients. The two-dimensional input signal is then converted into one-dimensional signal using MFCC and then categorized into frames with the condition ($M < N_m$), where N_m indicates the neighbor frames. This condition reduces the pixel information losses. Further by applying FFT, the frequency domain is attained through windowing process. Finally, Mel frequency is used to obtain spectrum magnitude.

3.4 Image Quality Enhancement

To improve the image quality, Pulse-Coupled Neural Network (PCNN) is employed in the proposed work which is familiar network model formulated based on the cat and monkey pulse bursts. In the proposed method, neurons pulse synchronization and global coupling characteristics are employed. The local information of a VIS and IR images is fused using PCNN. The neurons are geared up by the values of a single pixel under the spatial domain. Thus, the spatial frequency with gradient energy is employed in this study. Here, multimodal-PCNN (m-PCNN) is a 2D-linked neuron which is laterally connected. Depending on the pixels' count, the neurons for the network is defined; similarly, the neuron stimulus is selected based on the gray pixel values. In the neuron internal state, data fusion is performed and it is essential that the input images must have identical resolution. Deviation in the resolutions will reduce the fusion performances. PCNN provides better fusion performance over traditional approaches and preserves the texture and edge features in image which makes the fusion suitable for recognition applications.

3.5 Image Reverse-Fusion Process

To enhance the segmentation process in the proposed work, Inverse Discrete Cosine Transform (IDCT) is applied to extract the foreground objects from the images. General applications utilize DCT for compression, but in the proposed work, DCT is used to segment the object. For object segmentation, background and foreground subtractions are performed in DCT considering the decorrelation attributes. This preserves the features, reduces the computation complexity and enhances the retrieval efficiency. Further to enhance the effectiveness of segmentation, user-defined functions are employed. The steps involved in modified DCT are presented as follows:

- Select initial frames.
- Compare each frame with the first frame.
- Employ binary segmentation for frame subtraction and perform morphological operation to select the object from the image.

Background and foreground subtraction-based IDCT provides enhanced performance over conventional DCT. Finally, the output of IDCT is employed to detect the object presented in an image.

4 Results and Discussion

The proposed model performances are discussed in this section from the beginning to final PSNR and SNR measurements. Two types of images like Infrared image and VIS image are used for the fusion process, and finally, the fused image metrics are measured and analyzed.

Figure 1 presents the input images, wherein source image 1 denotes the IR image and the source image 2 presents the VIS images.

Figure 2 presents the preprocessed output. Here, sampling and filtering processes are applied into the images. The irrelevant noise in IR and VIS images is given.

Figure 3 presents the image enhancement that enhances image visual quality. Also, the pixel-level information has been deliberately performed to improve the visual quality.

Results of fused images are presented in Fig. 4, and to attain better performance, DCT functions are applied to resize the images.

Figure 5 presents the output of the PCNN technique. Here, the quality of the pixels is improved layer by layer.

Figure 6 presents the object detection. From the pixel-quality improved images, the objects present in the image are detected.

Figure 7 presents the performance analysis of the DCT technique by using two metrics, PSNR and SNR.

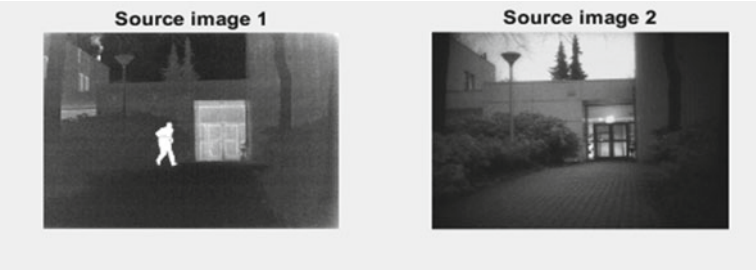


Fig. 1 Input images

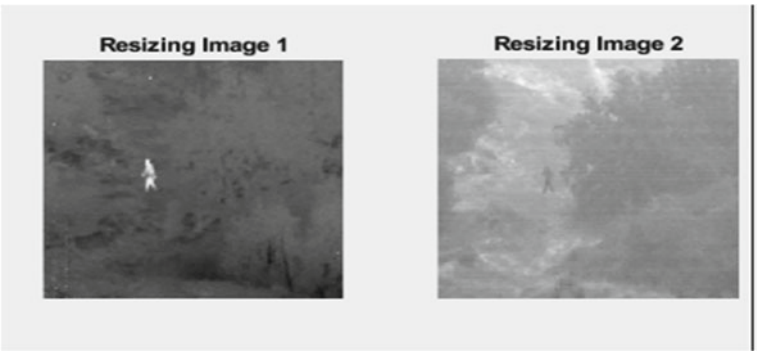


Fig. 2 Preprocessed image



(a)



(b)

Fig. 3 Image visual quality enhancement

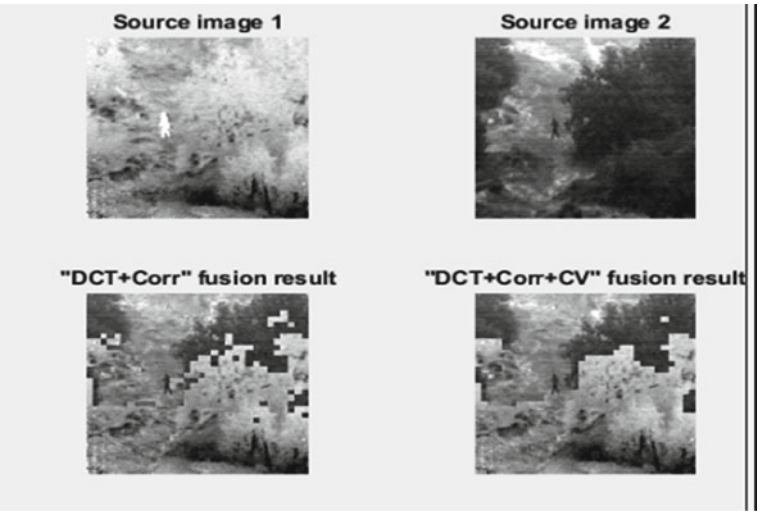


Fig. 4 Image fusion process

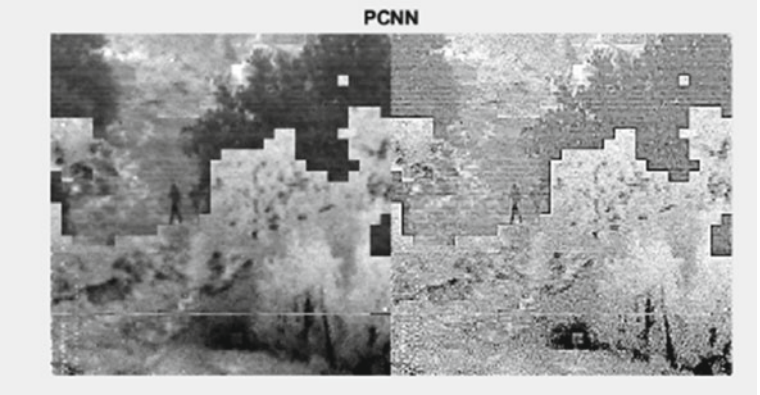


Fig. 5 PCNN process

Figure 8 presents the performance analysis of the IDCT-fusion technique. It is estimated to find the efficiency of the image fusion process.

Table 1 shows the error rate analysis of DCT-fusion technique and IDCT-fusion technique.

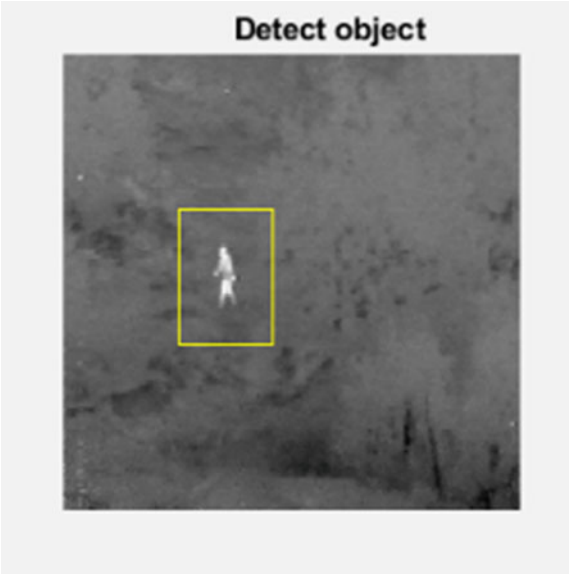


Fig. 6 Detecting the objects

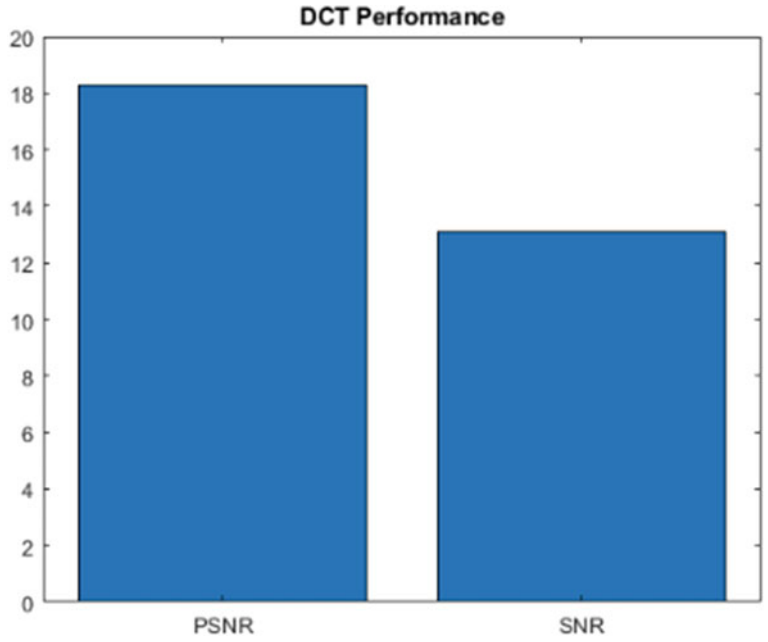


Fig. 7 Performance analysis of the DCT-fusion technique

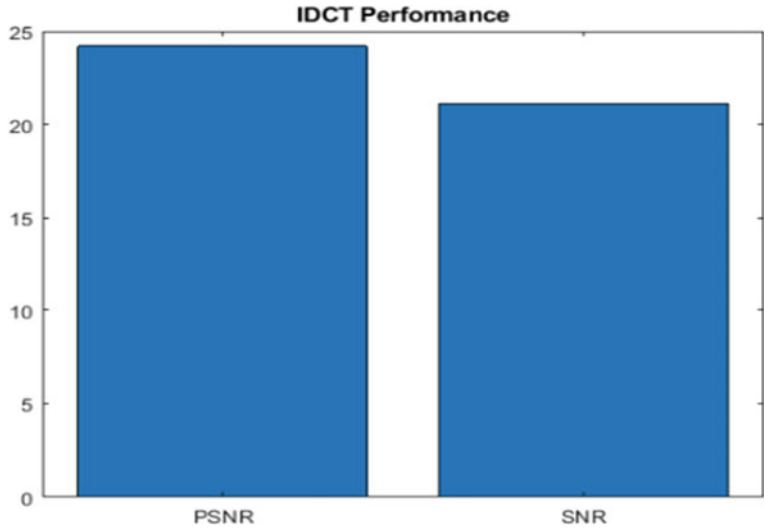


Fig. 8 Performance analysis of the IDCT-fusion technique

Table 1 Error rate analysis

Technique name	Error rate
DCT_PSNR	18.2800
IDCT_PSNR	24.2068
DCT_SNR	13.0999
IDCT_SNR	21.1067

5 Conclusion

In recent times, the analysis of multimodal images has been explored in real-time applications. In specific, object detection in industrial scenes in understanding environments becomes tedious tasks. Image fusion concept is one of the image processing steps focused by several researchers. It accelerates the growth of analyzing multimodal images. Multiple images are combined in the fusion process to provide rich information that assists for all phases involved in image processing and analysis tasks. It also helps in enabling the better interpretation of processed images. The fusion rules are designed to merge the information without any loss and without introducing spectral and spatial distortion in fused images.

References

1. Li X, Orchard MT (2001) New edge-directed interpolation. IEEE Trans Image Process 10(10):1521–1527
2. Zhang X, Wu X (2008) Image interpolation by adaptive 2-d autoregressive modeling and soft-decision estimation. IEEE Trans Image Process 17(6):887–896
3. Liu X, Zhao D, Xiong R, Ma S, Gao W, Sun H (2011) Image interpolation via regularized local linear regression. IEEE Trans Image Process 20(12):3455–3469
4. Guo K, Yang X, Zha H, Lin W, Yu S (2012) Multiscale semilocal interpolation with antialiasing. IEEE Trans Image Process 21(2):615–625
5. Zhong B, Ma KK, Lu Z (2019) Predictor-corrector image interpolation. J Vis Commun Image Represent 61:50–60
6. Bhateja V, Patel H, Krishnan A, Shahu A (2015) Multimodal medical image sensor fusion framework using cascade of wavelet and contourlet transform domains. IEEE Sensor J 25:6783–6790
7. Wang X, Meng J, Liu F (2016) Fusion of infrared and visible image based on compressed sensing and nonsubsampled shearlet transform. Int J Signal Process 9(4):37–46
8. Liu Y, Chen X, Peng H, Wang Z (2017) Multi-focus image fusion with a deep convolutional neural network. Inf Fusion 36:191–207
9. Liu Y, Chen X, Cheng J, Peng H, Wang Z (2018) Infrared and visible image fusion with convolutional neural networks. Int J Wavelets Multiresolution Inf Process 16(03):1850018
10. Qayyum A, Anwar SM, Awais M, Majid M (2017) Medical image retrieval using deep convolutional neural network. Neurocomputing 266:8–20
11. Tang H, Xiao B, Li W, Wang G (2018) Pixel convolutional neural network for multi-focus image fusion. Inform Sci 433:125–141
12. Li J, Song M, Peng Y (2018) Infrared and visible image fusion based on robust principal component analysis and compressed sensing. Infrared Phys Technol 89:129–139
13. Li H, Wu XJ, Kittler J (2018) Infrared and visible image fusion using a deep learning framework. In: 2018 24th international conference on pattern recognition (ICPR). IEEE, pp 2705–2710
14. Li H, Wu XJ, Durrani TS (2019) Infrared and visible image fusion with resnet and zero-phase component analysis. Infrared Phys Technol 102:103039
15. Xia KJ, Yin HS, Wang JQ (2019) A novel improved deep convolutional neural network model for medical image fusion. Cluster Comput 22(1):1515–1527

U-Net-Based Segmentation of Coronary Arteries in Invasive Coronary Angiography



A. Azeroual, Y. El Ouahabi, W. Dhafer, M. H. El yousfi Alaoui, B. Nsiri, and A. Soulaymani

Abstract Coronary artery disease continues to be the primary reason for death, creating a difficult situation for worldwide health strategies. Among the main techniques to diagnose these diseases clinically is invasive coronary angiography. Consequently, the development of a precise deep learning model for coronary artery segmentation becomes imperative for the diagnosis and evaluation of coronary artery disease. This is particularly important to address the challenges arising from issues in coronary angiography, which can adversely affect image quality and hinder cardiologists' medical interpretation. In this study, U-Net model is developed on images from invasive coronary angiography to achieve accurate and effective segmentation. This project's dataset includes 99 patients who underwent invasive coronary angiography systems. The average accuracy of the suggested model was 98% and an intersection over union of 86%. Indeed, the experimental results demonstrated big effectiveness of the U-Net model to help cardiologists in diagnosing this disease.

Keywords U-Net model · Deep learning · Coronary artery disease · Invasive coronary angiography · Binary image segmentation

A. Azeroual (✉) · W. Dhafer · M. H. El yousfi Alaoui · B. Nsiri
Research Center STIS, M2CS, National Graduate School of Arts and Crafts of Rabat, Mohammed V University in Rabat, Rabat, Morocco
e-mail: amal.azeroual@um5r.ac.ma

Group of Biomedical Engineering and Pharmaceuticals Sciences, National Graduate School of Arts and Crafts of Rabat (ENSAM), Mohammed V University, Rabat, Morocco

M. H. El yousfi Alaoui
e-mail: h.elyousfi@um5r.ac.ma

B. Nsiri
e-mail: b.nsiri@um5r.ac.ma

Y. El Ouahabi · A. Soulaymani
Laboratory Health and Biology, Faculty of Sciences, Ibn Tofail University, Kenitra, Morocco
e-mail: soulaymani@uit.ac.ma

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2023

S. Shakya et al. (eds.), *Fourth International Conference on Image Processing and Capsule Networks*, Lecture Notes in Networks and Systems 798,
https://doi.org/10.1007/978-981-99-7093-3_10

1 Introduction

As per the guidelines provided by the "World Health Organization", coronary artery diseases (CAD) are classified as a sort of cardiovascular condition that causes death to 17.9 million person each year [1]. Cardiovascular diseases encompass a range of heart and blood vessel conditions, including CAD, cerebrovascular diseases, and rheumatic heart diseases. More than four out of five deaths from cardiovascular diseases are related to myocardial infarctions and strokes, with approximately one-third of these fatalities occurring prematurely in individuals aged below 70 [1].

CAD affects the human heart by a buildup of cholesterol plaques in the coronary arteries that provide the heart by oxygenated blood. The symptoms of CAD vary from being without symptoms to experiencing stable chest pain, acute coronary syndrome, and sudden cardiac death. Arrhythmias and heart failure are frequently observed complications, while myocardial infarction is the prevailing manifestation of CAD [2]. CAD can be clinically diagnosed using three primary techniques: "Invasive Coronary Angiography" (ICA), "myocardial perfusion imaging", and "drug stress echocardiography". These methods are effective in identifying angina symptoms related to myocardial ischemia [3]. The first technique entails the injection of a contrasting substance into the arteries surrounding the heart, allowing for the visualization of coronary arteries under the supervision of a cardiologist. The severity of the narrowing is manually determined by the physician visual assessment (PVA). Meanwhile, quantitative coronary angiography is considered a reliable computer-assisted procedure for measuring the severity of stenosis [4]. However, it is limited by a busy workflow and the measurement of complex lesions. These limitations have motivated the progress of deep learning (DL) and the convolutional neural network (CNN)-based tools that can rapidly and accurately segment and detect lesions (shown in Fig. 1).

Deep learning technology is utilized by Du et al. [5] in the interpretation of diagnostic coronary angiography, which assists cardiologists in detecting and diagnosing the severity and shape of lesions while conducting the intervention. Rjiba et al. [6] introduced a novel method based on a 3D CNN for the automatic extraction of centerlines of coronary arteries in "Coronary Computed Tomography Angiography"

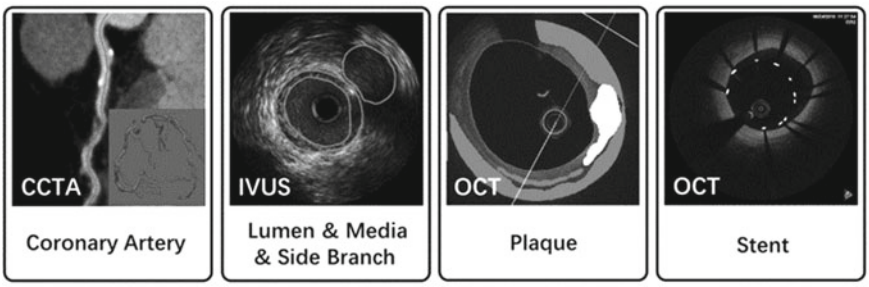


Fig. 1 Example of image segmentation on CAD

(CCTA) data. Gulsun et al. [7] investigated the application of CNN as a postprocessing method to differentiate between genuine CAD and leaks. More recently, Table 1 presents some studies on the utilization of CNN models for the purpose of image segmentation [8].

Also, U-Net model has proven its accurate performance in semantic segmentation of medical images and was applied in various segmentation tasks including retinal segmentation and brain arterial segmentation, that was challenging many aspects of ICA imaging, especially, that they have limited contrast and include blur noise [17]. Furthermore, by leveraging the power of U-Net, the segmentation process becomes more efficient and precise, enhancing the overall interpretation and analysis of ICA images. U-Net model requires a minimal number of annotated images and can easily be applied to various biomedical images' segmentation tasks thanks to the training process, which could be completed within a reasonable timeframe of up to 10 h using an "NVidia Titan GPU" (6 GB).

Long et al. constructed a U-Net network by drawing upon the principles of fully convolutional networks in their research [18]. Dorgham et al. present U-Net Computed Tomography Segmentation, an innovative automated DL segmentation framework tailored for computed tomography images. The primary objective is to merge the DL U-Net approach with computed tomography images, facilitating automatic segmentation within the realm of medical image analysis [19].

The subsequent sections of this paper are structured as follows: Sect. 2 presents a comprehensive review of the pertinent literature in the field. Section 3 of the document details the proposed approach, while in Sect. 4, the outcomes achieved were showcased and examined using the suggested models. Finally, the concluding section summarizes the key findings and implications of the study.

Table 1 Application of CNN models in image segmentation of CAD

Study	Year	Modality	Clinical application
Li et al. [9]	2022	CCTA	Stenosis diagnosis
Du et al. [10]	2022	IVUS	Plaque identification
Lin et al. [11]	2022	CCTA	Prediction of cardiovascular risk
Hong et al. [12]	2022	OCT	Risk stratification
Zhao et al. [13]	2021	XA	Stenosis diagnosis
Tu et al. [14]	2021	XA	FFR computation
Yu et al. [15]	2021	IVUS	FFR computation
Chu et al. [16]	2021	OCT	Plaque characterization

IVUS intravascular ultrasound, *OCT* optical coherence tomography, *XA* X-ray angiography

2 Related Work

2.1 Medical Imaging Works for Coronary Arteries

Medical image analysis involves the identification of specific informative details within an image or the classification of the image itself to address clinical issues. In the existing literature, there have been significant advancements in coronary artery analysis, through the utilization of various image processing techniques, including level-set, graph-cut, Gaussian distribution of lumen, and active contour methods. These methods typically necessitate crucial preprocessing and postprocessing steps [20].

CNN techniques have been extensively utilized in the realm of cardiology for medical imaging purposes, spanning multiple imaging modalities. Moeskops et al. [21] have developed a unified CNN capable of segmenting six different tissues in Magnetic Resonance (MR) images. Their approach extends to segmenting the pectoral muscle in MR breast images as well as the CAD. In other studies, Han et al. [22] carried a study to evaluate the practicality of utilizing DL methods for analyzing CCTA images concerning the detection of coronary artery stenosis. They conclude that the diagnostic accuracy of CCTA-Artificial Intelligence surpasses that of traditional CCTA.

2.2 Image Segmentation with U-Net

In recent years, DL is extensively employed to deliver precise image analysis across diverse medical domains. The key attribute of DL models lies in their ability to extract features, enabling them to provide accurate and distinct descriptions of different types of tissues. Medical image segmentation is a highly demanding undertaking in the realm of medical image analysis, as it endeavors to extract valuable data and enhance the accuracy of clinical diagnosis [23].

CNN techniques have exhibited exceptional levels of accuracy and dependability in numerous applications in computer vision, like image classification, object detection, and semantic segmentation. Previous research utilizing DL methods in the context of medical imaging in cardiology has predominantly concentrated on tasks related to segmentation. Alakwaa et al. have introduced a novel automated approach that utilizes a three-dimensional CNN (3D CNN) for the detection of lung cancer. Their method focuses on segmenting nodules from three-dimensional computed tomography (CT) scans [24].

Similarly, numerous techniques utilizing U-Net and its variations have been suggested to accomplish automated process of segmenting medical images. Recently, Baccouch et al. conducted a comparative analysis between U-Net and standard CNN models for the segmentation of medical images. Their study revealed that U-Net is better suited in the context of medical applications, specifically for segmentation tasks

in the field of health care, when compared to other CNN models [23]. Ronneberger et al. have introduced an innovative framework named U-Net, which is a significant architecture for semantic segmentation in CNN. U-Net demonstrates remarkable performance for tasks involving pixel-level predictions [25]. Chen et al. have introduced TransUNet, a novel variation of the U-Net architecture. TransUNet combines transformers with U-Net to address the limitations of U-Net in explicitly capturing long-range dependencies, which can be challenging due to the inherent localized nature of convolution operations in tasks involving medical image segmentation [26]. In other research, Zhou et al. have introduced an enhanced architecture, U-Net++, for the task of segmenting medical images. Unlike the standard U-Net architecture, U-Net++ is a neural network that establishes interconnectedness between the decoder and encoder sub-networks by incorporating a series of nested, dense skip pathways. These pathways are designed to minimize the semantic gap between the feature maps of the encoder and decoder, resulting in a more powerful framework for accurate segmentation of medical images [27]. In summary, U-Net models have been used in many studies for the task of segmenting medical images, and in the rest of the paper, the model based on U-Net was presented.

3 Materials and Methods

3.1 Dataset

The dataset employed in this project was collected from 99 patients' coronary angiography imaging (Github), at the "Jiangsu Province People's Hospital" in China, an interventional angiography system (AXIOM-Artis, Siemens, Munich). The X-ray images were captured at a rate of 15 frames per second and each one with a size of 215×215 pixels. Expert cardiologists have annotated the dataset of left coronary and right coronary artery images. A total of 616 images, along with their corresponding masks, were collected for this study and 512×512 of size divided into 80% for training and 20% for testing.

3.2 Method

The main purpose is to train a U-Net model employed to perform segmentation of coronary arteries in imaging using ICA, which will be used in the next step in the assessment of stenosis and its classification after using oriented methods for the detection of stenosis. The model was applied on both left coronary artery (LCA) in Fig. 2 and right coronary artery (RCA) images in Fig. 3 in order to segment vessel tree. In the overall, the model was implemented utilizing TensorFlow/Keras and took about 15 min in training achieving training and a validation accuracy of 98%.

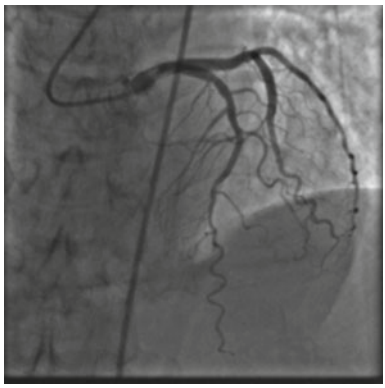


Fig. 2 Left coronary artery (LCA) X-ray image



Fig. 3 Right coronary artery (RCA) X-ray image

U-Net Architecture. U-Net model includes an encoder and decoder path. The encoder path adheres to CNN architecture standards by including repetition 3 × 3 convolution layers followed by max-pooling functions and rectified linear unit (ReLU). A kernel of 3 × 3 size was applied across the 512 × 512 input for feature extraction with a total of 16 filters used (shown in Fig. 4). Then, a max-pooling layer got introduced between layers with the aim of down-sampling the input in weight and height directions using a stride of 2 according to the following formula:

$$\text{Floor}((W - F)/S + 1), \tag{1}$$

where W represents the input x or y shape, F the pooling window size, and S stands for the stride. Conversely, the next path of the U-Net model is the decoder one using transposed convolution to up sample the output of the encoder path with both Kernel and stride of size two, according to the following formula:

$$Z = (W - 1)S + K, \tag{2}$$

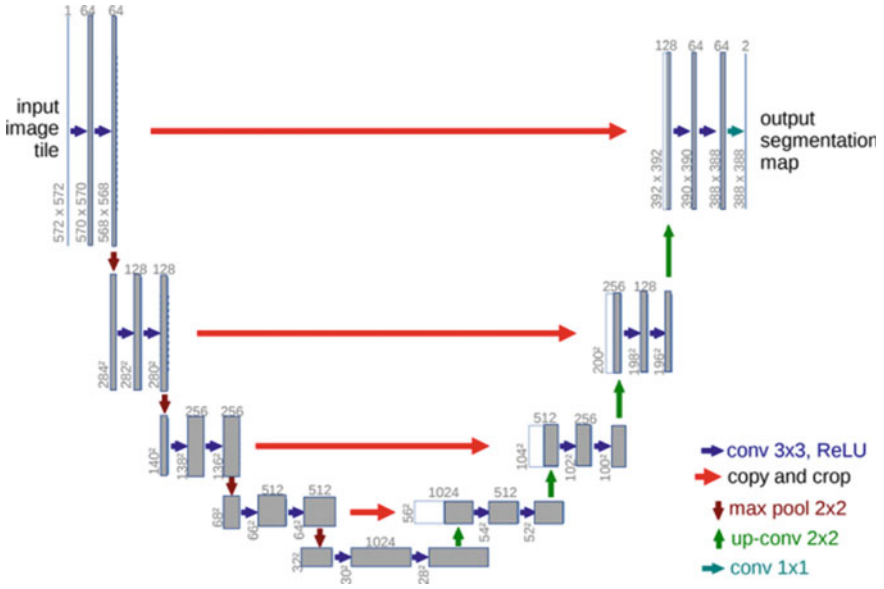


Fig. 4 U-Net architecture (*Source* Navab et al. [28], p. 235)

where Z stands for the output after the transposed convolution operation and K for kernel size.

Training. This U-Net model was applied on 616 images of 512 × 512 size split into 80% of the data which was allocated for training purposes, while the remaining 20% was reserved for testing. The training endures approximately 15 min for 100 epochs using GPU Colab and TensorFlow/Keras libraries.

4 Results and Discussion

In general, the training has achieved 98% in training and validation accuracy and approximately 86% for mean intersection over union (IoU) which are fairly good results, that could be improved after data augmentation to make it more representative for validation (shown in Figs. 5, 6, 7).

Centerlines' Extraction. In this step, this input is the segmented masks of coronary artery images in BGR color space to finally have binary images with centerline extracted [11]. Figure 8 shows the next process:

- Images are converted from BGR color space to grayscale (meant to be used on reading images using cv2.imread).
- Thresholding to ensure all the images are binary.

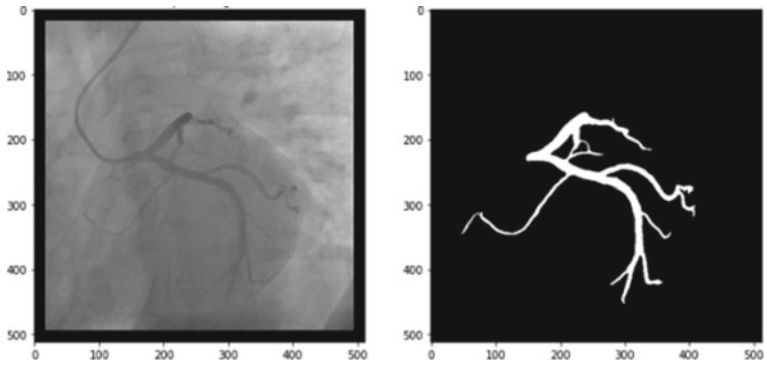


Fig. 5 U-Net segmentation results

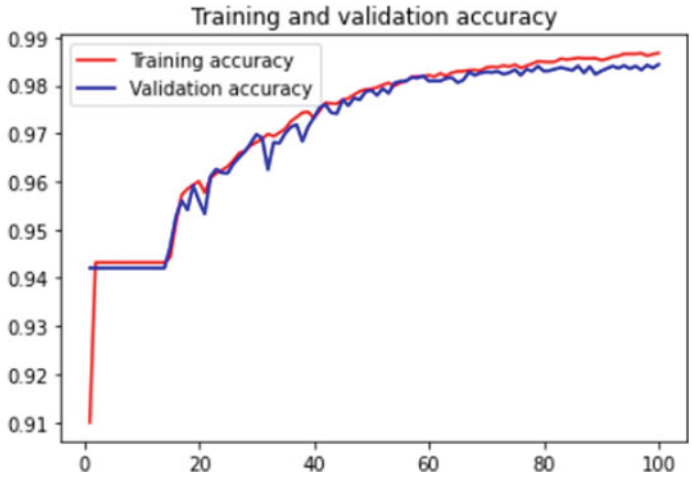


Fig. 6 U-Net learning curve/results of training accuracy

- Normalizing the image, so they only have values 0, 1.
- Closing morphological operation was used to remove any disconnected segments
- Skeletonizing to extract the centerline (it is the process of iteratively removing the border pixels until no more pixels can be removed).

Diameter Calculation. Figure 9 shows the process followed:

- Extracting centerlines.
- Distance transform is used with Euclidean distance; this returns an image in which each pixel represents its distance from the border of the contour.
- Bitwise and operation are performed on the resulting distance transform map with the centerline as a mask, this results in those pixels which are in the centerline containing the distance from the border, and all the remaining pixels are set to 0.

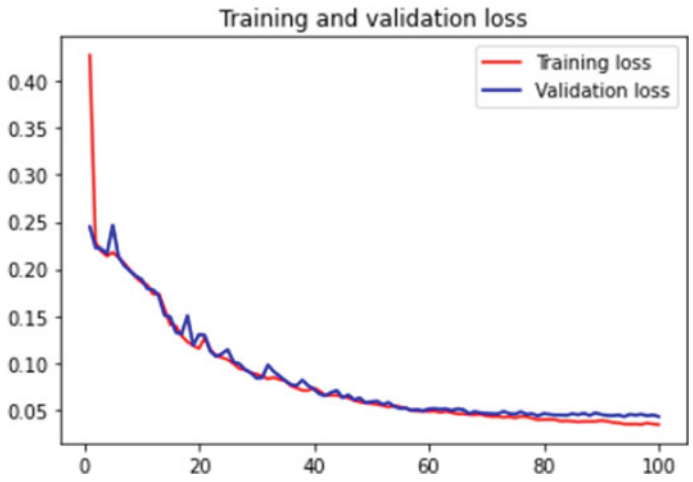


Fig. 7 U-Net learning curve/results of training loss

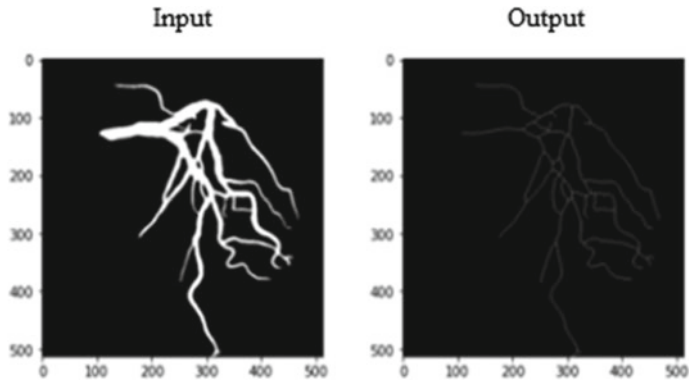


Fig. 8 Centerlines' extraction

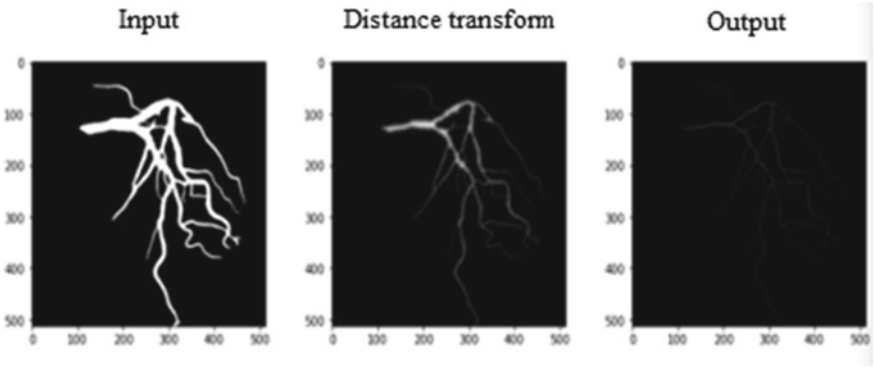


Fig. 9 Diameter calculation

5 Conclusion

In conclusion, having its architecture constantly improved over time, U-Net model is nowadays considered as one of the highly accurate biomedical image segmentation CNN-based models. The basic model, which was utilized throughout this project, reached 98% of accuracy as well as 86% of IoU in the segmentation of coronary arteries from ICA. The subsequent phase aims to further improve the architecture and enhancement of the dataset. This will produce more accurate outcomes that may be exploited in detecting stenosis after measuring arteries' centerlines and diameters.

Acknowledgements The authors would like to extend their heartfelt gratitude to Professor Abdelfettah Daoudy, ESL teacher in Amideast and researcher in the field of Applied Linguistics at Mohammed V University for proofreading this article.

References

1. Cardiovascular Diseases (2023) Available online at: https://www.who.int/health-topics/cardiovascular-diseases#tab=tab_1 (accessed Jun 08, 2023)
2. Hajikhani B et al (2023) COVID-19 and coronary artery disease; a systematic review and meta-analysis. New Microb New Infect 53:101151
3. Zhu S et al (2022) A knowledge graph-based analytical model for mining clinical value of drug stress echocardiography for diagnosis, risk stratification and prognostic evaluation of coronary artery disease. Int. J. Cardiol. 2:109181
4. Alizadehsani R et al (2019) Machine learning-based coronary artery disease diagnosis: a comprehensive review. Comput Biol Med 111(May):103346
5. Du T et al (2021) Training and validation of a deep learning architecture for the automatic analysis of coronary angiography. EuroIntervention 17(1):32–40
6. Rjiba S et al (2020) CenterlineNet: automatic coronary artery centerline extraction for computed tomographic angiographic images using convolutional neural network architectures. In: 2020 10th International Conference on Image Processing Theory, Tools and Applications, IPTA 2020
7. Gülsün MA, Funka-Lea G, Sharma P, Rapaka S, Zheng Y (2016) Coronary centerline extraction via optimal flow paths and CNN path pruning. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol 9902, pp 317–325
8. Chu M, Wu P, Li G, Yang W, Gutiérrez-Chico JL, Tu S (2023) Advances in diagnosis, therapy, and prognosis of coronary artery disease powered by deep learning algorithms. JACC 3(1):1–14
9. Li Y, Wu Y, He J, Jiang W, Wang J, Peng Y, Jia Y, Xiong T, Jia K (2022) Automatic coronary artery segmentation and diagnosis of stenosis by deep learning based on computed tomographic coronary angiography. Eur Radiol 32:6037–6045
10. Du H, Ling L, Yu W, Wu P, Yang Y, Chu M, Yang J, Yang W, Tu S (2022) Convolutional networks for the segmentation of intravascular ultrasound images: evaluation on a multicenter dataset. Comput Methods Programs Biomed 215:106599
11. Lin A et al (2022) Deep learning-enabled coronary CT angiography for plaque and stenosis quantification and cardiac risk prediction: an international multicentre study. Lancet Digital Health 4(4):e256–e265
12. Hong H et al (2022) (2022) Risk stratification in acute coronary syndrome by comprehensive morphofunctional assessment with optical coherence tomography. JACC 2(4):460–472

13. Zhao C et al (2021) Automatic extraction and stenosis evaluation of coronary arteries in invasive coronary angiograms. Comput Biol Med 136(March):104667
14. Tu BXM, Ding D, Chang Y, Li C, Wijns W (2021) Diagnostic accuracy of quantitative flow ratio for assessment of coronary stenosis significance from a single angiographic view: a novel method based on bifurcation fractal law. Catheter Cardiovasc Interv 97(S2):1040–1047
15. Yu W et al (2021) Accuracy of intravascular ultrasound-based fractional flow reserve in identifying hemodynamic significance of coronary stenosis. Circul Cardiovasc Interv 14(2):E009840
16. Chu M et al (2021) Artificial intelligence and optical coherence tomography for the automatic characterisation of human atherosclerotic plaques. EuroIntervention 17(1):41–50
17. Siddique N, Paheding S, Elkin CP, Devabhaktuni V (2021) U-net and its variants for medical image segmentation: a review of theory and applications. IEEE Access 9:82031–82057
18. Long J, Shelhamer E, Darrell T (2015) Fully convolutional networks for semantic segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 3431–3440 (2015)
19. Dorgham O, Naser MA, Ryalat MH, Hyari A, Al-Najdawi N, Mirjalili S (2022) U-NetCTS: U-Net deep neural network for fully automatic segmentation of 3D CT DICOM volume. Smart Health 26(August):100304
20. Huang W, Huang L, Lin Z, Huang S, Chi Y, Zhou J, Zhang J, Tan RS, Zhong L (2018) Coronary artery segmentation by deep learning neural networks on computed tomographic coronary angiographic images. In: Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBS), vol 2018-July(July 2018), pp 608–611
21. Moeskops P, Wolterink JM, Van der Velden BHM et al (2016) Deep learning for multi-task medical image segmentation in multiple modalities. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol 9901 LNCS, no October, pp 478–486
22. Han D, Liu J, Sun Z, Cui Y, He Y, Yang Z (2020) Deep learning analysis in coronary computed tomographic angiography imaging for the assessment of patients with coronary artery stenosis. Comput Methods Prog Biomed 196:105651
23. Baccouch W, Oueslati S, Solaiman B, Labidi S (2023) A comparative study of CNN and U-Net performance segmentation of medical images: application to cardiac MRI. Proc Comput Sci 219(2022):1089–1096
24. Alakwaa W, Nassef M, Badr A (2017) Lung cancer detection and classification with 3D convolutional neural network (3D-CNN). Int J Biol Biomed Eng 11(8):66–73
25. Ronneberger O, Fischer P, Brox T (2015) U-Net : convolutional networks for biomedical image segmentation. In: Medical Image Computing and Computer-Assisted Intervention, pp 234–241
26. Chen J, Lu Y, Yu Q, Luo X, Adeli X, Wang Y, Lu L, Yuille AL, Zhou Y (2021) TransUNet: transformers make strong encoders for medical image segmentation, pp 1–13
27. Zhou Z, Rahman Siddiquee MM, Tajbakhsh N, Liang J (2018) UNet++: a nested U-Net architecture for medical image segmentation BT—deep learning in medical image analysis and multimodal learning for clinical decision support. MICCAI 11045(2018):3–11
28. Navab N, Hornegger J, Wells WM, Frangi AF (2015) Medical image computing and computer-assisted intervention. In: MICCAI 2015: 18th International Conference Munich, Germany, October 5–9, 2015 proceedings, part III, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol 9351, pp 12–20

Change Detection for Multispectral Remote Sensing Images Using Deep Learning



M. Lasya, Radhesyam Vaddi, and S. K. Shabeer

Abstract Urbanization is a global process. Rapid industrialization and urbanization in developing nations represent a serious threat to the environment. The method of "change detection" examines how a region's features have changed during a span of two or more times. Deep learning methods have outperformed more conventional change detection methods in tests. Artificial neural networks are the foundation of the specialized machine learning method known as deep learning. Comparing the deep learning model to other machine learning techniques reveals significant capability and versatility. Two visuals taken at various timestamps are given as input to a Siamese neural network for change detection. For each of the input images, higher-level feature vectors are produced using a sequence of convolutional layers. This method aids in the extraction of significant features that may be utilized to evaluate and quantify the differences between the images that were viewed. The Euclidean norm is then used to compare the feature vectors to determine the degree of change. Deep Learning Change Detection (DLCD) outperforms traditional change detection due to three factors: better information representation, enhanced change detection techniques, and performance improvements. To encourage better decision-making, it is essential to identify agricultural changes, which we can do with DLCD.

Keywords Deep Learning Change Detection (DLCD) · Computer vision · Multispectral images · Remote sensing · Change detection · Siamese neural network

M. Lasya · R. Vaddi · S. K. Shabeer (✉)
Department of IT, VR Siddhartha Engineering College, Vijayawada, India
e-mail: shaikshabeer2403@gmail.com

R. Vaddi
e-mail: syam.radhe@vrsiddhartha.ac.in

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2023

S. Shakya et al. (eds.), *Fourth International Conference on Image Processing and Capsule Networks*, Lecture Notes in Networks and Systems 798,
https://doi.org/10.1007/978-981-99-7093-3_11

1 Introduction

It is essential to comprehend the connections and interactions between human and natural occurrences in order to make better decisions, which calls for rapid and accurate change detection of Earth's surface features. The "change detection" method assesses how the characteristics of a given area have changed over the course of two or more time periods. Change detection can be broadly categorized into two types: supervised change detection and unsupervised change detection [1]. By utilizing change detection techniques [2], it becomes possible to identify alterations in land use and land cover (LULC) [3]. A common approach involves comparing aerial or satellite images captured at different time points to identify and analyze the observed changes in a region.

In order to identify and maintain track of an area's physical characteristics, remote sensing entails monitoring the radiation that an area reflects and emits from a distance (typically from satellite or aircraft).

Remote sensing involves capturing photographs of the Earth from a distance using specialized cameras, which provide scientists with a way to "feel" the Earth and study its features. The cameras are mounted on satellites and aircraft, allowing for large areas of the Earth's surface to be captured in photographs. This provides researchers with a much broader view of the Earth than what can be observed from the ground. With the help of remotely sensed Earth imagery, it is possible to track changes in features such as the growth of cities, alterations in agriculture, and changes to woodlands over periods of years or decades.

1.1 Applications of Remote Sensing

Remote sensing is the practice of using sensors and other instruments to collect data about the Earth from a distance, often from space. The measurement of soil moisture content using active and passive sensors is one of the main applications of remote sensing. Other applications include mapping land use, weather forecasting, environmental analysis, natural hazards analysis, and resource discovery [4].

In addition to these applications, remote sensing can also be used for monitoring large forest fires from space, which allows rangers to observe a much larger area than they could on the ground [5]. This is just one example of the many ways in which remotely sensed photographs of the Earth can be used, including watching erupting volcanoes, tracking clouds to anticipate the weather, and detecting dust storms [6].

2 Proposed Work

2.1 Datasets

The initiative, which is a component of the Copernicus Programme started by the European Union, utilizes Sentinel-2 satellite data. Sentinel-2 is equipped with a multispectral instrument (MSI), which is a multispectral imaging sensor designed to capture high-resolution images of the Earth's surface. It is a global monitoring program with the objective of disseminating reliable and timely information about the climate, land, and oceans of the planet. The European Space Agency (ESA) created the mission, which frequently takes high-resolution optical shots of the Earth's land and coastlines at a distance of between 10 and 60 m. For the years 2012 and 2015, the data was obtained from Swisstopo (Figs. 1, 2). Swisstopo is a federal agency in Switzerland that is responsible for surveying and mapping the country. They provide a diverse range of geospatial datasets that cover the entire country. It is responsible for geodetic, topographic, and cartographic surveys, as well as the production and dissemination of geospatial data and maps. Swisstopo plays a crucial role in maintaining accurate and up-to-date geographical information for various applications, including land management, spatial planning, infrastructure development, and environmental monitoring. The organization collects and maintains data related to topography, elevation, land cover, cadastral information, and geodetic reference systems.

2.2 Architecture

Figure 3 shows the architecture diagram. ResNet34 serves as an encoder in a Siamese neural network. The images are changed to the necessary dimensions. The multispectral image is sent into the convolutional layers of the neural network to produce a feature vector. A feature vector holds all of the features of a picture. The Siamese neural network generates feature vectors, denoted as $f(\mathrm{X})$ and $f(\mathrm{Y})$, for images X and Y captured at different timestamps T1 and T2, respectively. These feature vectors are obtained by passing the respective images through identical convolutional layers that share the same parameters. The resulting feature vectors are encoded with 128 bits. To assess the dissimilarity between the two feature vectors, a difference vector is calculated by the Siamese neural network. This difference vector captures the variations and discrepancies between $f(X)$ and $f(Y)$, enabling the network to quantify the level of change between the two images.



Fig. 1 **a–d** Images of Switzerland in the year 2012

2.3 Proposed Work

Neural networks are nearly perfect at every task in the deep learning era of today, but they need more data to do so. However, we cannot always rely on obtaining additional data to tackle problems like face recognition and signature verification. To address these kinds of difficulties, we have a new form of neural network architecture called Siamese networks.

In order to obtain better predictions, it only uses a small number of photographs. Siamese networks have been more well-liked in recent years due to their capacity to learn from relatively little data. This post will explain Siamese networks and how to use them to build a signature verification system with Pytorch.

Siamese neural networks are a type of neural network architecture that has gained popularity due to their ability to compare feature vectors and determine the similarity between inputs. The name "Siamese" refers to the fact that these networks have two

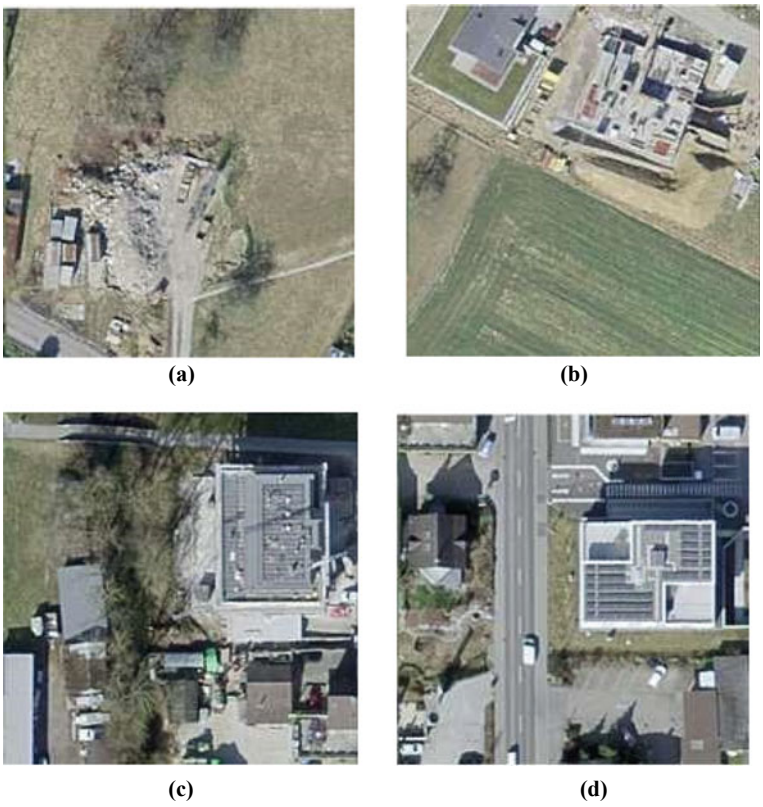


Fig. 2 **a–d** Images of Switzerland in the year 2015

or more identical subnetworks, which share the same configuration, including their parameters and weights. This means that when the network is trained, the updates to the parameters of one subnetwork are mirrored in the other subnetwork, ensuring that both subnetworks remain identical. The Siamese network takes in two images as input and produces two feature vectors, one for each image. These feature vectors are then compared to determine the degree of similarity between the two images.

Neural networks are capable of predicting different classes, but it might be difficult to add or remove new classes because the network must be retrained using the complete dataset. For training, deep neural networks need a lot of data. When taught with a similarity function, Siamese neural networks (SNNs) can assess whether two images are the same. As a result, SNNs can categorize fresh data without the need for network retraining.

To prepare the input data, we preprocess it. Before converting the data to NumPy arrays, it normalizes the data by taking the mean from each channel's value and dividing it by the standard deviation.

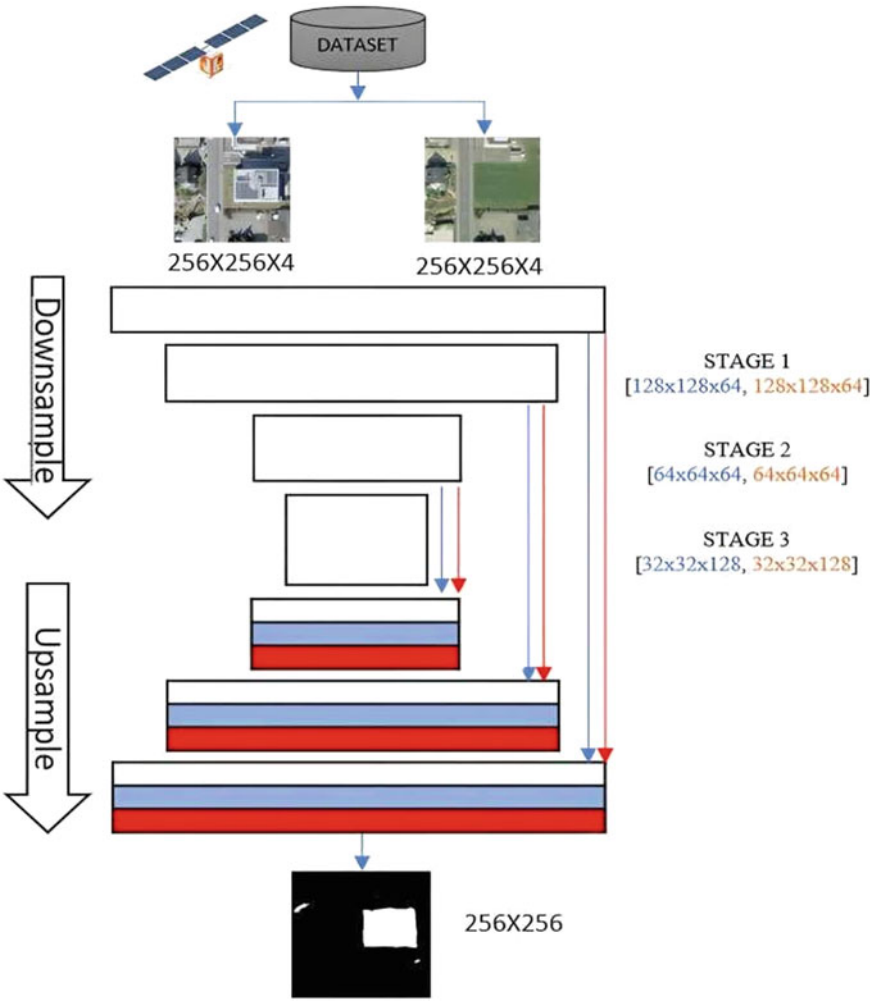


Fig. 3 Architecture diagram

Algorithm: Siamese Neural Network Input: Very high-resolution satellite images.

Output: Change map.

1. Start
2. Consider the two input images as $X = (cX, tX)$ and $Y = (cY, tY)$.
3. Compute the semantic relatedness of the two matrices as the Euclidean distance between their matrix representations:

$$D\phi(X, Y) = \|\{\phi(CX) \oplus \phi(tX)\} - \{\phi(CY) \oplus \phi(tY)\}\|2 \quad (1)$$

$\varphi(cX)$ and $\varphi(cY)$ are caption representations and $\varphi(tX)$ and $\varphi(tY)$ are matrix representations for the query and the candidate matrix, respectively.

4. Define the contrastive loss as a quadratic function of the pairwise distances, aiming for $D\varphi(X, Y)$ to be small (close to zero).
5. If table Y is similar to table X and equal to or larger than the margin m otherwise:

$$L(y, \phi, X, Y) = \frac{1}{2}(1 - y)D_{\phi}^{2}(X, Y) + \frac{1}{2}y\left\{\max(0, m - D_{\phi}(X, Y)\right\}^{2} \quad (2)$$

6. y is the true label of the pair.
7. Obtain binary classification labels by thresholding the distance at half of the margin, $m/2$. Pairs with distance scores below this threshold are considered similar, while others are considered dissimilar.
8. Output the difference label of the images $L(X, Y)$.
9. End

Recognizing and quantifying change over time can provide valuable insights in various domains. For example, assessing the progress of a city or town's infrastructure development can help gauge its economic success. However, any model designed for this purpose needs to identify and focus on relevant changes while disregarding unrelated ones. In the case of tracking structural changes, it becomes necessary to filter out variations such as alterations in water bodies, trees, and roadways that may not be of interest. This challenge is particularly evident in data normalization since photographs captured over time can exhibit inconsistencies that are difficult to consistently account for. When using a smartphone to capture facial images, factors like lighting conditions and positioning settings can vary between each capture, further complicating the normalization process. Similarly, satellite images are subject to variations caused by the azimuthal and elevational angles of the satellite, fluctuations in cloud cover, and the reflection of sunlight. These factors introduce additional complexities in analyzing satellite imagery consistently over time.

According to Ref. [7], the traditional method of measuring change by directly comparing pixel differences between images is no longer considered the simplest and most direct approach. While more sophisticated methods exist, they still yield good results. An alternative method involves using a Siamese neural network, which takes two images captured at different timestamps as input. The network applies a series of convolutional layers to each image to produce higher-level feature vectors [8]. The Euclidean norm is then used to evaluate these feature vectors and determine the degree of change. According to the theory, if there is not a clear difference between the two images, the corresponding feature vectors in the dimensional space will look comparable.

To train Siamese networks, one uses a contrastive loss function. If there is no change, it seeks to minimize the distance between the two feature vectors, and vice versa. The picture that follows tries to demonstrate it using a situation with two classes. The actual label would be 1 if neither of the two photos included the modifications we were training for [9–12].

3 Result Analysis

We downloaded satellite images from the Swiss Federal Office of Topography, specifically the Sentinel-2 images, which show views of Switzerland in multiple spectrums. Comparing images with different time stamps is often necessary to identify changes that have occurred in a particular area over time. This approach is useful for monitoring changes in LULC over several years. The process involves comparing images acquired at different times to identify any differences. The Deep Learning Change Detection (DLCD) algorithm generates a binary image that indicates the locations of LULC changes [7]. Outcomes are shown in Figs. 4, 5, 6 and 7.

The suggested method results in the confusion matrix. When assessing a classification model's performance, a confusion matrix is a table that displays the percentage of accurate and inaccurate predictions made. Its four subgroups are True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN). When both the anticipated and actual values are positive, TP occurs. Negative values of



Fig. 4 **a** The image of land in Timestamp 1, **b** the image of land in Timestamp 2, and **c** the change map which shows the change that occurred in agricultural lands due to urbanization

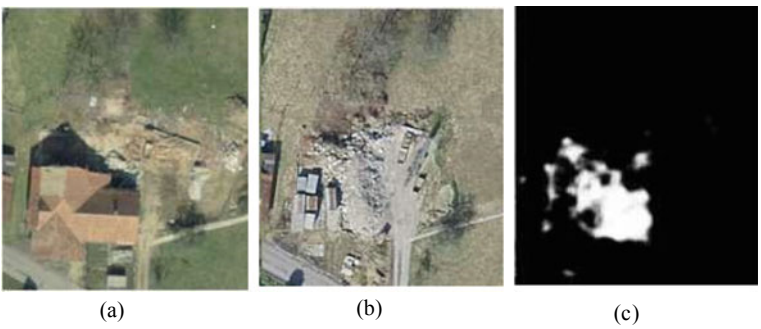


Fig. 5 **a** The image of land in Timestamp 1, **b** the image of land in Timestamp 2, **c** the change map which shows change occurred due to the demolition of the building



(a) (b) (c)

Fig. 6 **a** The image of land in Timestamp 1, **b** the image of land in Timestamp 2, and **c** the change map which shows the change occurred due to urbanization



(a) (b) (c)

Fig. 7 **a** The image of land in Timestamp 1, **b** the image of land in Timestamp 2, **c** the change map which shows the change that occurred due to the construction of a new building in that area

the expected and actual values result in TN. FP, often referred to as a Type 1 error, occurs when the forecast is positive but the actual result is negative. On the other hand, when the actual value is positive, but the forecast is negative, FN, also known as a Type 2 error, takes place.

Table 1 presents the confusion matrix. Metrics like accuracy, precision, and F1-Score are generated by the Siamese neural network to assess its effectiveness in change detection. The percentage of accurate forecasts among all made predictions is known as accuracy. While the F1-score combines precision and recall, taking into account both false positives and false negatives, precision concentrates on the accuracy of positive predictions. A high F1-score indicates accurate positive and negative class prediction, whereas a low score indicates bias toward one class. These metrics shed light on the model's effectiveness and its precision in identifying changes.

Table 2 presents the metrics obtained for the Siamese neural network.

Table 1 Confusion matrix actual values

	Change	No change
Change	90	1
No change	12	106

Table 2 Metrics

Metrics	Formula	Values
Accuracy	(TP + TN)/(Total)	0.938
Precision	(TP)/(TP + FP)	0.989
Recall	(TP)/(TP + FN)	0.882
F1-score	2(Precision*Recall)/(Precision + Recall)	0.932

- The final output of our project is a binary image, where the white area symbolizes the change and the black area symbolizes no change.
- It was observed that the model produces an accuracy of 93.8%.
- Precision is a measure of the proportion of positive class forecasts that actually belong to the positive class. It was discovered that the model produced data with a precision of 98.9%.
- Recall quantifies the number of accurate class predictions generated from the entire dataset's successful cases. Recall for the model was found to be 88.2%.
- A single score provided by F-measure strikes a balance between recall and precision issues.

4 Conclusion

In this project, we used two separate timestamps to look for changes in Switzerland as a particular location. Our investigation utilized multispectral remote sensing photographs taken by the Sentinel-2 satellite to detect changes in the target area. To achieve change detection, we employed a Siamese neural network. Throughout our tests, we discovered that a number of acquisition functions and prediction uncertainty calculation strategies worked effectively. These methods achieved performance comparable to a model trained on the whole dataset with only a few hundred samples, outperforming a naive random baseline by a large margin.

Our project's ultimate objective is to use remote sensing images to track changes in agricultural areas. As a result, we focused on detecting changes in agricultural areas, which may include changes in land use, crop yield, and soil conditions. By identifying these changes, we hope to provide insights that can help improve agricultural practices and increase productivity while minimizing environmental impact. Calculating the percentage change before and after natural disasters, and other features will be a part of our future research. To enhance the performance of a Siamese neural network model for change detection, we can employ techniques such as data augmentation

to increase the diversity of training samples, explore different network architectures to capture complex patterns, consider alternative similarity metrics, and consider ensemble methods for combining multiple models. These strategies can collectively improve the model's ability to detect changes accurately and robustly.

References

1. De Jong KL, Bosman AS (2018) Unsupervised change detection in satellite images using convolutional neural networks. Preprint at: arXiv:1812.05815. https://arxiv.org/abs/1812.05815
2. Singh A (1989) Review article digital change detection techniques using remotely-sensed data. Int J Remote Sens 10(6):989–1003
3. Su L, Gong M, Zhang P, Zhang M, Liu J, Yang H (2017) Deep learning and mapping based ternary change detection for information unbalanced images. Pattern Recognit 66:213–228
4. Deng L, Hinton L, Kingsbury B (2013) New types of deep neural network learning for speech recognition and related applications: an overview. In: Proceedings of IEEE International Conference Acoustics, Speech Signal Process, pp 8599–8603
5. Gillanders SN, Coops NC, Wulder MA, Goodwin NR (2008) Application of land-sat satellite imagery to monitor land-cover changes at the Athabasca Oil Sands, Alberta, Canada. Can Geogr 52:466–485. https://doi.org/10.1111/j.1541-0064.2008.00225.x
6. Ma L, Liu Y, Zhang X, Ye Y, Yin G, Johnson BA (2019) Deep learning in remote sensing applications: a meta-analysis and review. ISPRS J Photogramm Remote Sens 152:166–177
7. Daudt RC, Le Saux B, Boulch A, Gousseau Y (2018) Urban change detection for multispectral earth observation using convolutional neural networks. In: International Geoscience and Remote Sensing Symposium (IGARSS). IEEE
8. Hou B, Wang Y, Liu Q (2017) Change detection based on deep features and low rank. IEEE Geosci Remote Sens Lett 14(12):2418–2422
9. Zhang L, Zhang L, Du B (2016) Deep learning for remote sensing data: a technical tutorial on the state of the art. IEEE Geosci Remote Sens Mag 4(2):22–40
10. Cao C, Dragićević S, Li S (2019) Land-use change detection with convolutional neural network methods. Environments 6(2):25
11. Hamdi ZM, Brandmeier M, Straub C (2019) Forest damage assessment using deep learning on high-resolution remote sensing data. Remote Sens 11(17):1976
12. Coppin P, Jonckheere I, Nackaerts K, Muys B, Lambin E (2004) Digital change detection methods in ecosystem monitoring: a review. Int J Remote Sens 25(9):1565–1596

Explainable AI for Black Sigatoka Detection

Yiga Gilbert, Emmy William Kayanja, Joshua Edward Kalungi, Jonah Mubuuke Kyagaba, and Ggaliwango Marvin

Abstract Banana plants are susceptible to the dangerous fungal disease known as Black Sigatoka, which has a negative impact on global economies. Early detection and timely intervention are crucial for preventing the spread of the disease. In recent years, machine learning (ML) has shown great potential for detecting and diagnosing plant diseases, including Black Sigatoka. However, the lack of transparency and interpretability of ML models raises concerns about their use. In this paper, we propose explainable AI approaches for Black Sigatoka detection using Local Interpretable Model-Agnostic Explanations (LIME) and Integrated Gradients. Our methodology involves the utilization of Mobilenet V2 and AlexNet models which are trained on an extensive dataset of banana leaf images and generating explanations to provide a better understanding of the CNN's decision-making process. We demonstrate the effectiveness of our approach through extensive experiments and show that it outperforms existing state-of-the-art methods for Black Sigatoka detection. Our approach not only provides accurate and interpretable results but also promotes responsible AI practices for plant disease diagnosis.

Keywords Black sigatoka · Explainable AI (XAI) · Local interpretable model-agnostic explanations (LIME) · Convolutional neural network (CNN) · Plant disease diagnosis

Y. Gilbert · E. W. Kayanja · J. E. Kalungi · J. M. Kyagaba · G. Marvin (✉)
Department of Computer Science, Makerere University, Kampala, Uganda
e-mail: ggaliwango.marvin@mak.ac.ug

Y. Gilbert
e-mail: yiga.gilbert@students.mak.ac.ug

E. W. Kayanja
e-mail: emmy.kayanja@students.mak.ac.ug

J. E. Kalungi
e-mail: joshua.kalungi@students.mak.ac.ug

J. M. Kyagaba
e-mail: jonah.kyagaba@students.mak.ac.ug

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2023

S. Shakya et al. (eds.), *Fourth International Conference on Image Processing and Capsule Networks*, Lecture Notes in Networks and Systems 798,
https://doi.org/10.1007/978-981-99-7093-3_12

1 Introduction

1.1 Backgroung and Motivation

Millions of people in Eastern Africa rely heavily on bananas in their diets, making up as much as one-fifth of the region's total calorie consumption per person [4]. Bananas are grown by more than 75% of the smallholders in East Africa [13] due to their special benefits, which include yields that are adequate despite unpredictable rainfall and their longstanding nature combined with their perennial fruiting nature [16]. In terms of consumption per person and total production, as well as the area occupied in Uganda, bananas come in first [15]. These qualities set bananas apart as a desirable crop for domestic food production, nourishment, and steady revenue [1]. Despite being a staple food, the productivity of bananas in Uganda is rapidly declining. This is partially brought on by ailments like Sigatoka leaf disease, which is brought on by fungi belonging to the genus Pseudocercospora [5]. With Black sigatoka, a fruit bunch may develop unevenly and prematurely as a result of severely infected leaves dying, limiting fruit production drastically [2]. In the Ugandan districts of Luwero with a mean Disease Severity Index of 40.4% and Mbarara with a mean DSI = 37.9%, Black Sigatoka disease was equally severe [6]. One potential solution to mitigate the spread of Black Sigatoka is through the use of Artificial Intelligence (AI) systems that can detect the disease early on and enable effective management and control measures [7, 12].

In constrained resource settings, such as smallholder farms, rural communities, and all developing countries at large, there may be limitations on the availability of resources such as computing power, Internet connectivity, and technical expertise [14]. Therefore, any AI system designed for Black Sigatoka detection in these settings must be efficient, cost-effective, and easy to use.

To address these challenges, Explainable AI (XAI) can be utilized to overcome these difficulties by enhancing the AI system's accuracy and dependability while ensuring transparency and interpretability. XAI provides a framework for understanding the reasoning behind the AI system's decision-making process, allowing users to evaluate the system's performance and identify any potential biases or errors.

1.2 Research Contribution

- Creating visual and analytical tools for exploring banana Sigatoka datasets.
- Develop and test models for detecting Black Sigatoka in bananas.
- Compare the performance of different Black Sigatoka detection models and suggest the best one.
- Generate and contrast interpretable explanations for Black Sigatoka classification to enhance model transparency.

2 Research Problem Definition

Banana Black Sigatoka has caused a significant deterioration in the quality and quantity of bananas produced in Uganda, a drop in farmer income, the death of banana trees, and an increase in production costs throughout the control process of BBS. Smallholders unknowingly spread banana Black Sigatoka through plant materials such as suckers locally known as "Ensukusa" from infected gardens and use ineffective techniques to identify the disease; by the time they do, the disease has advanced and begun to manifest symptoms, at which point it is too late.

Farmers require early warning of the disease outbreak. However, all deep learning models used to identify banana Black Sigatoka (BBS) have a black-box nature and are comparably weak in explaining their inference processes and outcomes, making them uninterpretable, non-intuitive, less human-understandable, and providing no explanations for AI predictions.

We opt to develop explainable deep learning models that are credible, comprehensible, and self-explanatory as well as algorithm strategies that will aid in banana Black Sigatoka (BBS) detection. This will enable farmers with limited resources and those without access to extension services to mitigate the economic, social, and environmental impacts of the BBS.

3 Research Approach and Methodology

Figure 1 is the methodology visualization, and the aim was to develop an explainable AI model to detect Black Sigatoka in banana leaves from the image data, and provide explanations for its predictions. The model aimed to help smallholders with limited resources to identify infected plants early. The steps were:

Study existing models for plant disease detection and review their strengths and weaknesses. The main weakness was the lack of explainability in the models.

Develop two models using AlexNet and MobileNet architectures, which are successful in image classification tasks. AlexNet used deep convolutional layers to learn complex features from the images. MobileNet used depth-wise separable convolutions to achieve a trade-off between accuracy and efficiency.

Train both models on a dataset of banana leaf images with different stages of Black Sigatoka infection. Fine-tune and optimize the models for better performance.

Apply Lime and Integrated Gradients techniques to both models to generate explanations for the predictions. These techniques highlighted the image regions that influenced the models' decisions, helping smallholders to understand the model's logic.

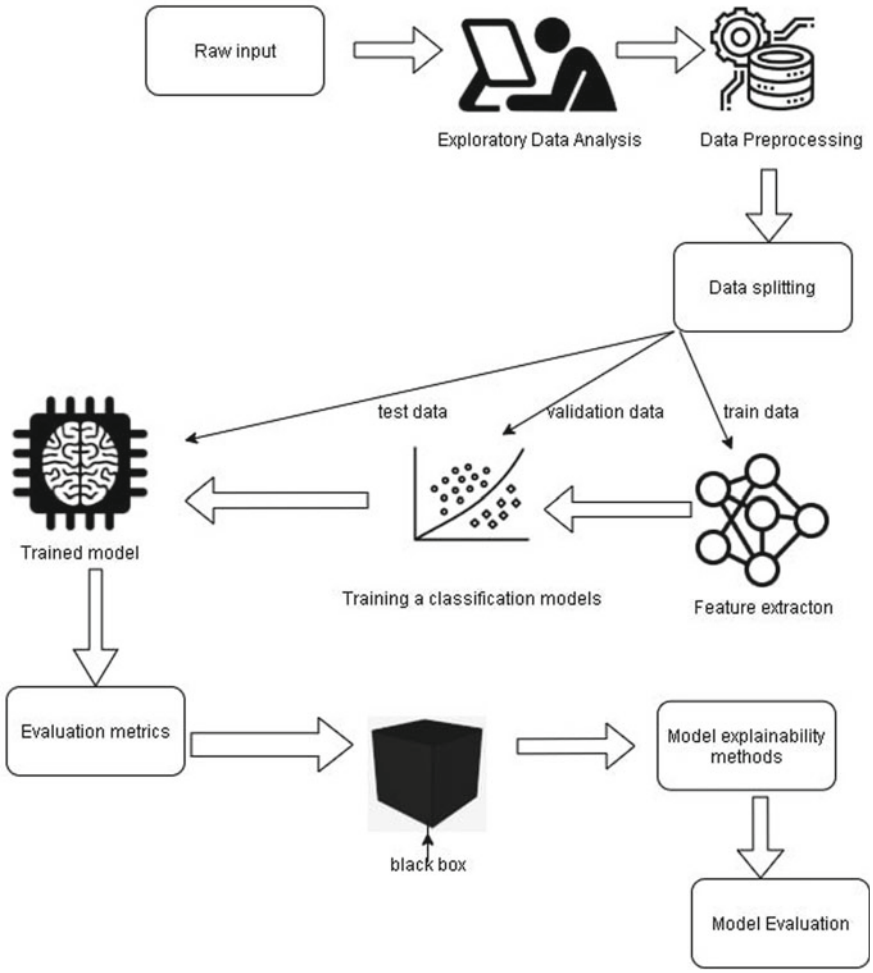


Fig. 1 Methodology visualization

3.1 Data Collection and Preprocessing

We used the Harvard Dataverse banana leaf disease dataset [10], which has 5883 images of healthy leaves and 6147 images of leaves infected with Black Sigatoka. To gain more insights into the disease process and potentially improve the accuracy of our model, we included images of different stages of disease progression in our dataset. We then applied multi-class classification instead of binary classification, dividing the images into four classes: "healthy", "early stage affected", "intermediate stage affected", and "late stage affected" as shown in Fig. 2.



Fig. 2 Image classes

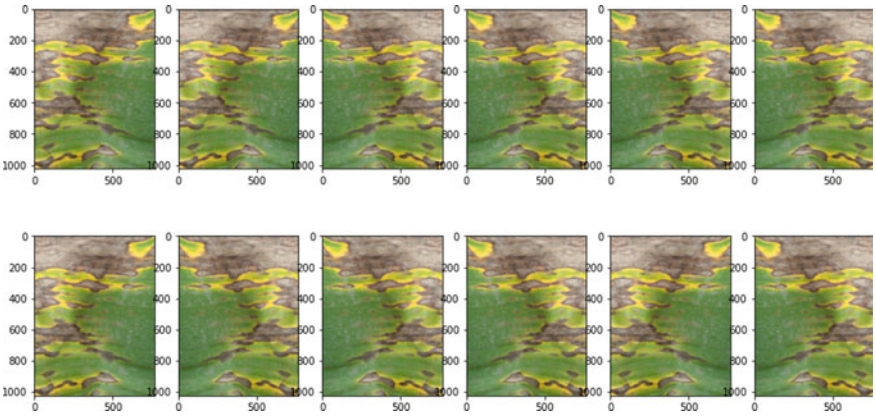


Fig. 3 Augmented banana leaves

We labeled our dataset with the four classes and applied various techniques to optimize our model's performance, such as data augmentation, transfer learning, and hyperparameter tuning. The evaluation criteria we employed for multi-class classification were precision, recall, and $F1$-score for each class. We also visualized the performance of our model in each class using confusion matrices.

As demonstrated in Fig. 3, we employed data augmentation to artificially increase the amount and diversity of training data from the original dataset. We applied slight modifications to each existing data point and generated 16 new data points in the latent space of the original data using machine learning models. We used techniques such as image resizing, rotation, flipping, noise addition, cropping, scaling, and translation. We used the ImageDataGenerator class from the Keras deep learning package to apply data augmentation automatically during model training.

3.2 Model Implementation

AlexNet Architecture and MobileNet

AlexNet is a deep convolutional neural network (CNN) model that consists of eight layers: five convolutional layers and three fully connected layers [3, 11]. The model uses various techniques to enhance its feature extraction and generalization abilities, such as local response normalization, dropout, rectified linear units (ReLU), and max-pooling. The model has over 60 million parameters and is deeper and wider than previous models. The input shape for AlexNet is $227 \times 227 \times 3$. The ReLU activation function is used for all convolutional layers. Dropout with a rate of 0.5 is applied after the fully connected layers to reduce overfitting. Batch normalization is used after each convolutional layer to improve training stability. Max pooling is used to downsample the feature maps after some convolutional layers. The feature maps are flattened before passing through the fully connected layers. The softmax activation function is used in the final layer to output the class probabilities.

The AlexNet Model Architecture is illustrated in Fig. 4.

MobileNet is a CNN model that aims to overcome the difficulties of deploying deep learning models on limited-resource devices like mobile phones [17]. It emphasizes efficiency and compactness without compromising on performance. MobileNet employs depth-wise separable convolutions, a factorized form of standard convolutions, to significantly reduce computational complexity and model size. This approach enables MobileNet to achieve a good balance between accuracy and inference speed. By utilizing depth-wise convolutions followed by point-wise convolutions, the model captures spatial, and channel-wise correlations efficiently. MobileNet's architecture as illustrated in Fig. 5 offers various configurations, allowing trade-offs between model size and accuracy. The input shape for MobileNet is

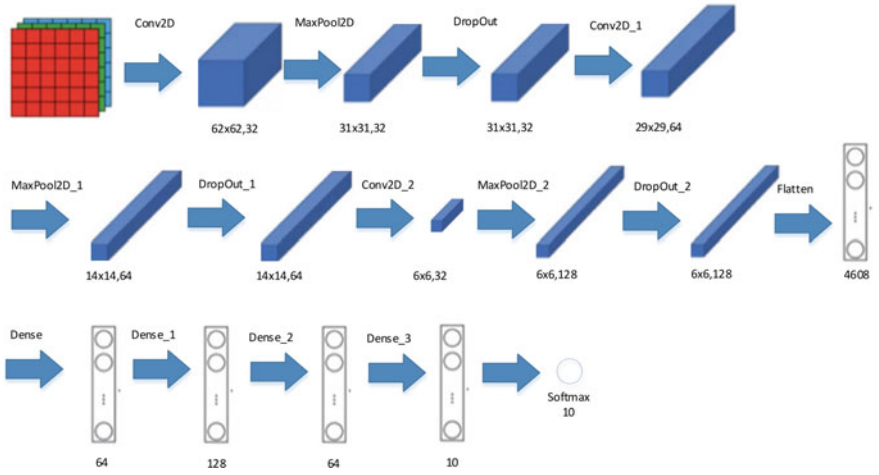


Fig. 4 AlexNet model architecture

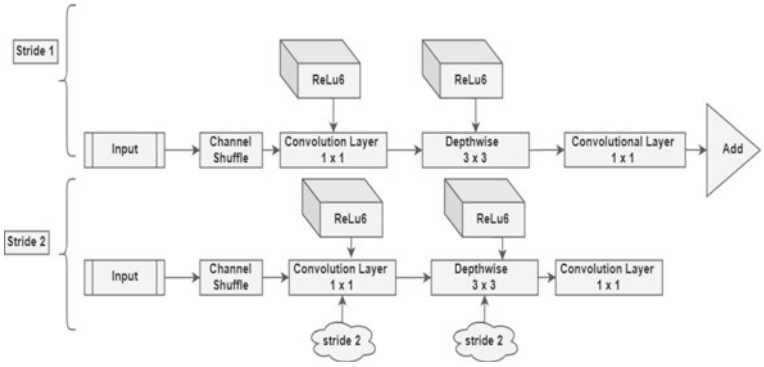


Fig. 5 MobileNet V2 model architecture

$224 \times 224 \times 3$. The activation functions used in MobileNet are ReLU and softmax. Dropout is used in MobileNet to prevent overfitting. The dropout rate is 0.5.

Table 1 shows the AlexNet Architecture that we employed.

Table 2 shows the MobileNet Architecture that we employed.

Model Training

The models underwent training using the Adam optimizer and utilized categorical cross-entropy loss. During the training process, the dataset was iterated over a predetermined number of epochs. Within each epoch, the model received training on mini-batches of images, and the weights were adjusted based on the gradients computed through backpropagation. The training progress was tracked using metrics like loss and accuracy, which offered valuable information about the model's performance throughout the training phase. Once the model training is finished, a separate validation set is employed to assess the performance of the trained models.

4 Major Research Findings

4.1 Model Evaluation

The evaluation process involved a range of performance metrics such as loss, accuracy, precision, recall, and $F1$-score as illustrated in Tabels 3 and 4. To aid in identifying potential misclassifications, a confusion matrix is generated in Fig. 6, visually representing the classification results. These evaluations provide insights into the model's effectiveness in accurately predicting and classifying the validation data.

Figure 7 shows the training and validation accuracy together with the training and validation loss of MobileNet and AlexNet models trained on a dataset of banana leaf images.

Table 1 AlexNet architecture

Layer (type)	Output shape	Parameters #
conv2d (Conv2D)	(None, 55, 55, 96)	34,944
batch_normalization (BatchNormalization)	(None, 55, 55, 96)	384
max_pooling2d (MaxPooling2D)	(None, 27, 27, 96)	0
conv2d_1 (Conv2D)	(None, 27, 27, 256)	614,656
batch_normalization_1 (BatchNormalization)	(None, 27, 27, 256)	1024
max_pooling2d_1 (MaxPooling2D)	(None, 13, 13, 256)	0
conv2d_2 (Conv2D)	(None, 13, 13, 384)	885,120
batch_normalization_2 (BatchNormalization)	(None, 13, 13, 384)	1536
conv2d_3 (Conv2D)	(None, 13, 13, 384)	1,327,488
batch_normalization_3 (BatchNormalization)	(None, 13, 13, 384)	1536
conv2d_4 (Conv2D)	(None, 13, 13, 256)	884,992
batch_normalization_4 (BatchNormalization)	(None, 13, 13, 256)	1024
max_pooling2d_2 (MaxPooling2D)	(None, 6, 6, 256)	0
flatten (Flatten)	(None, 9216)	0
dense (Dense)	(None, 4096)	37,752,832
dropout (Dropout)	(None, 4096)	0
dense_1 (Dense)	(None, 4096)	16,781,312
dropout_1 (Dropout)	(None, 4096)	0
dense_2 (Dense)	(None, 4)	16,388
Total parameters: 58,303,236		
Trainable parameters: 58,300,484		
Non-trainable parameters: 2752		

Considering the overall performance, both models demonstrated high accuracy and achieved competitive results. However, based on the provided evaluation, AlexNet appeared to have a slight advantage in terms of precision, recall, and $F1$-scores across multiple classes. Therefore, the AlexNet model was suggested as the best model for detecting Black Sigatoka in banana leaves.

Table 2 Mobile nets architecture

Layer (type)	Output shape	Parameters #
mobilenetv2_1.00_224 (Functional)	(None, 7, 7, 1280)	2,257,984
average_pooling2d_1 (AveragePooling2D)	(None, 3, 3, 1280)	0
flatten_1 (Flatten)	(None, 11,520)	0
dense_3 (Dense)	(None, 512)	5,898,752
dropout_2 (Dropout)	(None, 512)	0
dense_4 (Dense)	(None, 50)	25,650
dropout_3 (Dropout)	(None, 50)	0
dense_5 (Dense)	(None, 4)	204
Total parameters: 8,182,590		
Trainable parameters: 6,337,406		
Non-trainable parameters: 1,845,184		

Table 3 Comparison of AlexNet and MobileNet results

Metrics	Results	
	AlexNet	MobileNet
Train loss	0.0035	0.0663
Train accuracy	0.9995	0.9892
Row 3, cell 1	0.2164	0.5419
Test accuracy	0.9462	0.9167

4.2 XAI Results

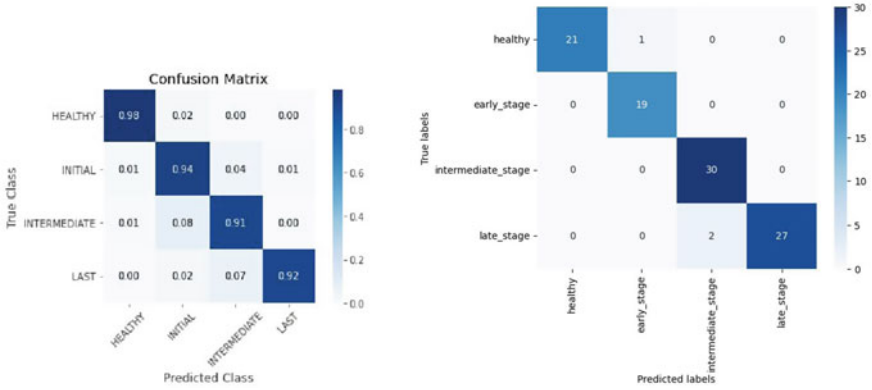
To gain insights into the decision-making processes of AlexNet and MobileNet, we employed LIME and Integrated Gradients techniques [8]. LIME provided local explanations by identifying the salient regions of an image that influenced the model's predictions. By visualizing these explanations, we gained a better understanding of the features considered by each model during classification. Integrated Gradients assigned importance values to individual pixels, allowing us to interpret the contributions of different image regions to the final prediction. These interpretability techniques enhanced our understanding of the inner workings of both models, providing valuable insights into their decision processes.

LIME

We used LIME, or Local Interpretable Model-agnostic Explanations, to comprehend how our black-box classifier model behaves [9]. Permutation of the data input LIME started by making a number of fictitious data closely related to the data point categorized in one class rather than the other. LIME produced a large number of samples

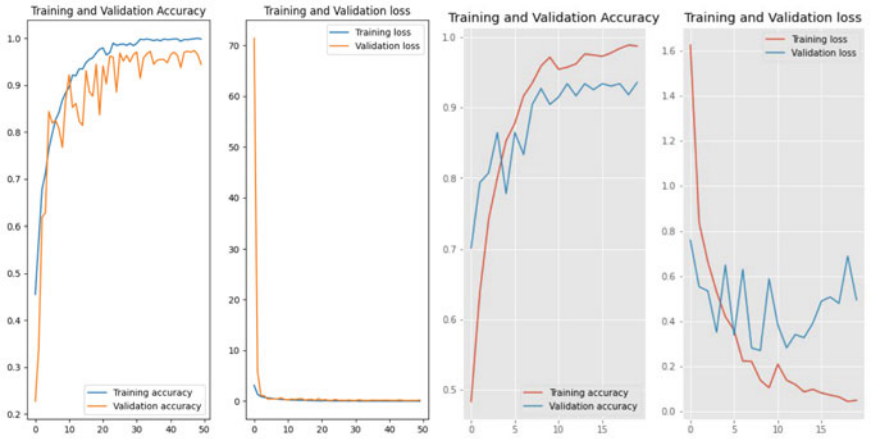
Table 4 Classification results

Class	AlexNet				MobileNet			
	Precision	Recall	$F1$-score	Support	Precision	Recall	$F1$-score	Support
HEALTHY	1.00	1.00	1.00	18	0.97	0.96	0.97	141
INITIAL_STAGE	1.00	1.00	1.00	26	0.92	0.89	0.90	147
INTERMEDIATE_STAGE	0.93	1.00	0.97	28	0.87	0.87	0.87	168
LAST_STAGE	1.00	0.93	0.96	28	0.91	0.96	0.93	134
Accuracy			0.98	100			0.92	590
Macro avg	0.98	0.98	0.98	100	0.92	0.92	0.92	590
Weighted avg	0.98	0.98	0.98	100	0.92	0.92	0.92	590



(a) MobileNet (b) AlexNet

Fig. 6 Confusion matrix comparison

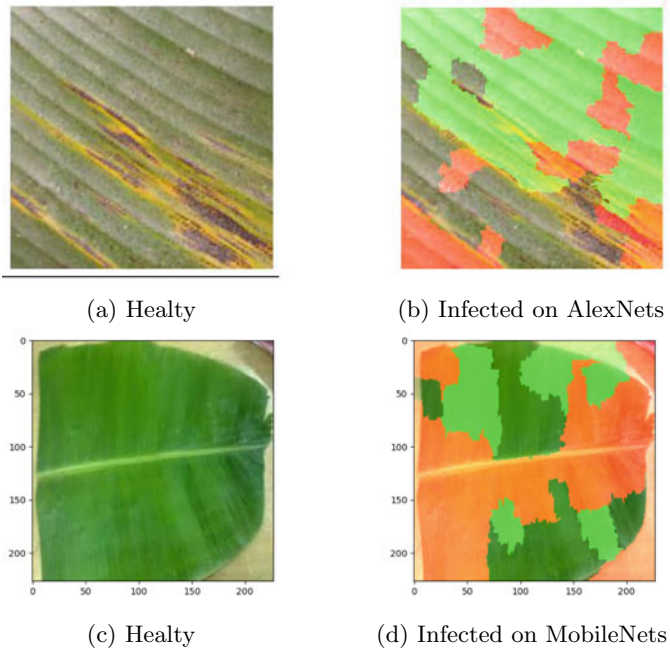


(a) AlexNet (b) MobileNet

Fig. 7 Graphical representation of training and validation metrics for MobileNet and AlexNet models on banana leaf images

that were identical to our input image by turning various super-pixels in our input image on and off.

Each fabricated data point's class was predicted. Following that, LIME forecasted whether each artificial data point was generated using our trained model, which represented the healthy class or the diseased class. Calculated the weights of all artificial data points. The weight of each fabricated data was then determined to determine its significance. The cosine distance metric was commonly utilized in the operation to calculate the distance between created data points and the original input data. This distance was then converted into a number between 0 and 1 using a kernel



(a) Healty (b) Infected on AlexNets

(c) Healty (d) Infected on MobileNets

Fig. 8 Lime visualizations

function. The mapped value was closer to one the closer the distance, increasing the weight as a result. The weight of a created data point increased its significance. After that, the cosine distance between each edited image and the original image was computed. The more closely a disturbed image matched the original image, the more weight and significance it had.

To describe the most relevant features, a linear classifier was fitted. The final stage involved creating a linear regression model using the weighted synthetic data points. Following this stage, the fitted coefficient for each feature was determined, much as in a standard linear regression analysis. If we sorted the coefficients immediately, the qualities with the highest coefficients had a substantial impact on how accurately our black-box machine learning model anticipated the future.

LIME's efficiency depends on generating many perturbed instances and training a simpler model on them. This lets LIME approximate the complex model's behavior near the chosen instance, without accessing the complex model's internals. Also, LIME uses interpretable features and an interpretable model to provide understandable explanations for model predictions.

In Fig. 8, the red regions represent features that contradicted the model output, the green regions are features that supported the model output, and the unhighlighted regions are features that had little or no impact on the model output.

Integrated Gradients
The importance of input features entering the model to the related prediction was also visualized using Integrated Gradient (IG). The Integrated Gradient is based on two assumptions that must be true:

Sensitivity: As a starting point for the Sensitivity computation, we created a Baseline image. The Integrated Gradients are then determined by interpolating from a baseline image to the real image using a sequence of photographs.

Implementation Invariance: This is satisfied when two functionally comparable networks assign the same attributions to the same input image and baseline image. Despite having wildly disparate implementations, dual networks are functionally comparable when their outputs are the same for all inputs.

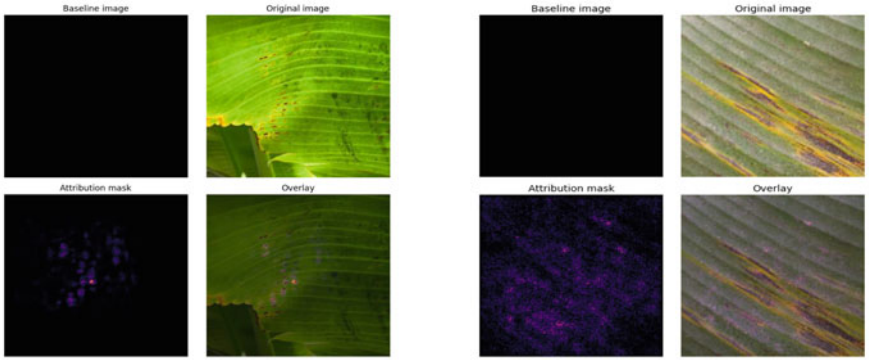
Calculating and visualizing integrated gradients (IG)

Step 1: We started from the baseline, which could be either a random image or a dark image with all of its pixels set to zero.
Step 2: Between the baseline and the original image, we created a linear interpolation. The feature space was incrementally increased with each interpolated image's intensity in interpolated images, which were modest steps since they lay amidst the baseline and the supplied image.
Step 3: To determine the correlation between changes to a feature and modifications to the model's predictions, we computed gradients. This gradient revealed which pixel had the greatest impact on the class probabilities predicted by the model. Various variable adjustments, the output, and the variable received some attribution to assist in determining the feature importance for the input image. No credit was given for a variable that had no impact on the result.
Step 4: Then we calculated the approximate numerical value by averaging gradients.
Step 5: Multiple interpolated images were added together, and the attribute quantities were in the identical units. Scale IG to the input image using the supplied image's pixel importance to represent the IG.

In Fig. 9, the purple regions indicate the features that increased the probability of the predicted class, while the black regions are features that decreased the probability of the predicted class. The white regions are features that had no impact on the model output.

5 Practical Implications

Transparent Decision-Making: We use explainable AI techniques, such as Lime and Integrated Gradients, to show how the AI models make predictions. This helps farmers and stakeholders understand the model's logic, trust the system, make informed decisions, and manage Black Sigatoka effectively.



(a) AlexNet (b) MobileNet

Fig. 9 Visualization comparison

Early Detection and Timely Intervention: We provide a practical tool for early detection of Black Sigatoka in banana leaves using explainable AI models. By detecting the disease early, farmers can intervene quickly to manage and control the disease. This can help reduce crop losses, increase yields, and protect farmers' income.

Enhanced Disease Management: We offer accurate and interpretable AI models for Black Sigatoka detection to improve disease management practices. Farmers can identify infected plants early, control the disease with targeted measures, and use resources such as fungicides optimally. This leads to more effective disease management, reduced crop losses, and improved agricultural productivity.

Potential for Adaptation to Other Diseases: We can adapt our methodology and approaches to other plant diseases as well. We can apply the knowledge and techniques from detecting Black Sigatoka in banana leaves to diagnose and manage other plant diseases, expanding the practical implications of our research beyond a specific disease.

6 Research Limitations

The model may face challenges in accurately distinguishing between banana leaves and other leaves that bear a strong resemblance to banana leaves like heliconia, Traveler's Palm, etc. This can result in misclassifications or difficulties in isolating leaves that share similar visual characteristics with banana leaves.

Trade-off between model performance and interpretability, where increasing interpretability may result in lower performance, or conversely, achieving better performance may come at the cost of interpretability which can lead to confusion or mistrust in the model.

Limited computational resources while trying to archive most accurate results.

7 Originality/Value

Explainable AI Models for Plant Disease Detection: We develop AI models that can detect Black Sigatoka in banana leaves and explain their decisions. Existing models are accurate but not interpretable, limiting their usefulness and transparency. We use techniques like Lime and Integrated Gradients to make our models more transparent and explainable, giving users insights into how they work.

Contribution to Responsible AI Practices: The incorporation of explainability in AI models for plant disease detection promotes responsible AI practices. By providing interpretable explanations for the models' predictions, users can evaluate the system's performance, identify potential biases or errors, and gain trust in the technology. This aspect of your research contributes to ethical considerations and encourages the adoption of AI in a responsible and accountable manner.

Comparative Evaluation and Performance Analysis: Our research involves a comprehensive evaluation of the proposed models, comparing their performance with existing state-of-the-art methods for Black Sigatoka detection. This analysis not only showcases the effectiveness of our approach but also adds value by providing a benchmark for future research and facilitating the advancement of the field.

8 Conclusion and Future Research Work

8.1 Conclusion

To achieve high accuracy in the model for detecting Black Sigatoka in banana leaves, we considered several factors. First, we ensured a high-quality and diverse dataset that accurately represents the problem and covers different stages of the disease. We preprocessed and augmented the data to enhance its quality and increase its diversity. We evaluated two models using our diverse dataset for image classification tasks: AlexNet and MobileNet. AlexNet demonstrated superior performance compared to MobileNet across multiple evaluation metrics. It achieved higher accuracy, precision, and recall rates, as evidenced by the confusion matrix. The misclassification rate of AlexNet was lower, indicating its ability to handle complex image patterns more effectively. The above results highlight the strengths of AlexNet in achieving high classification accuracy for a wide range of images. Integrated Gradients through visual inspection also were the most preferred method compared to LIME.

8.2 Future Works

While the initial focus may be on Uganda, we think about extending the application and study to other areas impacted by Black Sigatoka in order to tailor the solution to

various geographic locations and disease variants and work with regional organizations, scientists, and farmers.

We will extend our research efforts to develop a diverse range of explainable deep learning models specifically tailored for various banana plant diseases, aiming to enhance both accuracy and interpretability in agricultural disease detection systems.

Develop a mobile application to support multiple languages commonly spoken in the target regions for farmers in different regions that will be deployed on devices rather than the cloud.

Acknowledgements We want to express our heartfelt gratitude to the RISE Seed Fund at Makerere University College of Computing and Information Sciences for the full facilitation of this work.

References

1. Almekinders CJM et al (2019) Why interventions in the seed systems of roots, tubers and bananas crops do not reach their full potential. Food Secur 11:23–42
2. Brat P et al (2020) Review of banana green life throughout the food chain: from auto-catalytic induction to the optimisation of shipping and storage conditions. Sci Hortic 262:109054
3. Dong Y-n, Liang G-s (2019) Research and discussion on image recognition and classification algorithm based on deep learning. In: 2019 international conference on machine learning, big data and business intelligence (MLBDBI). IEEE, pp 274–278
4. Haggblade S, Dewina R (2010) Staple food prices in Uganda. Tech. rep.
5. Jones DR (2000) Diseases of banana, abaca and enset. CABI Publishing
6. Kimunye JN et al (2020) Distribution of Pseudocercospora species causing Sigatoka leaf diseases of banana in Uganda and Tanzania. Plant Pathol 69(1):50–59
7. Gokula Krishnan V et al (2022) An automated segmentation and classification model for banana leaf disease detection. J Appl Biol Biotechnol 10(1):213–220
8. Marvin G, Alam MGR (2022) Explainable augmented intelligence and deep transfer learning for pediatric pulmonary health evaluation. In: 2022 international conference on innovations in science, engineering and technology (ICISET). IEEE, pp 272–277
9. Marvin G et al (2023) Local interpretable model-agnostic explanations for online maternal healthcare. In: 2023 2nd international conference on smart technologies and systems for next generation computing (ICSTSN). IEEE, pp 1–6
10. Mduma N et al (2022) The Nelson Mandela African Institution of Science and Technology Bananas dataset. Version V2. https://doi.org/10.7910/DVN/LQUWXW
11. Mytholla N, Chakshu V (2021) Image classification using convolutional neural networks by different activation functions. Image 8(7)
12. Orchi H, Sadik M, Khaldoun M (2022) On using artificial intelligence and the internet of things for crop disease detection: a contemporary survey. Agriculture 12(1):9
13. Salami A, Kamara AB, Brixiova Z (2010) Smallholder agriculture in East Africa: trends, constraints and opportunities. African Development Bank Tunis, Tunisia
14. Taneja M et al (2019) SmartHerd management: a microservices-based fog computing-assisted IoT platform towards data-driven smart dairy farming. Softw: Pract Exp 49(7):1055–1078
15. Tumuhimbise R (2013) Breeding and evaluation of cassava for high storage root yield and early bulking in Uganda. PhD thesis
16. Tushemereirwe W et al (2015) Performance of NARITA banana hybrids in the preliminary yield trial for three cycles in Uganda
17. Zhu M, Gupta S (2017) To prune, or not to prune: exploring the efficacy of pruning for model compression. arXiv preprint arXiv:1710.01878

Modified U-Net and CRF for Image Segmentation of Crop Images



Shantanu Chakraborty, Rushikesh Sanap, Muddayya Swami, and V. Z. Attar

Abstract Smart agriculture is the need of the hour, especially in a country like India whose economic and cultural backbone is agriculture. Detection of anomalies in crop images, like the presence of diseased spots and the presence of weeds is one of the most important strides in this direction. An important pre-requisite to this is the segmentation of crop images. A lot of related work has been done on the segmentation of crop images, using state-of-the-art neural network-based architecture and image processing-based techniques. This paper explores neural networks-based encoder–decoder architectures, particularly U-Net and introduces additions/modifications to it. On evaluating, the results of this modified U-Net are assessed qualitatively and quantitatively, it was found that the modified U-Net led to better results in terms of segmentation accuracy.

Keywords U-Net · Image segmentation · Residual block · Residual path · Conditional random fields

S. Chakraborty (✉) · R. Sanap · M. Swami
Department of Computer Engineering and IT, COEP Technological University, Pune 411005, India
e-mail: chakrabortyss19.comp@coeptech.ac.in

R. Sanap
e-mail: sanaprv19.comp@coeptech.ac.in

M. Swami
e-mail: swamimd19.comp@coeptech.ac.in

V. Z. Attar
School of Computational Science, COEP Technological University, Pune 411005, India
e-mail: vahida.comp@coeptech.ac.in

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2023

S. Shakya et al. (eds.), *Fourth International Conference on Image Processing and Capsule Networks*, Lecture Notes in Networks and Systems 798,
https://doi.org/10.1007/978-981-99-7093-3_13

1 Introduction

Agriculture has been the backbone of India for many years, economically and culturally. Owing to a declining natural resource base, dwindling agriculture areas and outbreaks of pests and diseases, maintaining this would be a huge problem. This has led to an increasing interest in smart agriculture and its applications. Detection of anomalies in crop images, like weeds and diseased spots is one of the first and important steps toward revolutionizing agriculture. An important step that goes hand-in-hand with this is the segmentation of crop images for object detection and classification. As discussed in Mohanty et al. [9], deep learning and neural networks have been crucial in designing architectures and models for image detection. Long et al. [10] throw light on the extensive use of neural networks, particularly convolutional neural networks for semantic segmentation. Neural networks in conjunction with other techniques, like transfer learning also show great promise for designing robust image segmentation models, as illustrated in Huang et al. [3] and Yang et al. [4]. Many works have been published in recent times related to this field. This paper is an attempt to propose certain modifications to certain state-of-the-art architectures and assess the performance of the proposed architecture using benchmark datasets. Section 2 of the paper gives an overview of related work relevant to the problem concerned. This is followed by Sect. 3 which discusses our proposed architecture at length. Section 4 presents a qualitative and quantitative evaluation of our architecture in contrast with U-Net, allowing us to draw certain conclusions in Sect. 5.

2 Related Work

2.1 U-Net

Ronneberger et al. [1] proposed U-Net (Fig. 1). U-Net is an encoder–decoder network that consists of two major components: a contraction/encoding pathway and a decoding/expansion pathway.

The contraction network consists of four encoder blocks, each of which contains two convolutional layers and a max pooling layer. The activation function used in convolutional layers is ReLu. The use of the max pooling layer downsamples the size by a factor of 2. The purpose of this block is to extract spatial features from the images. The output of these encoder blocks is a feature map that has all the encoded features.

The expansion network consists of four decoder blocks, each of which consists of a transposition convolutional layer that increases the height and width of incoming feature maps by a factor of 2, but decreases the number of feature maps by a factor of 2. This is followed by two more convolutional layers to increase the number of feature maps.

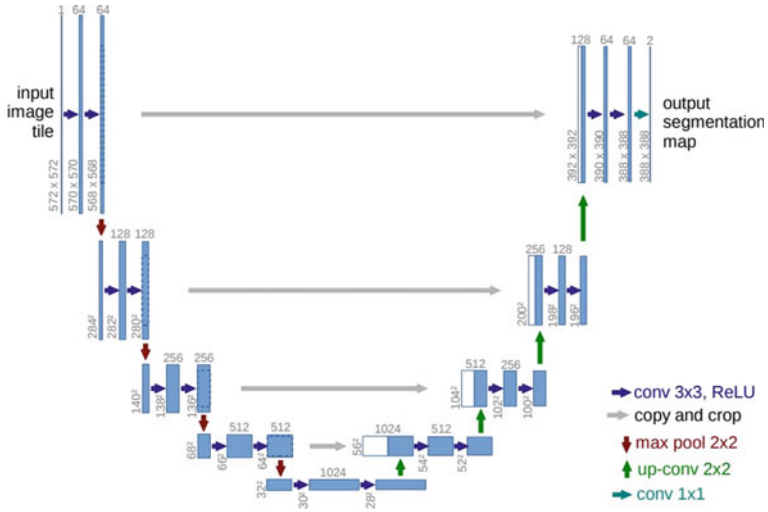


Fig. 1 U-Net

The feature maps outputted by the encoder blocks are spliced with the input to the decoder blocks with the help of skip connections. This causes the retention of spatial information to an extent. However, this does not solve the issue of a considerable semantic gap between the output of the encoder and the input of the decoder.

2.2 Residual Block (ResBlock)

The main role of convolutional layers is to extract features from images. The more CNNs present, the deeper the features extracted will be. However, if we keep adding more and more convolutional layers, the accuracy in the training set decreases. Since we are using a gradient descent-based optimizer (Adam optimizer), there exists a possibility that adding more convolutional layers will lead to finding local optimum solutions instead of finding global optimum solutions. A possible way of countering this is to replace multiple convolutional layers with a residual block introduced by He et al. [7]. Residual block extracts deep features from an image without facing the issues mentioned above.

The structure of the residual block is shown in Fig. 2. Unlike feed-forward networks where each layer feeds the next layer, a residual block feeds the next layer and also the layer a couple of hops away. And finally, the input to the next layer is subjected to an activation function before concatenating it with the original feature vector which was fed to a layer a couple of hops away.

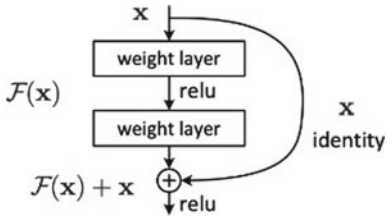


Fig. 2 ResBlock

2.3 Residual Path

In standard U-Net architecture, encoders extract low-level features. The convolutional layers present in the encoders, create feature maps while doing so. By splicing the low-level features with high-level features, the spatial information lost is retained. As a result of this, high-resolution details that were lost are retained and the overall segmentation output is refined. However, there exists a semantic gap between the low-level features and the high-level features; thus, their direct fusion is not appropriate.

To counter this issue, the proposed architecture makes use of a "Residual Path" (ResPath) instead of U-Net's skip connection. ResPath introduces some additional convolutional layers before concatenating low-level features with high-level features. This results in a reduced semantic gap and a better learning ability.

Figure 3 shows the structure of ResPath. It consists of four 3*3 convolutional layers and four 1*1 convolutional layers and four Resblocks. The low-level feature maps are spliced with the high-level feature maps by ResPath.

3 Proposed Architecture

U-Net has been used for a variety of image segmentation applications. However, its application for crop images is not prevalent. This section discusses the proposed additions and modifications to the existing U-Net architecture to improve the performance of crop images. The proposed and modified U-Net for crop images is a symmetric

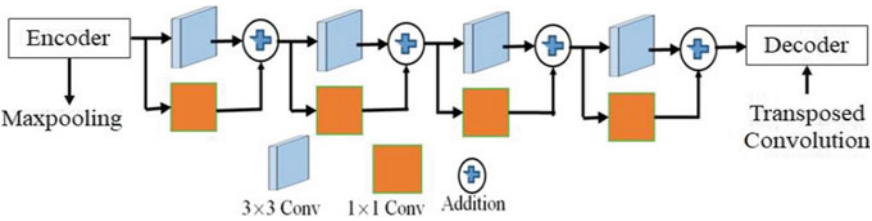


Fig. 3 ResPath

architecture with two major parts—a contraction network (encoding path) and an expansion network (decoding path). Each of these networks has four encoder and decoder blocks, respectively, whose architecture is described in detail below.

3.1 Selection of Algorithm

U-Net is a standard architecture for image segmentation and has enjoyed success with various benchmark datasets. As described in Gurita et al. [2] and Chen et al. [11], variations of encoder–decoder networks have been enjoying great success for image segmentation use cases. CED Net proposed by Khan et al. [5] also propose tweaks in the U-Net architecture to make it more efficient and accurate for crop images. Thus, U-Net as a backbone for the proposed architecture for segmentation of crop images shows great promise. Thus, the main motive of our proposed architecture is to improve upon this by addressing the issues that arise while using U-Net for crop images—namely a saturation of accuracy on increasing the number of convolutional layers, a significant semantic gap between the low-level feature maps outputted by encoder blocks and the high-level feature maps fed as input to decoder blocks, which make their splicing improper. The proposed architecture was devised with the aim to address these issues while leveraging the proven success of U-Net with multiple benchmark datasets. The proposed architecture uses CNN at the root of its functioning, because segmentation requires extraction of features, both low-level and high-level, and convolutional neural networks are by far the best deep learning paradigms for doing so.

As shown in Fig. 4, each encoder has two convolutional layers of size 3×3, followed by two ResBlocks. The first ResBlock doesn't use any activation function, however, the second ResBlock uses the ReLu activation function. The stride of each convolutional layer is 1. The resultant of this is subjected to a max pooling layer of size 2×2. After the application of the max pooling layer, the number of filters is doubled, but the size is downsampled by a factor of 2.

Every decoder block consists of a transposition convolutional layer of size 2×2 and stride of 2, followed by two convolutional layers of size 2×2 and stride 1, and two ResBlocks. The first ResBlock doesn't make use of any activation function; however, the second ResBlock uses the ReLu activation function.

U-Net makes use of skip connections to fuse low-level features with high-level features for the retention of spatial information. A downside to this approach, as mentioned previously, is the presence of a considerable semantic gap between low-level and high-level features. Our proposed architecture makes use of ResPath instead of skip connections. The use of ResPath leads to better retention of spatial information and addresses the issue of semantic gap.

An en-decoder block connects the contraction network and the expansion network.

All convolutional layers except the last one use ReLu as an activation function. The final convolutional layer uses sigmoid as an activation function.

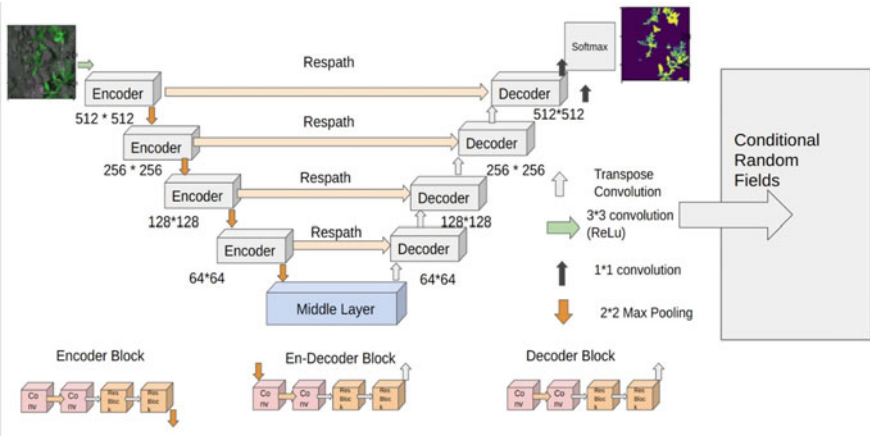


Fig. 4 Proposed architecture

The output of the abovementioned architecture is coarse. To refine it further, the segmented images are subjected to conditional random fields. The following subsection discusses CRF and its implementation in our proposed architecture in detail.

3.2 Conditional Random Field (CRF)

Conditional random field (CRF) is a statistical modeling technique that is applied when the class labels associated with various inputs are dependent on each other. In the context of the problem of image segmentation, the label associated with every pixel depends on the pixels in its neighborhood too. Discrete classifiers do not take this dependence into account, however, CRF does. In cases where the correlation between input data points and the labels is high, such classifiers do not perform well. CRF, on the other hand, assigns labels to data points (pixels, in our case) by considering the labels of the labels of input points dependent on it too. Thus, CRF postprocessing of segmented images refines them. The proposed architecture makes use of fully connected CRF which helps in the refinement of segmentation outputs by considering similar pixels in the neighborhood, calculating the unary and pairwise energies of these standalone pixels and pairs of pixels and attempting to minimize the energy, thus leading to a more precise labeling of the pixels. The following description explains the succinct description in more detail.

Krähenbühl et al. [6] introduce the use of "fully connected pairwise CRFs". This means that all the locations in the image are connected to each other in pairs. An edge between pixel i and pixel j indicates that the labels of these two pixels are dependent on each other. Our proposed architecture makes use of fully connected pairwise CRFs.

CRF is characterized by Gibb's distribution of the following form:

$$P(\boldsymbol{X} = x|\boldsymbol{I}) = \frac{1}{z(\boldsymbol{I})} \exp(-E(x|\boldsymbol{I})).$$

Here, **I** is the input, $E(x|\mathbf{I})$ is the energy function, **X I**s the random value corresponding to location **I** which represents its predicted label and $z(I)$ is the partition function which is obtained by the summation of all $\exp(E(x \mid I))$.

A simple interpretation of CRF can be presented this way: our prime goal in CRF is to minimize the energy function. Thus the energy function can be thought of as a cost function. By assigning the most probable label (x) to a given location (I), the energy function can be minimized, consequently leading to higher accuracy.

The energy function for CRF can be defined this way:

$$E(x) = \sum_i \psi_u(x_i) + \sum_i \psi_p(x_i, x_j).$$

Unary energy $\psi_u(x)$: The measure of the cost if the label assignment disagrees with the initial classifier.
Pairwise energy $\psi_p(x_i, x_j)$: The measure of the cost if two similar pixels, i.e., pixels having similar colors or adjacent pixels take different labels.

To present in a succinct way, CRF will take the predicted masks from our modified U-Net as inputs and take into account both Unary potential and pair-wise energy terms to output a more precise and accurate segmentation mask. The Unary potential corresponds to the label of a given pixel, and the pairwise potential corresponds to the labels of the pixels adjacent to it.

Our proposed architecture implements CRF in this fashion:

Convert the segmented mask into RGB (if it is grayscale) (this can be done using gray2rgb method in Matlab):

Transforming annotated image to RGB in case it is a grayscale image if (len(annotated image.shape) < 3):

annotated image = gray 2rgb (annotated image)

Convert the RGB color of the annotations to a single 32-bit integer:

annotated image is the segmented output

annotated label = annotated image [: , : , 0] + (annotated image [: , : , 1] ≪ 8) + (annotated image [: , : , 2] ≪ 16)

Create a mapping back to 32-bit colors this can be done in the following fashion:

Convert the 32 bit integer color to 0, 1, 2, . . . labels. # np is an alias for numpy here

color, labels = np . unique (annotated label)

color int labels = np . empty ((len (colors) , 3) , np. unit 8) color int labels [: , 0] = (colors & 0 x0000FF)

color int labels [: , 1] = (colors & 0 x00FF00) >> 8 color int labels [: ,2] = (colors & 0 xFF0000) >> 16

Get Unary potentials with the help of the labels of the mask (i.e., the cost if the label of a given location is not the same as the label predicted by the initial classifier)

Get pairwise potentials (i.e., the cost if two similar pixels have different labels):

#Number of labels in the masked image fed as input to CRF architecture

number of labels = len(set (labels .flat))

using 2 dimensional CRF #DenseCRF is an alias for dcrf here if use 2 d:

The APIs used here to set up CRF and get unary as well as

pairwise potentials are from pydensecrf library

d = DenseCRF. DenseCRF2D (org img. shape [1], org img. shape [0], number of labels)

For Unary energy

U = unary from labels (labels, number of labels, get prob =0. 7, zero unsure= d. set Unary Energy (U)

For pairwise energy factoring in color independent terms # in the following, k is usually set to 3

d. add Pairwise Gaussian (sxy= (k, k), compat=k, kernel=DenseCRF. DIAG KERNEL, normalization=DenseCRF. NORMALIZE SYMMETRIC)

This factors in color dependent terms

d. add Pairwise Bilateral (sxy = (80, 80), srgb = (13, 13, 13),

rgbim=org img, compat=10,

kernel=DenseCRF. DIAG KERNEL,

normalization=DenseCRF.NORMALIZE SYMMETRIC)

We need to compute the most likely label assignment for every pixel. Inference can be run using any inference algorithm for graphical models.

running inference 5 times

Q is the output after running inference 5 times (ideally)

Find out the most probable class for each pixel.

Estimate the most likely class for each pixel. LABELS = np. argmax (Q, a xis = 0)

Convert the labels back to their corresponding colors and save the image.

Convert the LABELS (labels) back to the colors corresponding to them

LABELS = color int labels [LABELS,:]

Reshape the resultant image according to the original image that was # given as an input.

4 Results and Discussions

For evaluating our proposed architecture against standard U-Net, we conducted experiments on two datasets.

- *Crop Weed Field Image Dataset (CWFID):* CWFID dataset comprises field images, segmentation masks and appropriate annotations. This dataset was published in Haug et al. [8]. The number of images in this dataset is 60. Since the images were quite meager in amount, basic data augmentation was performed by flipping the image about various axes. This dataset was acquired using an autonomous robot Bonirob in an organic carrot farm when the carrots were in the early true leaf growth stage.
- *Open Sprayer Dataset:* This is a collection of crops and weeds that were captured using an autonomous land drone. There are up to 1000 images in this dataset for training. The huge number of data points present is conducive to our project. It includes pictures of broad-leaved docks and pictures of land without broad-leaved docks. The dataset can be downloaded from kaggle.

4.1 Qualitative Evaluation

Figures 5, 6, 7 and 8 give a qualitative idea of the performance of our proposed architecture and the performance of standard U-Net architecture for images from CFWID as well as Open Sprayer Images datasets.

4.2 Quantitative Evaluation

We used a metric known as intersection over union (IoU), which is frequently used for image segmentation and other similarity challenges, for the quantitative assessment

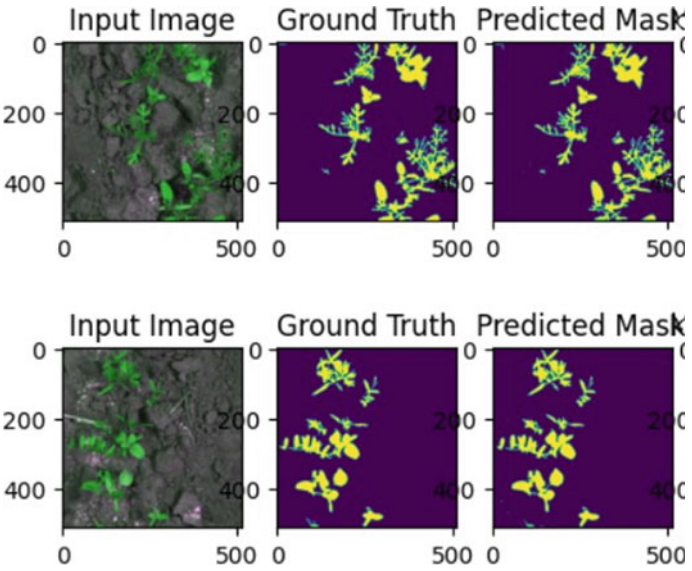


Fig. 5 Image segmentation of images from CWFID dataset using standard U-Net

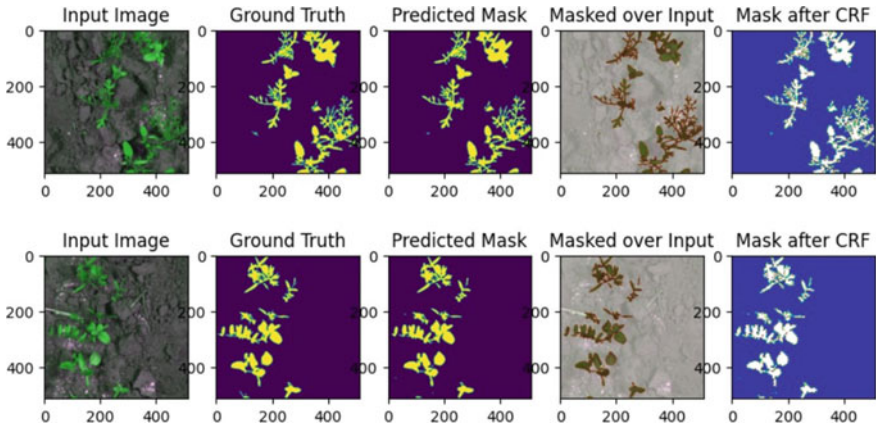


Fig. 6 Image segmentation of images from CWFID dataset using our proposed U-Net and CRF postprocessing

of our suggested model. IoU is calculated by dividing the area of the union of the expected mask and the ground truth by the area of the overlap between the predicted mask and the ground truth. IoU values are between 0 and 1. The ground truth and predicted masks do not overlap at all, as indicated by the value 0. The ground truth and anticipated mask completely coincide, as indicated by the number 1.

IoU is defined as follows:

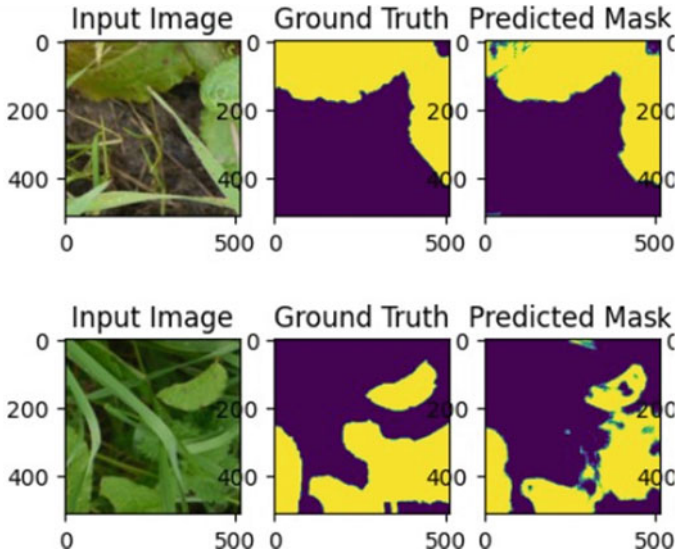


Fig. 7 Image segmentation of images from Open Sprayer dataset using standard U-Net

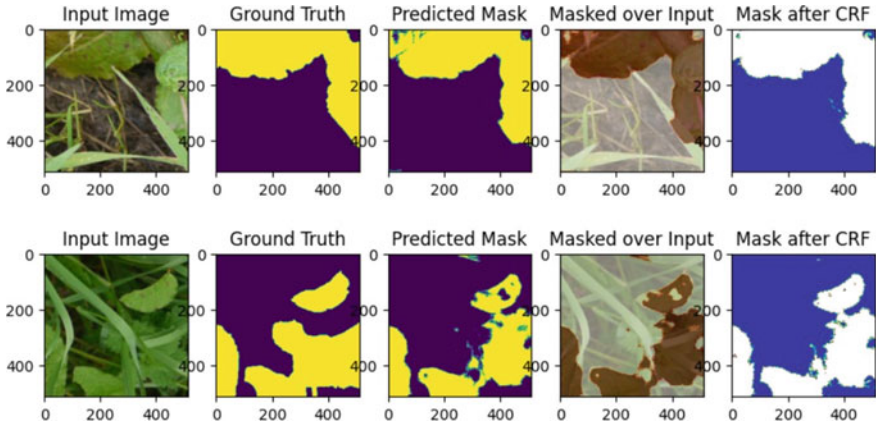


Fig. 8 Image segmentation of images from Open Sprayer dataset using standard U-Net and CRF postprocessing

$$\mathbf{G} = \mathbf{Groundtruth P} = \mathbf{PredictedMask IoU} = \mathbf{G} \cap \mathbf{P} / \mathbf{G} \cup \mathbf{P}$$

Tables 1 and 2 provide a quantitative evaluation of our proposed architecture against standard U-Net. Figures 9 and 10 visualize these statistics for better understanding.

Table 1 IOU values for standard U-Net

IOU	CWFID	Open Sprayer
Mean IOU before CRF	0.9427	0.737
Mean IOU after CRF	0.930	**0.743**
Min IOU before CRF	0.918	0.070
Min IOU after CRF	0.898	0.0789
Max IOU before CRF	0.959	0.985
Max IOU after CRF	0.957	0.9871

Table 2 IOU values for our proposed architecture

IOU	CWFID	Open Sprayer
Mean IOU before CRF	0.9459	0.737
Mean IOU after CRF	0.945	**0.758**
Min IOU before CRF	0.915	0.030
Min IOU after CRF	0.888	0.030
Max IOU before CRF	0.965	0.985
Max IOU after CRF	0.955	0.9871

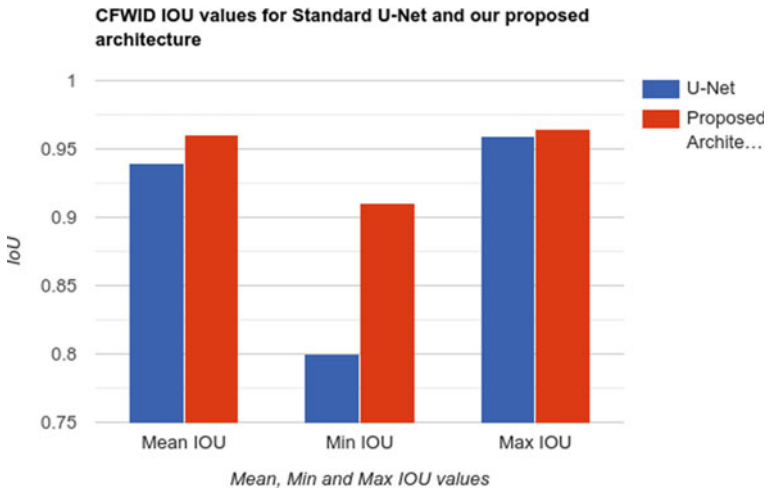


Fig. 9 Comparison of IOU values for U-Net and our proposed architecture for CFWID dataset

It can be seen that the mean IOU values corresponding to our proposed architecture is either greater than the mean IOU values corresponding to standard U-Net, for both CFWID and Open Sprayer Images datasets.

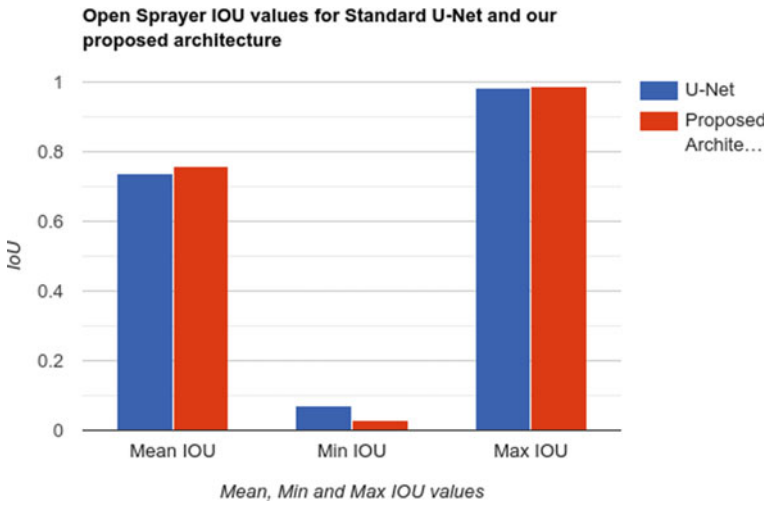


Fig. 10 Comparison of IOU values for U-Net and our proposed architecture for Open Sprayer dataset

4.3 Retention of Spatial Information

One of the major issues in standard U-Net is that, due to multiple encoders and convolutional layers a lot of spatial information is lost during the encoding process. This spatial information is retained by splicing the feature map outputted by the encoder blocks with the input of the decoder blocks. These connections between the feature maps outputted by encoders and the feature maps fed as input to the decoder blocks are called skip connections. However, these skip connections don't retain spatial information robustly and don't address the issue of the semantic gap between two combined feature sets. Our proposed architecture makes use of a residual path (ResPath) instead of skip connections to address the issue of semantic gap and to retain spatial information to a greater extent. The following images show how the usage of ResPath allows our model to retain spatial information better than standard U-Net.

It can be seen from Figs. 11 and 12 that the leaves obscured by weed are segmented by U-Net, but only to a slight extent. However, our proposed architecture segmented a greater area of the obscured leaf, proving that spatial retention is better using our proposed architecture.

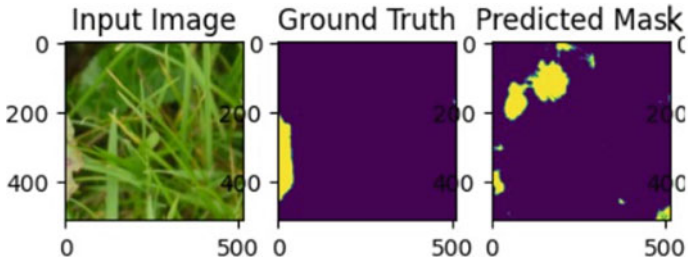


Fig. 11 Image segmentation of images from Open Sprayer dataset using standard U-Net

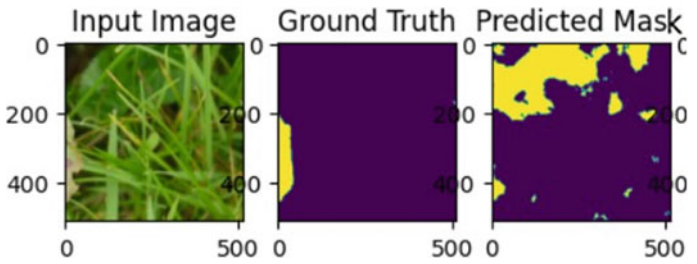


Fig. 12 Image segmentation of images from Open Sprayer dataset using our proposed U-Net architecture

5 Conclusion

Keeping in mind the problem of segmentation of crop images, an improved U-Net is proposed to tackle certain issues that could occur in standard U-Net. Our proposed modified U-Net makes use of ResBlock which allows deeper feature extraction without encountering issues like vanishing gradient and leading to an improvement in the network performance. For the aim of splicing low-level features with high-level features, U-Net skip connections are replaced with ResPath. This has resulted in enhanced spatial information retention and a narrowing of the gap between low-level and high-level features. The experimental findings suggest that our proposed modified U-Net outperforms regular U-Net in terms of segmentation accuracy. For both the CFWID and Open Sprayer datasets, the mean intersection over union values of segmentation using our modified U-Net architecture outperform the mean IoU values of the traditional U-Net architecture.

However, an exhaustive comparative study of our proposed architecture with multiple datasets will shed more light on its performance and robustness. The addition of ResPath and ResBlock increases the performance quantitatively and qualitatively but it also introduces more trainable parameters, which leads to a greater training time.

References

1. Ronneberger O, Fischer P, Brox T (2015) U-Net: convolutional networks for biomedical image segmentation. arXiv:1505.04597v1
2. Gurita A, Mocanu IG (2021) Image segmentation using encoder-decoder with deformable convolutions. Natl Libr Med 21(5):1570
3. Huang H-N, Zhang T, Yang C-T, Sheen Y-J, Chen H-M, Chen C-J, Tseng M-W (2022) Image segmentation using transfer learning and Fast R-CNN for diabetic foot wound treatments. Natl Libr Med 10:969846
4. Yang S, Zheng L, He P, Wu T, Sun S, Wang M (2021) High-throughput soybean seeds phenotyping with convolutional neural networks and transfer learning. Plant Methods 17(50)
5. Khan A, Ilyas T, Umraiz M, Mannan ZI, Kim H (2020) CED-Net: crops and weeds segmentation for smart farming using a small cascaded encoder-decoder architecture. Electronics 9(10):1602
6. Krähenbühl P, Koltun V (2012) Efficient inference in fully connected CRFs with Gaussian edge potentials. arXiv:1210.5644
7. He K, Zhang X, Ren S, Sun J (2015) Deep residual learning for image recognition. arXiv:1512.03385v1
8. Haug S, Ostermann J (2015) A crop/weed field image dataset for the evaluation of computer vision based precision agriculture tasks. In: ECCV 2014 workshops, Part IV, LNCS 8928, pp 105–116
9. Mohanty SP, Hughes DP, Salathé M (2016) Using deep learning for image-based plant disease detection. Front Plant Sci 7:1419
10. Long J, Shelhamer E, Darrell T Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern
11. Chen L-C, Zhu Y, Papandreou G, Schroff F, Adam H (2018) Encoder-decoder with atrous separable convolution for semantic image segmentation. In: Proceedings of the European conference on computer vision (ECCV), Munich, Germany, 8–14 Sept 2018

Securing Data in the Cloud: The Application of Fuzzy Identity Biometric Encryption for Enhanced Privacy and Authentication



Chandrasekar Venkatachalam, K. Manivannan, and Shanmugavalli Venkatachalam

Abstract Cloud computing has emerged as a highly innovative technology, offering various benefits to users. To ensure the security of user data, cloud storage schemes have been introduced, aiming to protect sensitive information from unauthorized access. In particular, the sharing of personal health records (PHR) is gaining prominence as a method for securely sharing healthcare data among users on cloud servers. The use of fuzzy techniques plays a crucial role in transforming original data into an encrypted form, known as ciphertext, which is challenging for unauthorized individuals to comprehend. This technique enhances the confidentiality of data, ensuring that only authorized parties can access and understand it. However, while cloud services provide a convenient platform for data sharing, they often lack efficiency in terms of data sharing capabilities. To address these challenges, a novel approach called fuzzy identity biometric encryption (FIBE) is introduced for PHR management. FIBE combines the benefits of fuzzy techniques and biometric authentication to achieve both high-security levels and user convenience simultaneously. This approach enables authorized users to have control over access to PHR data and ensures secure data sharing within a cloud environment. By integrating biometric authentication, FIBE enhances the security of PHR systems, as biometric characteristics are unique to each individual, making it difficult for unauthorized users to gain access. Moreover, the approach improves user convenience by eliminating the need for remembering complex passwords or using traditional authentication methods. In conclusion, the utilization of fuzzy identity biometric encryption (FIBE)

C. Venkatachalam (✉) · K. Manivannan
Department of Computer Science and Engineering, Faculty of Engineering and Technology, Jain (Deemed-to-be University), Bengaluru, Karnataka, India
e-mail: chandrasekar.v@jainuniversity.ac.in

K. Manivannan
e-mail: manivannan.k@jainuninversity.ac.in

S. Venkatachalam
Department of Computer Science and Engineering, KSR College of Engineering, Tiruchengode, Tamil Nadu, India
e-mail: drvshanmugavalli@gmail.com

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2023

S. Shakya et al. (eds.), *Fourth International Conference on Image Processing and Capsule Networks*, Lecture Notes in Networks and Systems 798,
https://doi.org/10.1007/978-981-99-7093-3_14

in PHR systems offers enhanced security and efficient data sharing in the cloud. This approach combines the advantages of fuzzy techniques and biometric authentication, providing authorized users with control over data access while maintaining a high level of security and user convenience.

Keywords Cloud computing · PHR · Intruders · Fuzzy technique · Fuzzy identity biometric encryption · Biometric authentication

1 Introductıon

Cloud computing has revolutionized the way clients access information, allowing them to retrieve data from anywhere and at any time through the Internet. One of the emerging applications in healthcare is the personal health record (PHR) system, which follows a patient-centric model of health information exchange. It empowers patients to create and manage their own PHR data in a centralized location using web-based applications. With the advent of cloud computing, individuals are increasingly utilizing this environment for efficient storage and retrieval of their individual PHRs. Storing PHRs in the cloud offers benefits such as cost reduction and simplified communication and access systems.

However, a primary concern in PHR systems is whether patients can maintain control over their own data. It is crucial to establish fine-grained access control and ensure the security of sensitive data stored in third-party cloud storage. The challenges lie in maintaining security, privacy, and health information authentication when PHRs are kept in the cloud.

To address these challenges, strong security measures and access controls are necessary. Encryption techniques, access management protocols, and authentication mechanisms can be employed to protect the confidentiality, integrity, and availability of sensitive health information. By implementing secure authentication protocols and leveraging advanced cryptographic techniques, the authenticity of data and the authorized access to PHRs can be ensured.

Furthermore, the usage of fuzzy-based biometric encryption (FBBE) provides a promising solution. FBBE considers identities as a set of descriptive attributes, allowing users with the secret key for a specific identity to decrypt ciphertexts encrypted with the corresponding public key. This approach offers high computation efficiency and security, addressing key management issues and providing scalability to the PHR framework.

The integration of cloud computing in PHR systems provides convenient access to healthcare information, but also introduces data security concerns. To mitigate these concerns, it is crucial to implement robust security measures including strong encryption techniques, access controls, and fuzzy-based biometric encryption. These measures play a vital role in safeguarding the confidentiality, integrity, and privacy of PHR data within the cloud environment. By enhancing data security, PHR systems can operate with improved efficiency, ensuring secure and user-friendly access for

both patients and healthcare providers. This ultimately leads to better healthcare outcomes and increased trust in the PHR system.

2 Related Work

Naveed et al. [1] contribute to the field of cloud security by emphasizing the importance of biometric authentication as a viable solution for enhancing security in cloud environments. By providing a structured view of different biometric techniques, the research offers valuable insights into the practical implementation and benefits of biometric authentication in the cloud.

Boneh and Franklin [2] introduced the first secure and practical identity-based encryption (IBE) scheme based on bilinear pairings. This groundbreaking work has sparked significant interest in IBE, leading to the development of numerous schemes and related systems. The ongoing research and development in the field of IBE aim to further optimize and enhance the performance of encryption schemes for multiple recipients. By reducing computational complexity and communication overhead, these advancements enable more efficient and scalable encryption techniques, ultimately benefiting various applications that require secure communication among multiple parties.

Canetti et al. [3] made a significant contribution to the field of identity-based encryption (IBE) by proposing the first construction that achieved provable security beyond the random oracle model. In their work, they introduced a slightly weaker model of security called the Selective-ID model. The introduction of provable security beyond the random oracle model in the context of IBE has paved the way for further advancements and developments in this area. Researchers have since explored new techniques, refined existing constructions, and extended the security analysis of IBE schemes, leading to improved protocols and a deeper understanding of their security properties.

Selvarani et al. [4] introduced a method to enhance data security in the cloud environment by employing a multimodal biometric technique. This approach aims to strengthen the protection of data during its transmission and storage. The encrypted data is then stored in the cloud environment, ensuring that it remains protected and inaccessible to unauthorized entities. When the data needs to be retrieved, a decryption technique is employed to convert the ciphertext back into its original plaintext form.

By employing the combination of encryption and multimodal biometrics, the proposed method enhances data security in the cloud environment. It provides a layered approach to protect sensitive information during transmission and storage, mitigating potential risks and unauthorized access attempts.

Rinesh et al. [5] introduced a method called multi-attribute attribute-based encryption (MA-ABBE) that addresses the access control policy in personal health records (PHR) systems. While many research studies have focused on the security issues of PHR, achieving efficient data sharing has remained a challenge.

The MA-ABBE method offers several advantages, including enhanced security, authentication, and efficient data sharing in the cloud environment. It provides a mechanism to control access to PHR data, protecting sensitive healthcare information from unauthorized access or breaches. Additionally, the use of biometric fingerprint authentication adds an extra layer of security, ensuring that only authorized users can access the PHR. Overall, the MA-ABBE approach addresses the security and efficient data sharing challenges in PHR systems, offering a reliable solution for access control and protecting the privacy of healthcare information.

Abdolahi et al. [6] addressed the issue of single biometric spoof attacks, which can result in unacceptable error rates due to noise or other factors. They proposed the use of a multimodal biometric system as a solution to mitigate these attacks and improve recognition accuracy. The decision-making process in the multimodal biometric system involves combining the recognition scores or features extracted from both fingerprint and iris biometric systems. Various fusion techniques, such as score-level fusion or feature-level fusion can be employed to obtain a final decision.

Wagh et al. [7] introduced a multimodal biometric system that utilizes the characteristics of both fingerprint and iris modalities. The system incorporates feature-level fusion to combine the features extracted from fingerprint and iris biometric templates. The fused features are then subjected to encryption using various security technologies. The experimental outcomes showed that the multimodal biometric system achieved higher accuracy compared with the unimodal biometric system. This improvement in accuracy can be attributed to the synergistic effect of combining the distinctive features of fingerprint and iris modalities. By leveraging the strengths of both modalities, the system becomes more robust and capable of handling variations in biometric data, leading to enhanced recognition performance.

Thaiyalnayaki et al. [8] proposed a fingerprint recognition framework that addresses the limitations of minutiae-based methods by incorporating surface analysis and utilizing a combination of features for multiscale and multidirectional recognition. The framework incorporates various silent features such as standard deviation, skewness, and kurtosis. The proposed framework aims to improve the performance of fingerprint recognition systems by reducing the impact of false minutiae and addressing the limitations of traditional minutiae-based approaches. By incorporating surface analysis and a combination of features, the framework provides a more comprehensive and reliable approach to fingerprint recognition.

Pokhriyal et al. [9] proposed a methodology for fingerprint verification that utilizes wavelets and Pseudo Zernike moments (PZMs) to extract both global and local features. PZMs are robust to noisy images, invariant to rotation, and have good image reconstruction capabilities, making them useful for global analysis and extracting global features such as the shape of the fingerprint image. On the other hand, wavelets are effective in local analysis and help extract local features or minutiae from a fingerprint. The proposed methodology contributes to the field of fingerprint verification by leveraging the strengths of PZMs and wavelets, resulting in a more robust and effective approach for feature extraction and fingerprint matching.

Gao et al. [10] have developed a novel topology-based representation system for fingerprint verification. This system utilizes neighbor structure data to match the

point patterns initially and then performs global matching. The proposed representation is designed to be invariant to rotation and translation by analyzing the relationships among minutiae, both genuine and spurious, without capturing extensive information about fingerprints. By using the relationship of point sets for matching, the proposed representation exhibits robustness to missing and spurious minutiae. This approach overcomes the limitations of relying solely on individual minutiae and provides a more comprehensive and resilient matching technique. Overall, the topology-based representation system proposed by Gao et al. offers a promising approach for fingerprint verification by leveraging the relationships among minutiae and providing robustness to variations in rotation, translation, and image resolution.

Prasath et al. [11], the authors focus on improving the accessibility of cloud computing by utilizing fuzzy logic. They propose a conceptual model that employs various attributes to evaluate the satisfaction of cloud computing users in an Internet service provider (ISP) context. To handle the inherent ambiguity and uncertainties associated with linguistic methods, the authors employ a fuzzy inference system (FIS). By using fuzzy logic, they aim to avoid any ambiguities that may arise during the evaluation process. By utilizing fuzzy logic and the FIS, the authors aim to provide a more comprehensive and accurate evaluation of user satisfaction in cloud computing services. This approach enables the consideration of multiple attributes simultaneously, allowing for a more nuanced understanding of user experiences and requirements.

The implementation proposed by Velciu et al. [12]. Focuses on enhancing the security of authentication methods in cloud storage. They introduce the concept of a bio-cryptographic infrastructure, which aims to provide safer access and encryption support within cloud storage sharing and cloud platforms. Specifically, their work implements a fuzzy vault authentication mechanism based on voice recognition. The fuzzy vault is a cryptographic concept that allows secure storage of private information while providing the ability to retrieve it even when there are slight variations in the input. In this case, the voice of the user serves as the basis for authentication.

This method helps ensure that only authorized individuals with a verified voice pattern can access the stored data and perform encryption operations. Overall, the intention of this work is to provide a more robust and secure authentication mechanism for cloud storage, leveraging voice recognition within the fuzzy vault framework.

3 System Model

The basic fuzzy selective-ID game is allowed to query for secret keys identities that are greater than **d** Hamming Distance away from the targeted identity [13].

4 Basic Fuzzy Selective-ID

Init: The adversary (A) declares the identity x that he wishes to be challenged upon.

key Setup: The challenger runs the key setup phase of the algorithm and the public parameters of adversary.

Phase 1: The adversary is allowed issue queries private keys for an identity y, where the Hamming distance between x and y is greater than d.

Challenge the adversary submits two equal length messages m_1, m_2, the challenger flips a random coin c and encrypts m_c. The cipher text is passed to the adversary.

Phase 2: Phase 1 is Repeated

Guess: The output of the adversary c' of c.
The main advantage A in this game is Pr $[c' = c] - 1/2$.

Attribute Identity Authentication (AIA)

A user can decide to move to an attribute identity authentication (AIA) and demonstrate that it is qualified to unspecified amount of the attributes managed by that authority, and request related to decryption keys. The attribute identity key generation algorithm will run by authority. Any user can perform encryption and decryption process when they should create key setup and key generation algorithm.

Key Setup

The key setup algorithm takes the implicit security parameter. It generates the public and master key parameters P and M generated by each AIAs (attributes identity authorities). The nth AIA defines a disjoint set of attributes Ur, which are similar for public users. These attributes are classified in their features.

$$\text{Key Setup}\,(1, N) \rightarrow \text{via}\,(\text{params}, \{(P_k, M_k)\}k \in \{1, N\})$$

Key Generation *(M, S)*

The key generation algorithm uses key setup algorithm's master key M and set of identity attributes UIr that describe the key and output of the secret key S for user U. S should contain at least one attribute from every type of attributes governed by AIA.

$$\text{A Key Gen}\,(S_k, \text{UID}, \text{IA}_k) \rightarrow \text{via}\, S_k[\text{UID}, \text{IA}_k],$$

where

UID *user with identity*
S_k *secret key*
IA_k *identity attribute set.*

Biometric Fingerprint Encryption

The biometric fingerprint-based encryption method is the oldest method for identification which has been successfully used in numerous applications. Everyone is having unique fingerprint and is changeless. Fingerprint identification systems often utilize both manual classification and automatic minutiae matching to identify individuals based on their fingerprints. The uniqueness of a fingerprint is determined by the pattern of ridges and valleys on the surface of the finger, as well as the specific minutiae points. Minutiae points are local ridge characteristics that occur at locations such as ridge endings or ridge bifurcations. These points serve as key reference points for fingerprint matching.

In order to identify an individual based on their fingerprint, the fingerprint data is compared with a large number of fingerprints stored in a database, e.g., the FBI database holds approximately 70 million fingerprints. Sorting and organizing these fingerprints can help reduce the search time and computational complexity. By organizing the fingerprints into subsets or categories, the data fingerprint only needs to be compared with a subset of fingerprints that are most likely to match. The process often involves manual classification, where human experts classify fingerprints into specific categories based on their overall patterns and characteristics. This manual classification step helps narrow down the search space and makes the subsequent automatic minutiae matching more efficient. After classification, automatic minutiae matching algorithms are used to compare the minutiae points of the data fingerprint with the minutiae points of the fingerprints in the selected subset. By identifying and matching minutiae points and measuring the similarity between the patterns, a fingerprint identification system can determine the likelihood of a match and provide a potential identity for the individual associated with the fingerprint.

Overall, the combination of manual classification and automatic minutiae matching enables accurate and efficient fingerprint identification in various applications, including personal health records (PHRs) and law enforcement databases.

(i) ***Minutiae Detection and Extraction***

Minutiae are specific points or features within a fingerprint that are crucial for fingerprint recognition (Fig. 1 shows overall process for minutiae detection and extraction process). These points include ridge endings, bifurcations, and dots (short ridges). Ridge endings are points where a ridge terminates, while bifurcations occur where a ridge splits into two branches. Dots or short ridges refer to small ridge segments within the fingerprint. The process of automatic minutiae detection (AMD) is vital, especially when dealing with low-quality fingerprints that may contain noise or have poor contrast. In such cases, it becomes challenging to distinguish genuine minutiae from pixel configurations that resemble minutiae or to uncover hidden minutiae [14]. Human fingerprints are unique to each individual, providing a means of personal identification. However, direct matching between an unknown fingerprint and a known fingerprint sample is sensitive to errors. Therefore, current systems primarily focus on extracting minutiae from the fingerprint image and comparing the sets of fingerprint features for matching (Fig. 2). When comparing two fingerprints, discrete features

called minutiae are used. These features correspond to specific points on the finger's friction skin, where ridges either end or split. The location of each minutia is represented by its coordinates within the fingerprint image, usually measured from a reference point in the bottom-left corner of the image.

(ii) ***Minutiae-Image Processing***

During the authentication process involving minutiae points, the following steps are typically followed:

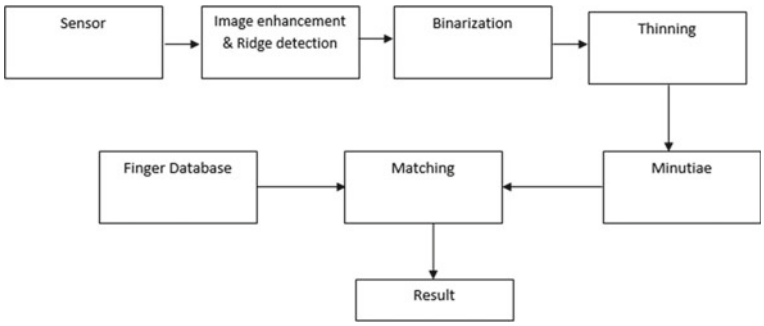


Fig. 1 Overall process for minutiae detection and extraction process

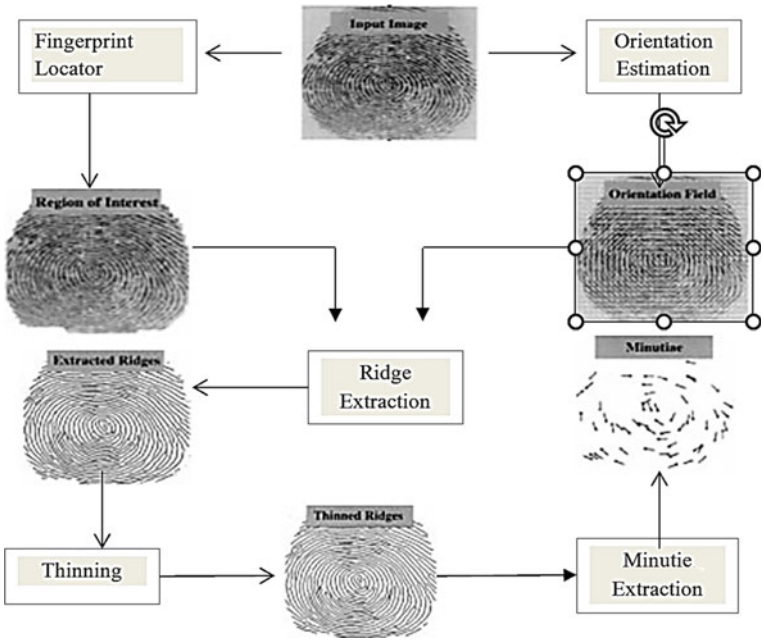


Fig. 2 Process of minutiae detection and extraction



Fig. 3 Image processing

1. Capture Fingerprint Images: Fingerprint images are acquired using a fingerprint scanner or sensor.
2. Image Processing: The captured fingerprint images undergo various image processing algorithms to enhance the quality and clarity of the image. This includes removing noise, smudges, artifacts, and healing scars or cuts. The goal is to obtain a clear and unambiguous representation of the fingerprint's skeletal image in Fig. 3.
3. Feature Extraction: The processed fingerprint image is then analyzed to extract the minutiae points. Minutiae refer to the specific details such as ridge endings, bifurcations, and dots within the fingerprint. These minutiae points are crucial for fingerprint recognition.
4. Fingerprint Matcher: The extracted minutiae points from the input fingerprint are compared with the minutiae points stored in the database. The fingerprint matcher algorithm compares the information from the input fingerprint with the records in the database to determine if there is a potential match.
5. Minutia Connections: Minutiae connections, one to another, are compared within a global context. Rather than treating the minutiae as areas within an X–Y coordinate system, they are considered as interconnected relationships. Each minutia is represented by its coordinates (x, y), and additional data chunks represent the minutia's orientation and neighborhood information.
6. Matching and Verification: If the input fingerprint's minutiae points match with those of the original fingerprint stored in the database, the authentication is successful. In this case, the health details or other authorized data can be shared with the corresponding user. However, if there is no match, no data exchange occurs, indicating that the user's identity could not be verified.

In this context, the biometric fingerprint serves as a guarantee for ensuring authentication in the personal health record system shown in Fig. 4. By verifying the matching minutiae points, the system ensures that only authorized users can access and exchange sensitive health information.

Fingerprint as Identity

The minutiae features of the fingerprint are extracted with the core point reference. The numerical value thus generated is used to create the unique ID in the form of a QR code [15]. Minutiae extraction algorithm gives the value for below fingerprint.

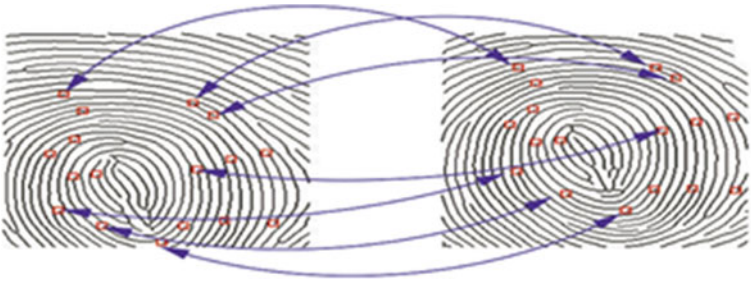


Fig. 4 Performance of minutiae matcher

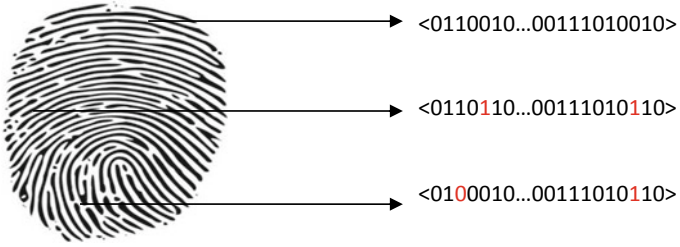


Fig. 5 Fingerprint as identity

Biometric measure changes a little each time during environment, sensors and small changes in trait are shown in Fig. 5.

Fuzzy ID-Based Encryption

The quality, efficient information offering, and authentication are characterized by the NIST framework. Matching consequences of these gatherings exhibit that the higher quality fingerprints in Fig. 6, all the more perhaps, come about a superior matching performance. Particularly for the quality number measure, the pattern that higher quality comes about better matching is clear and contrasted with the nearby quality standard, more poor matching performance fingerprints are characterized to the low-quality gathering. The great fingerprints demonstrate the authentication level of the client in PHR. Table 1 gives the unique mark results with other biometric techniques.

Efficient Data Sharing

The biometric fingerprint gives proficient data sharing to getting to personal health records. The security breakdown shows that the proposed plan is productive to safely deal with the data conveyed in the data sharing framework successfully.

Authentication

It decreases the expense for getting to the data as it could be gotten to from anyplace and at whatever time. The authentication is tolerating evidence of character given by

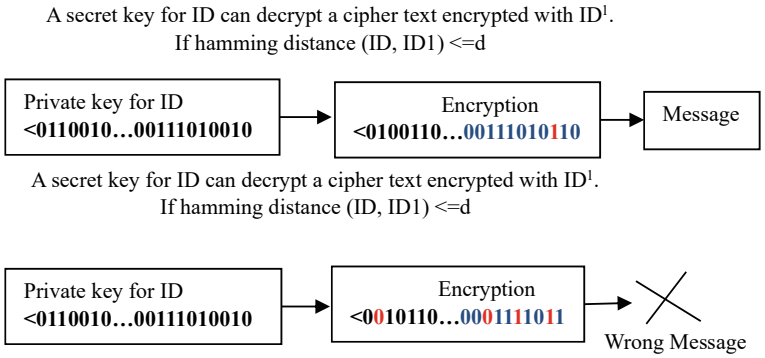


Fig. 6 Fuzzy ID-based encryption

Table 1 Different types of biometric technology (*1 (worst)–5 (best))

Technology	Accuracy	Convenience	Cost	Size
Fingerprint	5	5	4	4
Voice	1	5	5	5
Face	2	3	4	3
Hand	3	3	2	2
Iris	5	2	3	3

* represents Low

a biometric fingerprint. The methodology of distinguishing an individual normally focused on a fingerprint encryption strategy to attain the authentication. In security frameworks, authentication is distinct from authorization, which is the methodology of giving people access to framework items focused on their personality. The biometric method guarantees that approved individuals can retrieve health records (PHR) thereby preventing any unauthorized access.

5 Conclusion

The paper you mentioned introduces a concept called fuzzy identity-based encryption (FIBE), which is designed to provide an efficient platform for storing and sharing personal health records (PHR) in a cloud environment. FIBE leverages biometric encryption techniques to ensure that only authorized users can access their PHR. In traditional identity-based encryption schemes, a user's identity (such as an email address or username) is used as the public key for encryption. However, in FIBE, the notion of identity is more flexible and allows for error-tolerance. This means that slight variations or errors in the identities used to encrypt a ciphertext can still be

matched with the corresponding decryption key. By incorporating biometric encryption techniques, FIBE enhances the authentication process for accessing PHRs. Biometrics, such as fingerprints or voice patterns are unique to individuals and can serve as strong authentication factors. These biometric characteristics are utilized in the encryption and decryption process to ensure that only authorized users with the correct biometric attributes can access their PHR. Overall, FIBE provides a more robust and secure approach to storing and sharing PHRs in a cloud environment. It allows for error-tolerance in identity matching and leverages biometric encryption techniques for enhanced authentication. By utilizing FIBE, better authentication mechanisms can be achieved, ensuring that only authorized individuals can access their personal health records.

References

1. Naveed G, Batool R Biometric authentication in cloud computing. J Biom Biostat 6:258. https://doi.org/10.4172/2155-6180.1000258
2. Boneh D, Franklin M (2001) Identity-based encryption from the Weil pairing. In: CRYPTO 2001, LNCS 2139. Springer, Berlin, Heidelberg, pp 213–229
3. Canetti R, Halevi S, Katz J (2003) A forward-secure public-key encryption scheme. In: Proceedings of Eurocrypt 2003. Springer
4. Selvarani P, Visu P (2015) Multi-model bio-cryptographic authentication in cloud storage sharing for higher security. Maxwell Sci Organ 10(1):95–101
5. Rinesh S, Baskaran, Thamaraiselvi K (2015) Secure and resourceful data sharing in cloud using multiple authority attribute based biometric encryption. Int J Appl Eng Res Special issue10(21):20437–20449
6. Abdolahi M, Mohamadi M, Jafari M (2013) Multimodal biometric system fusion using fingerprint and iris with fuzzy logic. Int J Soft Comput Eng (IJSCE) 2(6):504–510
7. Wagh R, Choudhary AP (2013) Analysis of multimodal biometrics with security key. Int J Adv Res Comput Sci Softw Eng 3(8):1363–1365
8. Thaiyalnayaki K, Karim SSA, Parmar PV (2010) Fingerprint recognition using discrete wavelet transform. Int J Comput Appl 1(24)
9. Pokhriyal A, Lehri S (2010) A new method of fingerprint authentication using 2d wavelets. J Theor Appl Inform Technol
10. Gaol Z et al (2011) A novel matching technique for fingerprint recognition by graphical structures. In: Proceedings of the international conference on wavelet analysis and pattern recognition, Guilin, 10–13 July 2011
11. Prasath V, Bharathan N, Neetha NP, Lakshmi N, Nathiya M (2013) Fuzzy logic in cloud computing. Int J Eng Res Technol (IJERT) 2(3). ISSN 2278-0181
12. Velciu MA, Patrascu A, Patriciu VV (2014) Biocryptographic authentication in cloud storage sharing. In: Proceeding of the 9th IEEE international symposium on applied computational intelligence and informatics, pp 165–70
13. Sahai A, Waters B (2005) Fuzzy identity-based encryption. Int Assoc Cryptol Res 2005:457–473
14. Uludag U, Pankanti S, Prabhakar S, Jain AK (2004) Biometric cryptosystems issues and challenges. Proc IEEE 92(6):948–960
15. Lewko, Waters B (2011) Decentralizing attribute-based encryption. In: EUROCRYPT: Proceedings of 30th annual international conference on the theory and applications of cryptographic techniques: advances in cryptology, pp 568–588

Quantum Convolutional Neural Network for Agricultural Mechanization and Plant Disease Detection



Musa Genemo

Abstract Agricultural research is essential in addressing the challenges of food production and meeting the needs of a growing population. Severe crop disorders can lead to food insecurity, potentially causing disastrous effects on agricultural production, such as yield losses, reduced crop quality, and even complete crop failures. By leveraging computer vision techniques, it is possible to develop systems that detect signs of crop diseases at an early stage, facilitating prompt intervention and management. Environmental changes, such as rainfall and biosolids, can cause plants to become inflamed by microorganisms. While manual observation and assessment of leaf conditions have traditionally been used for crop disease detection, there is increasing interest in leveraging technology to automate and enhance the process. Manual detection of crop diseases based on leaf conditions is subjective, time consuming, and vulnerable to errors. Utilizing computer vision methods to detect plant diseases helps improve plant health. The concept of applying quantum convolutional neural networks (QCNNs) for agricultural mechanization and plant disease detection is an interesting area of research that combines quantum computing and deep learning techniques. However, it is important to note that quantum computing is still in its early stages, and practical implementations of QCNNs for real-world applications, including agriculture, are not yet widespread. The proposed work of paper focuses on the detection of leaf blight disease by classifying leaves into late blight, early blight, and healthy leaves. This work is divided into two stages: segmentation and classification. For segmentation, MobileNetV2 is used, and an emerging quantum machine learning technique named QCNN is used for classification. The main objective is to provide and propose an effective and fast automated technique to detect and classify blight disease, which will help farmers identify blight disease easily. The proposed model achieved a classification accuracy of 0.969. Furthermore, the results were further validated against recent findings in the field. The presented outcome implies that the model's performance is remarkable.

M. Genemo (✉)
Gumushane University, Gumushane, Turkey
e-mail: musa.ju2002@gmail.com

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2023

S. Shakya et al. (eds.), *Fourth International Conference on Image Processing and Capsule Networks*, Lecture Notes in Networks and Systems 798,
https://doi.org/10.1007/978-981-99-7093-3_15

Keywords Blight · Leaf · Image processing · Quantum convolutional neural network · Early blight · Late blight

1 Introduction

Agriculture is the only option to ensure food security in both developing and developed countries, especially as the global population continues to increase rapidly. This population growth puts significant pressure on agricultural systems to produce more food efficiently, sustainably, and resiliently. To overcome this challenge, there is no doubt that agricultural mechanization is the only solution.

Crop diseases, caused by pathogens such as fungi, bacteria, viruses, and nematodes, can result in significant yield reductions. They can infect plants at various stages of growth, affecting their development, growth, and reproductive capacity. Yield losses due to crop diseases can range from moderate to severe, depending on the pathogen, the crop's susceptibility, and environmental conditions.

Therefore, using machine learning in agriculture to detect diseases in plants early on is just as crucial as detecting diseases in humans. Plants can become infected by fungi, bacteria, and viruses due to changes in environmental factors such as rainfall, temperature, and soil fertility [1]. To effectively manage such farm activities, it is essential to use a machine learning mechanism that can detect plant diseases at an early stage.

Agricultural research focuses on developing innovative techniques, technologies, and practices to enhance crop yields, improve livestock production, and increase overall agricultural productivity. Here are a few key reasons why agricultural research is essential for meeting the requirements of an increasing population.

The concept of applying QCNNs for agricultural mechanization and plant disease detection is an interesting area of research that combines quantum computing and deep learning techniques. However, it is important to note that quantum computing is still in its early stages, and practical implementations of QCNNs for real-world applications, including agriculture, are not yet widespread. Machine learning makes judgments based on previous experience [2].

Plant disease detection often involves analyzing complex patterns and structures within images or genomic data. Traditional convolutional neural networks (CNNs) have proven effective in this regard, but QCNNs may offer the advantage of efficiently handling larger and more complex datasets. QCNNs could potentially capture intricate relationships within the data, leading to improved disease detection accuracy. Quantum computing has the potential to provide significantly higher computational power compared to classical computing. QCNNs could leverage this power to process and analyze large-scale agricultural data, such as high resolution imagery or genetic data, more efficiently. This could lead to faster and more accurate detection of plant diseases and better decision-making in agricultural mechanization.

In agriculture, computer vision has gained significant attention and shown great promise in various applications. CNNs can analyze plants and crops to identify

disease symptoms accurately. By training CNN models on large datasets of labeled images, they can learn to recognize specific visual patterns associated with different diseases [3–5]. This enables automated and rapid disease detection, allowing farmers to take timely action to manage and control diseases effectively.

CNNs can be used to distinguish between crops and weeds in images, facilitating targeted weed management strategies. By analyzing field images, CNN models can identify and differentiate weed species from crops, enabling precision herbicide application or mechanical weed removal techniques [6, 7]. This helps reduce the use of herbicides and increase overall crop yield [8, 9].

Computer vision techniques, combined with machine learning algorithms, can analyze aerial or satellite images to estimate crop yield and monitor crop growth [10]. By extracting features such as vegetation indices, plant density, or canopy cover from images, CNN models can provide insights into crop health, growth stage, and potential yield. This information assists farmers in making informed decisions regarding irrigation, fertilization, and harvesting.

The application of CNNs and computer vision methods in agriculture is continuously evolving and has the potential to revolutionize farming practices. By leveraging the power of deep learning and image analysis, these technologies contribute to improved crop management, increased productivity, and more sustainable agricultural systems.

This study focuses on utilizing a quantum machine learning technique called QCNN to recognize plant infections. This approach combines the power of quantum computing and convolutional neural networks to address the challenge of disease detection in plants. We collected a diverse dataset called "Plant Village" [11], consisting of images and other relevant data related to different plant diseases. This dataset serves as the training data for the QCNN model.

QCNNs play a crucial role in agricultural mechanization and plant disease detection, improving accuracy and efficiency. This approach expands the horizons of quantum machine learning and showcases the versatility of quantum technologies beyond traditional domains. By addressing challenges in plant disease detection, QCNNs pave the way for future advancements in hardware, software, and algorithms, ultimately revolutionizing agricultural practices and enhancing productivity.

The study aims to develop a QCNN model specifically tailored for plant disease detection. QCNNs utilize quantum computing principles to enhance the computational power and potential accuracy of CNNs. The model will be designed to process the training dataset and learn the complex patterns and features associated with different plant diseases.

The rest of this paper is organized as follows: Sect. 2 provides a review of relevant research that focuses on segmentation and detection techniques for plant diseases. It would present previous studies, methodologies, and findings related to the topic, highlighting the existing research landscape and advancements in the field. Section 3 provides methodologies used for segmentation and detection of plant diseases. It would detail the specific techniques, algorithms, or approaches employed, such as image processing methods, machine learning models, or other relevant methodologies utilized for disease segmentation and detection in plants. Section 4 includes

quantitative metrics, visual representations, or comparisons with ground truth data or existing approaches. Section 5 provides a concise summary of the paper's objectives, methodologies, results, and overall importance in advancing plant disease management.

2 Related Work

In this work, the most recent and pertinent research was chosen for this investigation using a systematic review methodology. Many papers were published in international journals and conferences. Researchers are working on hierarchical networks, neural networks, and device domain methods. Some classification methods that researchers consistently use are support vector machines (SVM), neural networks, traditional regression pairs, k-nearest neighbors, and genetic algorithms. Babu et al. proposed a backpropagation neural network for leaf recognition [12, 13]. Zhang et al. worked on cucumber disease detection through sparse representation classification. This work focuses on leaf images, where the photograph is segmented into small segments using k-means clustering [14].

The disease classification is performed with the aid of sparse matrices. The results of the proposed approach demonstrate that it reduces computation time and improves performance, and the results of this method are comparable to alternative techniques [15]. Another proposed approach for leaf disease detection utilizes the concepts of image segmentation and soft computing strategies. This approach automatically classifies and detects diseases in leaves. Surveys of various diseases related to plants are also used for detection. This work is done with the assistance of a genetic algorithm, which efficiently optimizes the variables. R-CNN mask is one of the methods [16]. This method won the 2016 COCO competition.

The R-CNN mask extends the Faster R-CNN network architecture and classifies each object in the image by adding a mask. The R-CNN mask can be used to find the target in pixels. This is an instance-based segmentation model. Instance segmentation refers to classifying every part of the photograph. Bhangea [17] developed a new method, which is a network-based problem-solving tool. This method adjusts the size of the first image uploaded to the Web portal and extracts image features, including shadows, shapes, etc. The author uses K-Tools to group and classify SVMs [18]. Many traditional AI techniques are used to detect diseases, including support vector machines (SVM) [19, 20], and K-Tools [21], and decision trees. Vision and object detection, pest detection, and the use of deep learning techniques have become active research topics in recent years. The Normalized Difference Vegetation Index, a strategy for generating heterogeneous data from NDVI, was suggested [22].

In many of the articles on image-based plant disease identification, the background is often removed, and the lesion tissue of the diseased plant is segmented. Discriminative features are extracted to further evaluate plant illness, and a model is developed for disease classification. Specific threshold values are used as suggested by Al-Hiary et al. [23, 24].

3 Materials and Methods

This section provides a detailed description of the materials and methodologies employed in the development of a quantum convolutional neural network for agricultural mechanization and plant disease detection. The methodology used in the study is presented in the "Segmentation" section below.

3.1 Dataset

The dataset was collected as a diverse collection of images and other relevant data pertaining to various plant diseases. Most of the datasets used in this work are publicly available datasets for plant disease detection. For this purpose, we collected some datasets from the Plant Village dataset, which contains over 54,000 images of healthy and diseased plant leaves, covering 14 different crop species and 26 diseases. Additionally, some datasets were obtained from the Plant Pathology 2020 dataset, which consists of over 18,000 images of apple tree leaves with labels for multiple diseases and health conditions. A portion of the data was also taken from the Citrus Fruits dataset, which includes over 300 images covering diseases such as citrus canker, citrus greening, and citrus scab. Table 1 displays the dataset used for training, which includes the total number of samples. Figure 1 shows the sample Plant Village dataset images.

This dataset was used as the training data for the QCNN model. We utilized more than 4000 datasets for training the model, and out of these, 20% were reserved for evaluating the effectiveness of the model. Publicly available datasets were organized and manually labeled based on image quality.

In addition, other datasets such as the Leafsnap dataset [13], the Cassava Leaf Disease [22] dataset, the Merged dataset, and the Tomato Disease dataset [20] were

Table 1 Dataset used for training

Disease	Crop	Total sample used
Leaf spot	Tomato	650
Powdery mildew	Grapes	300
Rust	Wheat	400
Bacterial blight	Rice	350
Anthracnose	Pepper	500
Healthy	Apple	300
Leaf curl	Tomato	500
Black rot	Citrus	500
Early blight	Potato	200
Fusarium wilt	Banana	300

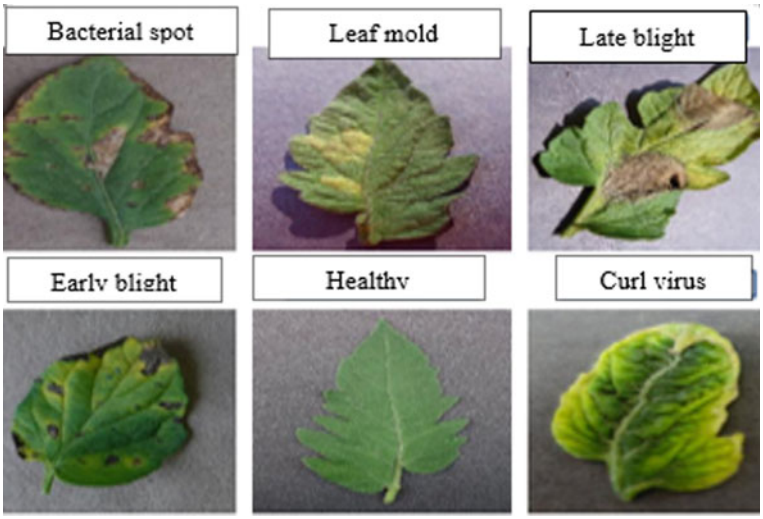


Fig. 1 Plant village dataset [13]

also utilized. The model is specifically tailored for plant disease detection. QCNNs utilize quantum computing principles to enhance the computational power and potential accuracy of convolutional neural networks. The model would be designed to process the training dataset and learn the complex patterns and features associated with different plant diseases.

3.2 Feature Extraction

To obtain representative features from the input data in the plant disease detection dataset, the process starts by resizing the images and normalizing pixel values. Then, the Quantum Circuit Born Machine (QCBM) model is applied to extract features. QCBM is used to generate samples from a probability distribution to obtain optimal parameters. To reduce the cost function, quantum gradient descent is applied using the equation:

$$\Delta\alpha = -\eta * G^{-1} * m \tag{1}$$

Here, $\Delta\alpha$ represents the change in parameters, η is the learning rate, G^{-1} is the inverse information matrix, which quantifies the curvature of the cost function surface, and m is the gradient of the cost function.

Then, the model was trained based on these labeled images. Principal component analysis (PCA) was applied to extract representative features from the input data. Through an iterative process, the model's parameters would be adjusted to minimize

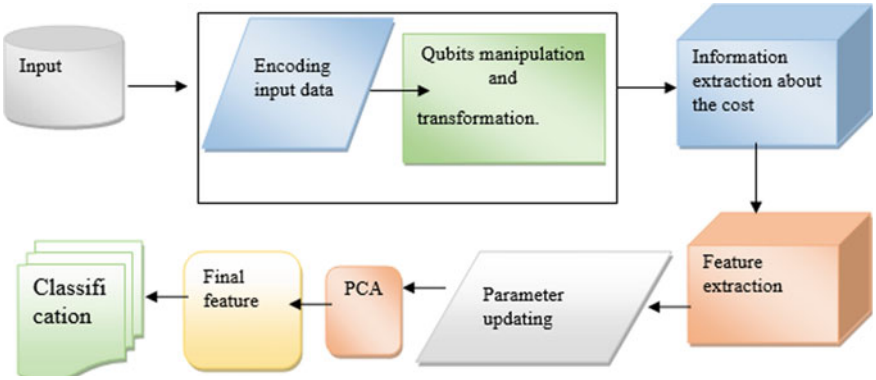


Fig. 2 Feature extraction procedure

the detection errors and improve its performance. Figure 2 illustrates the feature extraction procedure.

3.3 Segmentation

In this work, region-based segmentation is employed to group the color and texture of the leaf. This approach allows the pixels to be grouped into regions based on the healthiness of the leaf. To achieve this, the similarity function is applied to compare the similarity between the leaf's color and texture using cosine similarity.

Image segmentation is performed using the MobileNetV2 model, and classification is performed using the QCNN. The flow of leaf diseases detection typically involves several steps, which can vary depending on the specific approach used. Figure 3 indicates flow of leaf disease detection by using a QCNN.

The first step involves collecting a dataset of leaf images that represent a variety of healthy and diseased plants. The collected leaf images then undergo preprocessing steps to enhance their quality and remove noise. This may involve resizing, normalization, filtering, or other necessary image processing techniques as discussed earlier.

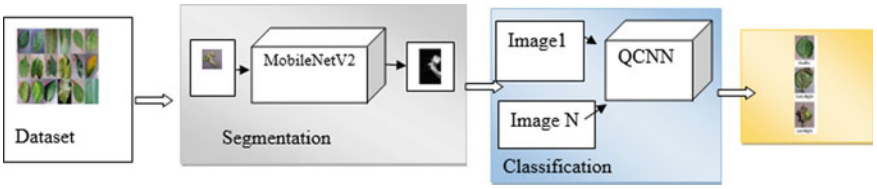


Fig. 3 Flow of proposed model

Table 2 Parameters of MobileNetV2 model

Parameters	Layers	Total parameters
Trainable parameters	53	409,225
Non-trainable parameters	53	7184

Next, features are extracted from the preprocessed leaf images to capture relevant information for disease detection. This step involves analyzing the shape, texture, color, or other characteristics of the leaves. Texture analysis was specifically utilized for feature extraction in the study.

The extracted features are utilized to train a classification model, which can be a machine learning algorithm or a deep learning model. The model learns to distinguish between healthy and diseased leaves based on the extracted features. The training data is labeled with the corresponding disease classes. In the approach, a pre-trained MobileNetV2 model was utilized for segmentation, which underwent fine tuning for the specific task.

The labeled training data is then used to train the classification model. This process involves optimizing the model's parameters to minimize classification errors and improve its performance. The training process typically includes iterations over the training data, adjusting the model's weights or parameters using techniques like gradient descent or backpropagation.

After training the model, a separate set of labeled test data, which was not used during the training phase, is employed to evaluate the performance of the trained model. The evaluation is based on various criteria such as accuracy, precision, recall, or F1-score. This evaluation process assesses the model's ability to accurately detect and classify leaf diseases.

The following Table 2 is a representation of our MobileNetV2 model definition, along with information on the total parameters used, trainable parameters, and none trainable parameters.

During the training and testing of our image segmentation model, we selected a relatively low number of epochs, specifically 20. Despite this limited number of iterations, the model achieved satisfactory results with high accuracy. The obtained results from the image segmentation model are saved and subsequently utilized in our next model, QCNN. The QCNN model is trained using the saved results from the image segmentation model, enabling it to learn the complex patterns and features associated with different leaf diseases.

3.4 Classification

Plant diseases classification involves categorizing different types of diseases or disorders that affect plants based on their visual symptoms or characteristics. The QCNN architecture is used as the training model. The input data is preprocessed classically, encoded into a quantum format, processed through a parameterized quantum circuit,

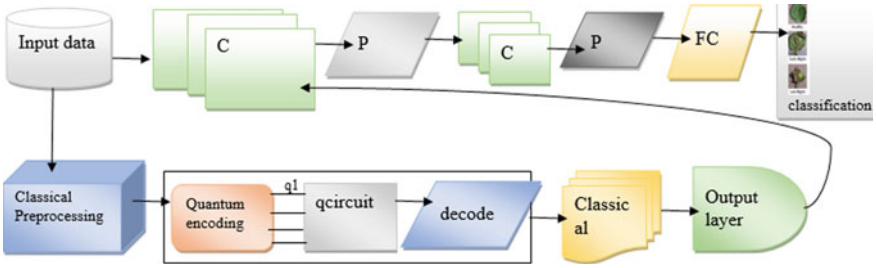


Fig. 4 QCNN training model setup

Table 3 Parameters used in the model

Number of Q bits	04
Learning rate	0.0004
Batch size	0.04
Number of V layers	0.6
Q weight	0.01
Max epochs	20

post-processed classically, and finally passed through a convolutional layer to reduce the dimensions of features using a pooling layer. Information from previous layers is integrated using a fully connected network and mapped for classification purposes. Once the model is trained and evaluated, it can be used for disease detection on unseen leaf samples. The model predicts the diseases using attributes extracted from the new leaf images as input. The result could be in the form of disease labels, probabilities, or severity scores. Additional post-processing was also applied to the detection results, such as thresholding, filtering, and noise removal, to refine the output. In the quantum circuit shown in Fig. 4, the chosen configuration involved six convolutional layers and a batch size of 4, indicating the utilization of four samples for each training step.

The QCNN network was trained with 4 quantum bits, initialized with random quantum weights of 0.1. A learning rate of 0.0004 per step was used, and a reduction in the learning rate was applied after every 10 epochs with a gamma_lr_scheduler value of 0.1. The parameters used in the model are listed in Table 3.

4 Results and Discussion

It is important to note that the specific techniques, algorithms, or models used in plant diseases classification may vary depending on the available data, the complexity of the problem, and the desired accuracy. Additionally, advancements in deep learning, such as the use of CNNs, have shown promising results in plant disease classification tasks,

providing high accuracy and the potential for automated detection and monitoring. Table 4 presents an accomplishment of the segmentation model. In which accuracy, loss, validation, precision, recall, and dice-coef of the model are mentioned.

With the batch size set to 4 in the parameters section, four images are plotted in the output section, indicating that the number of images displayed corresponds to the batch size [num_images = batch size]. Table 5 presents the performance evaluation of the classification model. The evaluation measures used to assess the effectiveness of the model include metrics such as accuracy, precision, recall, and F1-score. The result underlines that performance metrics with those reported in previous studies or benchmark datasets are more accurate.

4.1 State of the Art

In comparison with previous works, the proposed model has achieved significant and fruitful results. By employing quantum convolutional neural networks (QCNNs) on publicly available data, the proposed approach can accurately classify the class of a leaf disease. Table 6 showcases the performance of the proposed work in comparison with previous studies, demonstrating superior accuracy in plant disease detection with a score of 0.966. These results emphasize the potential effectiveness and reliability of the proposed methodology using QCNNs for plant disease detection.

5 Conclusion

The main issues faced while detecting leaf blight disease in the past were the classification of leaf blight disease according to its early and late blight classes. It is hard to classify the blight disease and its stages. Although the work in this field is enormous, this work addressed classification issues by using quantum neural network, which is new in this field. We have achieved high accuracy in our classification, which is good for future work in the disease detection field. Results show that our proposed system results in good, segmented images and high accuracy in the classification of those segmented images. In the future, the objective is to develop software that enables fully automated image processing, encompassing tasks ranging from segmentation to classification.

Table 4 Performance evaluation of segmentation

Loss%	Precision %	Recall%	Dice-coef %	Val-loss %	Val-recall %	Val-precision %	Val-accuracy %	Accuracy %
0.04	0.93	0.77	0.76	0.4	0.10	0.91	0.92	0.96

Table 5 Performance evaluation result

Dataset	Accuracy	Loss
Plant Village	94.17%	5.83%

Table 6 Comparison with state of the art

Model	Performance	Loss
QCNNs	0.966	0.034
CML	0.910	0.09
CNN	0.932	0.068
SVM	0.891	0.109
KNN	0.912	0.088

References

1. Varsha VS, Misra AK (2015) Detection of unhealthy region of plant leaves using image processing and genetic algorithm. In: International conference on advances in computer engineering and applications (ICACEA), IMS Engineering College, Ghaziabad, India, pp 1028–1032
2. Amara J, Bouaziz B, Algergawy A et al (2017) A deep learning-based approach for banana leaf diseases classification. In: BTW (Workshops), pp 79–88
3. Abade A, Ferreira PA, de Barros Vidal F (2021) Plant diseases recognition on images using convolutional neural networks: a systematic review. Comput Electron Agric 185
4. Dhaka VS, Meena SV, Rani G, Sinwar D, Ijaz MF, Woźniak M (2021) A survey of deep convolutional neural networks applied for prediction of plant leaf diseases. Sensors 21:4749
5. Nagaraju M, Chawla P (2020) Systematic review of deep learning techniques in plant disease detection. Int J Syst Assur Eng Manag 547–560
6. Lu J, Tan L, Jiang H (2021) Review on convolutional neural network (CNN) applied to plant leaf disease classification. Agriculture 11:707
7. Bangari S, Rachana P, Gupta N, Sudi PS, Baniya KK (2022) A survey on disease detection of a potato leaf using CNN. In: Proceedings of the 2nd IEEE international conference on artificial intelligence and smart energy (ICAIS), Coimbatore, India, vol 23, pp 144–149
8. Naik BN, Ramanathan M, Palanisamy P (2022) Detection and classification of chilli leaf disease using a squeeze-and-excitation-based CNN model. Ecol Inform 69:101663
9. Sahu P, Chug A, Singh AP, Singh D, Singh RP (2021) Deep learning models for beans crop diseases: classification and visualization techniques. Int J Mod Agric 10:796–812
10. Santosh A et al (2018) Tomato plant diseases detection system. Plant village.org web site visited date august (2020)
11. Mukti IZ, Biswas D (2019) Transfer learning based plant diseases detection using ResNet50. In: Proceedings of the 4th IEEE international conference on electrical information and communication technology (EICT), Khulna, Bangladesh, 20–22 Dec, pp 1–6
12. Arya S, Singh R (2019) A comparative study of CNN and AlexNet for detection of disease in potato and mango leaf. In: Proceedings of the IEEE international conference on issues and challenges in intelligent computing techniques (ICICT), Ghaziabad, India, 27–28 Sept, vol 1, pp 1–6
13. Ganatra N, Patel A (2018) A survey on diseases detection and classification of agriculture products using image processing and machine learning. Int J Comput Appl 180(13)
14. Zhang S, Zhang L, Wu X, You Z (2018) Leaf image-based cucumber disease recognition using sparse representation classification, vol 18
15. Sing V, Misra AK, He K, Gkioxari G, Dollar P, Girshick R (2017) Detection of plant leaf diseases using image segmentation and soft computing techniques. 59:100–112

16. Bhangea M (2015) Smart farming: pomegranate disease detection using image processing. 58:280–288. 12
17. Abayomi-Alli OO, Damaševičius R, Misra S, Maskeliūnas R (2021) Cassava disease recognition from low-quality images using enhanced data augmentation model and deep learning. Exp Syst 38:e12746
18. Mokhtar U, Ali MA, Hassanien AE, Hefny H (2015) Identifying two of tomatoes leaf viruses using support vector machine. In: Mandal JK, Satapathy SC, Sanyal MK, Sarkar PP, Mukhopadhyay A (eds) Information systems design and intelligent applications. Springer, pp 771–782
19. Arivazhagan S et al (2015) Detection of unhealthy region of plant leaves and classification of plant leaf diseases using texture features. Agri Eng Inter CIGRJ 15:211–217
20. Shrestha G, Deepsikha, Das M, Dey N (2020) Plant disease detection using CNN. In: 2020 IEEE applied signal processing conference (ASPCON), pp 109–113
21. (2018) Image set for deep learning: field images of maize annotated with disease symptoms. BMC Res Notes 11
22. (2019) Three-channel convolutional neural networks for vegetable leaf disease recognition. Cogn Syst Res 53:31–41
23. Atila M, Uar M, Akyol K, Uar E (2021) Plant leaf disease classification using EfficientNet deep learning model. Ecol Inform 61:101182
24. Safari B, Alborzi Y, Najafi E (2022) Automated wheat disease detection using a ROS-based autonomous guided UAV. arXiv preprint arXiv:2206.15042

Innovative Method for Alzheimer's Disease Detection Using Convolutional Neural Networks



Md. Sajid Anam Ifti, Md. Redwan Ahmed, S. M. Arafat Rahman, Sheikh Shemanto Afridi, Sanjeda Sara Jennifer, and Ahmed Wasif Reza

Abstract Deep learning (DL) approaches have gained significant traction in recent years for medical image analysis. In this research, magnetic resonance imaging (MRI) data were employed to classify Alzheimer's disease (AD) using DL techniques. Alzheimer's disease (AD), a fatal neurological condition, is characterized by early signs such as memory loss and cognitive decline. Accurate and timely diagnosis is crucial for appropriate patient treatment and subsequent care. Unlike traditional feature learning methods like machine learning, deep learning approaches can directly learn complex and high-level features from datasets. However, this research extends the classification procedure by incorporating forecasting for additional disease stages. The proposed approach obtained an impressive accuracy of 98.02% using the XCeption model. The automatic extraction of intricate features from MRI data empowers improved diagnostics, proactive disease management, and personalized treatment strategies for individuals affected by Alzheimer's disease.

Keywords Alzheimer's disease · Deep learning · XCeption · Convolutional neural network · Magnetic resonance imaging

1 Introduction

Alzheimer's disease (AD), an autoimmune neurological disorder, is spurred on by nerve tissue degradation in certain brain regions. The first indication of cognitive impairment in a patient is their incapacity to do crucial operations due to memory loss,

Md. Sajid Anam Ifti · Md. Redwan Ahmed · S. M. Arafat Rahman · S. S. Afridi · S. S. Jennifer · A. W. Reza (✉)
Department of Computer Science and Engineering, East West University, Dhaka 1212, Bangladesh
e-mail: wasif@ewubd.edu

Md. Sajid Anam Ifti
e-mail: info@sajidifti.com

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2023

S. Shakya et al. (eds.), *Fourth International Conference on Image Processing and Capsule Networks*, Lecture Notes in Networks and Systems 798,
https://doi.org/10.1007/978-981-99-7093-3_16

speech impediments, and other cognitive problems [1]. The World Health Organization (WHO) ranks it the sixth most prevalent global cause of adult death. According to the Alzheimer's Association, the sixth most prevalent factor leading to the end of a person's life in the USA is due to AD [2]. Logic, attention, and memory are all affected when the patient's memory area of the brain shrinks and matures. This is the main reason for this disease. Genetic mutations, which affect 1% or fewer people, are suggested to be another possible cause of AD [3]. To identify AD and aid in diagnostics, feature wavelets were traditionally constructed using the discrete wavelet transform (DWT) technique. Additional investigation is required [4] using machine learning techniques since it is difficult to identify injuries. Regardless of elements, representatives, or populations, cooperative approaches have a direct influence on most optimization strategies. In the initial rounds, the genetic algorithm (GA), particle swarm optimization (PSO), and Tsallis Kapur's entropy were the optimization techniques that were employed [5]. Following that, the bat, bees, and cuckoo search (CS) metaheuristic algorithms were recommended [6]. While establishing a conclusive assessment of AD presents challenges, in practical terms, the diagnosis primarily relies on the assessment of clinical history and neuropsychological information, including the utilization of magnetic resonance imaging (MRI) [7]. The gray wolf optimization (GWO) strategy, which is focused on leadership attributes in method identification, was also promoted to enhance global search [8]. Part of the reason this algorithm is getting so much attention is due to its aggressive discoveries, its increased capacity for problem-solving, and its ability to offer an optimum solution superior to previous optimization techniques [9, 10].

In deep learning problems, the custom convolutional neural network (CNN) excels remarkably. The outcomes achieved in image-based applications, such as the remarkable progress in the largest image classification dataset (ImageNet), as well as advancements in natural language processing (NLP), and computer vision have been truly impressive [11]. Using custom CNN in conjunction with existing CNN models was shown to produce the best outcomes in this study.

The rest of the paper is organized in the following manner: Sect. 2 presents the related work, highlighting previous studies and relevant work. The materials used and the methods employed are outlined in Sect. 3. Section 4 exhibits the results obtained from the study and provides in-depth discussions and interpretations of the findings. Section 5 serves as the conclusion of the paper, presenting a concise overview of the primary findings, examining their significance, and proposing potential paths for future investigation.

2 Related Work

Deep learning was used to assess Alzheimer's disease using brain MRI data in [1]. The paper introduces CNN and Google Net as methods for analyzing the OASIS dataset, emphasizing their effectiveness in dealing with small medical images and

emphasizing the need for further investigation of unsupervised and semi-supervised procedures as potential research gaps or limitations.

Learning algorithms have been tested to detect Psoriatic arthritis using brain MRI scans from both normal and those with Alzheimer's disease. Deeper learning methods could be used to develop an effective device for advanced Alzheimer's pathogen detection [2].

Deep learning algorithms can be used to automatically generate Alzheimer's diagnoses based on MRI data, which could help improve premature detection and diagnosis, leading to more effective therapy options. The study article employs CNN as a technique and the ADNI dataset to provide a considerable contribution to the accuracy of AD identification, with a specificity of 0.93. However, the report notes the need for more research to improve CNN's effectiveness in this setting [8].

Deep learning techniques are used to diagnose early Alzheimer's disease, which could lead to faster and more economical remedies. ConvNet is used as a method in the study article, and the ADNI dataset is used to illustrate its superiority over conventional approaches, providing a substantial contribution. The study, however, emphasizes the need for more research to address existing research gaps and limitations [9].

Deep learning algorithms can be used to isolate people with early-stage Alzheimer's disease from those without it with extreme accuracy, potentially leading to more efficient remedies. The research study employs the DCssCDBM technique using the easy monitor technology dataset to outperform various sophisticated models. However, the report underlines the need of investigating DCssCDBM's potential to overcome existing research gaps and constraints [10].

Deep learning (DL) algorithms can be used to analyze data, but there are still obstacles to be addressed. However, a useful tool for preventing and analyzing AD. The study employs machine learning (ML) and DL approaches, as well as the Neuroimaging Informatics Technology Initiative (NIFTI) and Open Access Series of Imaging Studies (OASIS) datasets, to provide a significant contribution through biomarker and neuroimaging analyses. However, the report admits the presence of congenital limits and underlines them as research gaps and limitations [12].

A DL system was developed to utilize fluorine 18 fluorodeoxyglucose PET brain scans for predicting the eventual diagnosis of AD. The algorithm beat radiologic readers in the ROC domain, with an average of 75.8 months before the definitive diagnosis. The program outperformed radiologic readers with 82% specificity and 100% sensitivity. The saliency map exhibited a notable emphasis on recognized regions of interest by directing attention to the entirety of the brain [13].

The study addresses Alzheimer's disease detection by segmenting brain subregions with optimization methods, reaching 98% accuracy with deep learning classifiers and 95% accuracy with conventional classifiers. The hippocampus is vital, exceeding other areas in terms of accuracy, sensitivity, and specificity [5].

Early detection of AD is crucial for treatment; machine learning methods face challenges like class imbalance and data leakage [14]. This paper reviews conventional ML methods for classifying and predicting AD using MRI, discussing challenges like class imbalance and data leakage [15]. The study's goal is to use deep

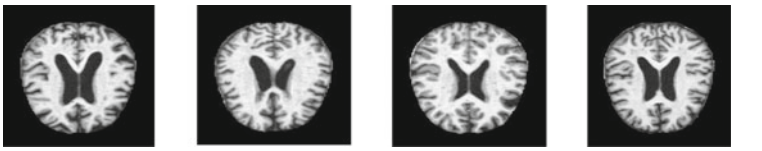


Fig. 1 Sample images from each class

learning and MRI to identify AD and moderate cognitive impairment. It assesses CNN models, adds data augmentation, and creates visually understandable heatmaps. The method outperforms previous methods by emphasizing damaged areas such as the lateral ventricle and brain [16].

3 Materials and Methods

3.1 Dataset Description

This study employed a dataset from Kaggle [17], which included precisely gathered MRI pictures from several sources with confirmed annotations. The collection contains 6400 pictures, each of which depicts a different stage or feature of Alzheimer's disease (AD), such as mild demented, moderate demented, non-demented, and very mild demented. The major goal of publishing this information is to aid in the creation of a very accurate algorithm for predicting the stage of Alzheimer's disease. Figure 1 depicts a graphic representation of an exemplary image from each category.

3.2 Dataset Preprocessing

In order to prepare the dataset for further processing, the images need to go through a series of essential steps, including scaling, normalizing pixel values, and dividing the data into two sets. The images were scaled to consistent dimensions of 64 $\times$ 64 and 71 $\times$ 71 pixels, ensuring uniformity in input size across the dataset. This scaling process establishes a standardized framework for subsequent analysis. Furthermore, to bring the intensity values within a desirable range of 0–1 pixel, value normalization was employed. By normalizing the pixel values, the data is rescaled to a consistent scale, facilitating effective comparison and analysis across images. This normalization step plays a crucial role in ensuring that the model can learn meaningful patterns from the data. The provided code uses the rectified linear unit (ReLU) activation function in the convolutional layers of the model for Alzheimer's disease classification. ReLU

Table 1 Image scaling size of different models

S. No.	Model name	Size
01	Custom CNN	64 × 64
02	ResNet-50	64 × 64
03	XCeption	71 × 71

introduces nonlinearity, enabling the model to learn complex patterns and features from the input images. By setting negative values to zero, ReLU enhances the model's ability to capture intricate relationships and extract relevant features. Applying ReLU in the convolutional layers helps the model learn and extract meaningful features, leading to improved performance in AD classification.

To evaluate the performance of the model and assess its generalizability, the dataset was divided into two sets: a train and a test set. The 'train_test_split()' function from the 'sklearn.model_selection' module was utilized, assigning 80% of the data for training purposes and utilizing the remaining 20% for testing. This partitioning allows for the evaluation of the model's performance on previously unseen data, providing a measure of its efficacy in real-world scenarios. Moreover, the input data was reshaped to include the batch size dimension, which is a fundamental requirement for feeding data into convolutional neural network (CNN) models. Reshaping the data ensures compatibility with the expected input format of the CNN, specified as '(batch_size, height, width, channels)'. In this case, the channel dimension is set to 1, reflecting the grayscale nature of the images. By diligently performing these preprocessing steps, the data is appropriately prepared, ensuring consistency, facilitating effective analysis, and enabling the subsequent training and evaluation of the deep learning models. Table 1 summarizes the transformations applied to the images for each respective model, encapsulating the alterations undertaken in the pre-processing pipeline.

3.3 Model Architecture and Design

This section presents the overall workflow of the system architecture and the models that were chosen to be employed for this research. The effectiveness of various CNN designs in identifying AD is evaluated in this research. There are 50 epochs per fold, and a total of five-fold cross-validation was employed for training the dataset. A deep learning framework, TensorFlow, was employed for model training and evaluation, and performance evaluation was conducted using metrics such as accuracy, precision, recall, and $F1$-score. Custom CNN had the best accuracy and training time. Figure 2 shows the entire workflow of the proposed planned methodology involving every step, followed by the algorithm description.

This research decided to experiment with custom CNN, ResNet-50 architecture, and Xception architecture. ResNet-50 and Xception are two prominent architectures that are widely used in medical image analysis due to their superior performance and

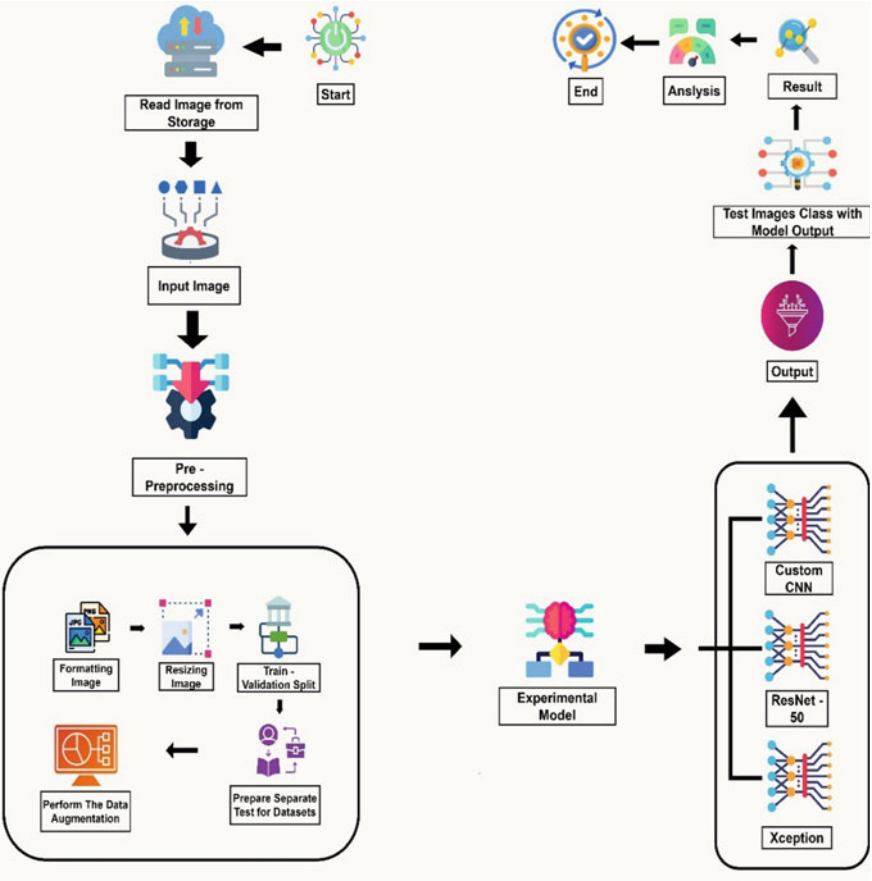


Fig. 2 Hierarchical overview of the proposed methodology

accessibility to pre-trained models. The computational complexity is low due to the depthwise separable convolutions and residual connections.

Custom CNN: The Keras framework was used to create the first CNN, which was trained for 50 iterations per fold using training data. Figure 3 shows the custom network of the CNN model, prepared for implementation.

ResNet-50: The CNN model was created using pre-trained ResNet-50, global average pooling, dropout regularization, accuracy metric, and Adam optimizer. Figure 4 represents the ResNet50 architecture.

XCeption: We created a CNN model based on the pre-trained XCeption model with global average pooling and dropout regularization. Figure 5 illustrates the proposed custom XCeption model.

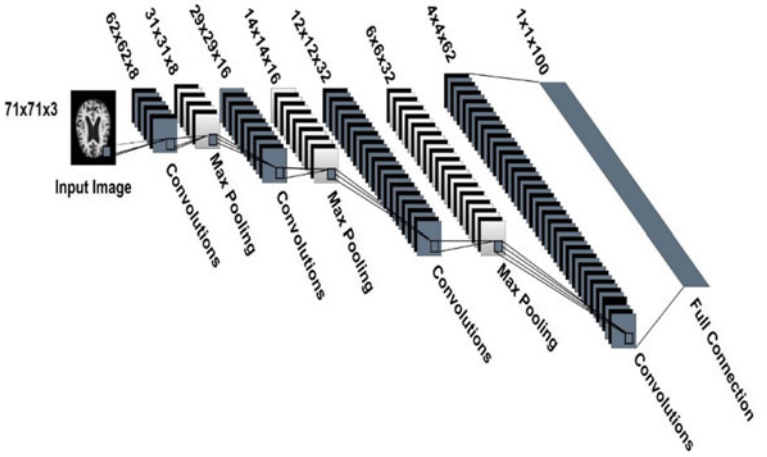


Fig. 3 Custom CNN model

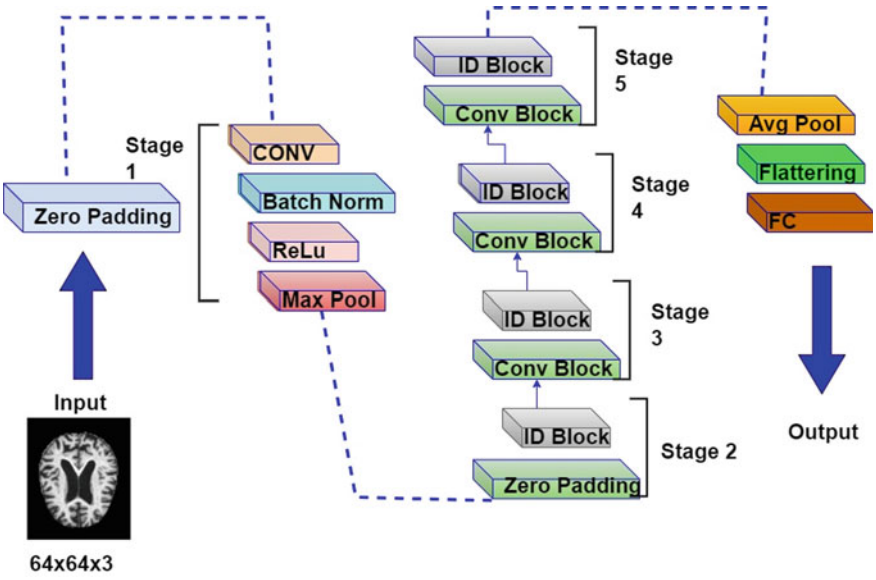


Fig. 4 ResNet50 model

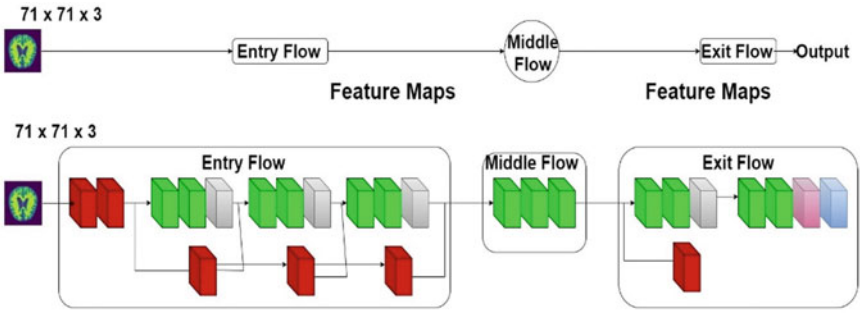


Fig. 5 Proposed custom XCeption model

4 Result Analysis and Discussion

4.1 Experimental Setup

The proposed system was built on Jupyter Notebook, a computing platform with an 8-core AMD CPU, 1TB of storage, 16 GB of RAM, and AMD GPU RX Vega 11. Models were built using TensorFlow and Keras deep learning frameworks.

4.2 Result Analysis and Performance Evaluation

A system using 6400 Alzheimer pictures was developed utilizing custom CNN, ResNet-50, and XCeption. Except for XCeption, which used 71 × 71 input vectors, other models used 64 × 64 input vectors. Each model was trained for 50 training epochs before being tested for accuracy and loss. In order to evaluate the performance of our model, precision, recall, $F1$-score, and accuracy were chosen as the evaluation matrices [18]. Their mathematical formulation is presented in Eqs. (1), (2), (3), and (4), and Table 2 gives the performance analysis:

$$\text{Precision} = \frac{\text{true positive}}{(\text{true positive} + \text{false positive})} \tag{1}$$

$$\text{Recall} = \frac{\text{true positive}}{(\text{true positive} + \text{false negative})} \tag{2}$$

$$F1\ \text{Score} = \frac{2 \times (\text{precision} \times \text{recall})}{(\text{presion} + \text{recall})} \tag{3}$$

$$\text{Accuracy} = \frac{\text{true positive} + \text{true negative}}{\text{true positive} + \text{true negative} + \text{false positive} + \text{false negative}} \tag{4}$$

Apart from the precision, recall, $F1$-score, and accuracy, confusion matrices that use real labels and predicted values are also used to evaluate the model, as shown in Figs. 6, 7 and 8. Five-fold cross-validation was performed on the dataset by splitting it into five equal folds. The model is trained using four folds, with the final fold being utilized for validation. Once for each fold, the validation set is used in this procedure five times. To produce an overall performance statistic, the iteration outcomes are averaged. This method aids in determining how well the three models can generalize and identify overfitting or underfitting. In making decisions about model selection and optimization, it offers a trustworthy approximation of the model's performance. A graphical representation of the models after performing five-fold cross-validation is illustrated in Figs. 9, 10 and 11.

Figure 6 illustrates the performance of a custom CNN model in classifying the severity of dementia. With an overall accuracy of 94.4%, it faces challenges in accurately categorizing instances within the ModerateDemented and NonDemented categories. This arises due to the higher similarity in symptoms exhibited by these two categories compared to others.

The dementia diagnosis model depicted in Fig. 7 achieved an overall accuracy of 96%. However, it encountered challenges in accurately classifying instances as ModerateDemented and VeryMildDemented, with accuracies of 75% and 68.8%,

Table 2 Performance analysis of the architectures

Model	Precision (%)	Recall (%)	Avg. $F1$-score (%)	Test loss	Test accuracy (%)
Custom CNN	94	94	94	0.0767	94
ResNet-50	96	96	96	0.0132	96
XCeption	98	98	98	0.0178	98

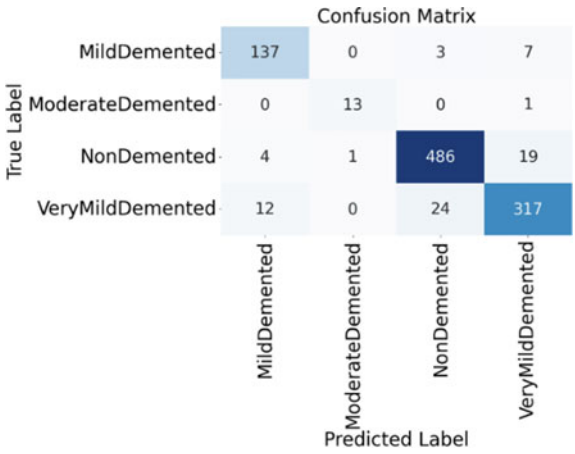


Fig. 6 Confusion matrix of the custom CNN model

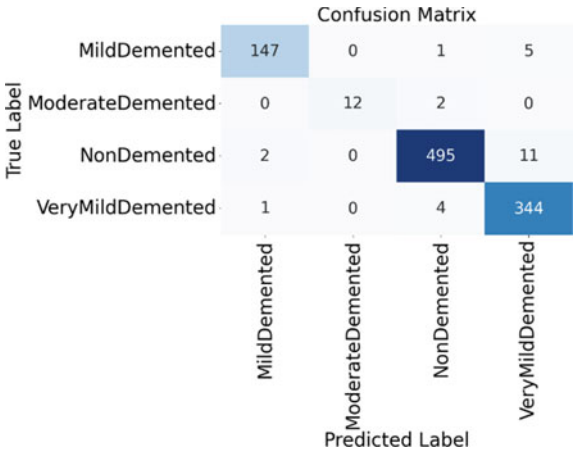


Fig. 7 Confusion matrix of the ResNet-50 model

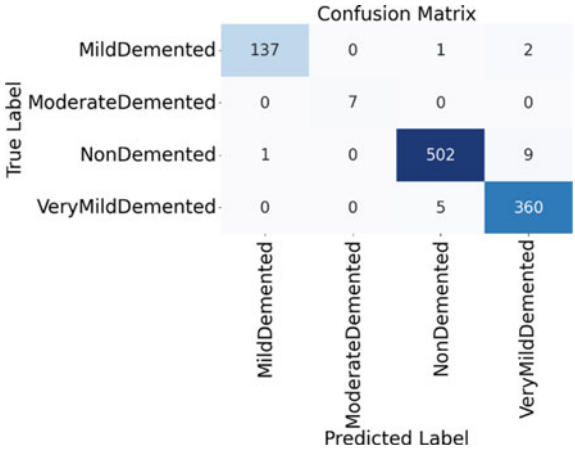


Fig. 8 Confusion matrix of the XCeption model

respectively. The model exhibited a tendency to misclassify ModerateDemented instances as MildDemented, likely due to the similarities between these categories.

The XCeption model displayed a 98% overall classification accuracy in the confusion matrix. However, it struggled to accurately classify instances as ModerateDemented and VeryMildDemented. Notably, it misclassified a significant number of ModerateDemented instances as MildDemented and a significant number of VeryMildDemented instances as NonDemented. This challenge may arise from the similarities between these categories.

Figure 9 shows the accuracy and loss curves of a machine learning model trained on five folds of data. The accuracy curve for all five folds is above 90%, with the highest accuracy being 98% for fold 5. The loss curves for all five folds are below 10%, with the lowest loss being 4% for fold 5. The accuracy curves plateau after epoch 30, while the loss curves continue to decrease. This indicates that the model

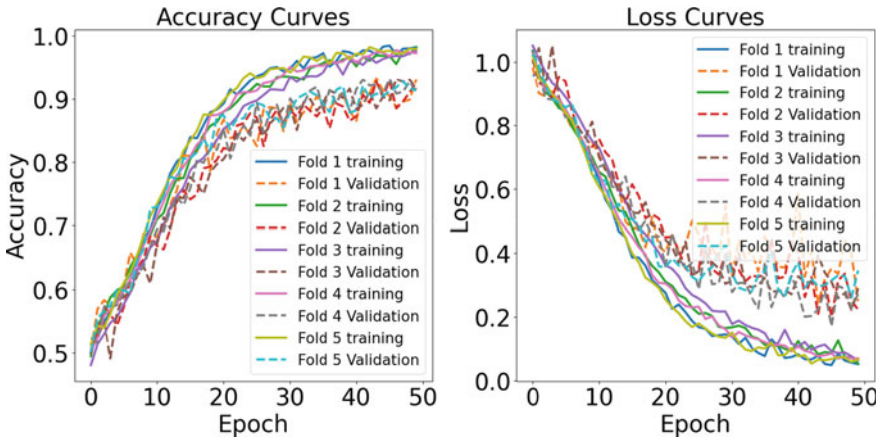


Fig. 9 Custom CNN accuracy and loss

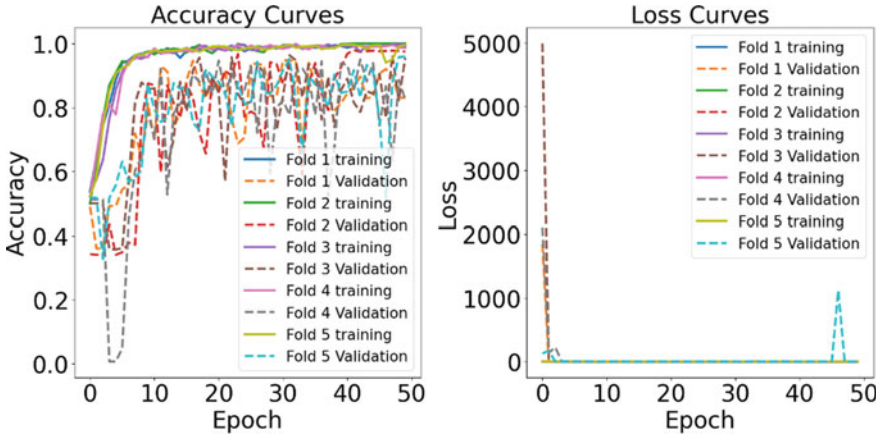


Fig. 10 ResNet-50 accuracy and loss

can achieve high accuracy with low loss. However, the fact that the accuracy plateaus after epoch 30 suggest that the model is likely overfitting the training data.

The accuracy curves of ResNet-50 on the training set and the validation set are converging, but the validation curve is not as steep as the training curve, as shown in Fig. 10, which indicated an overfitting issue on the training set. The loss curves of the model on the training and validation set are converging, but the validation curve is not as low as the training curve. This suggests that the model is not generalizing as well as it could.

Figure 11 shows that, after 10 epochs, the accuracy of the model on the training and the validation set reaches 90% and 85%, respectively. After 10 epochs, the loss of the model on the training set decreases from 0.5 to 0.1, while the validation set

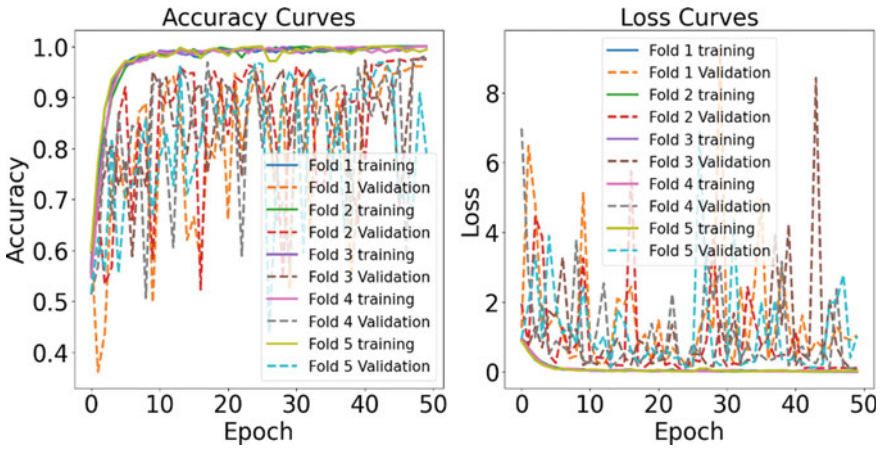


Fig. 11 XCeption accuracy and loss

goes from 0.6 to 0.2. The accuracy curves for the different folds are all similar, which suggests that the model is not overfitting to any fold of the data. The loss curves for the different folds are all similar, which suggests that the model is converging to a similar solution for all the folds.

The custom CNN model took approximately 200.23 s to train on each fold, and the total training time for all five folds was 1076.88 s. On the other hand, the ResNet50 model took approximately 517.73 s to train on each fold, with a total training time of 2586.49 s. The XCeption model took approximately 694.42 s to train on the first fold, and the total training time for all five folds was 3675.44 s. Comparing the training times, we can see that the custom CNN model had the shortest training time among the three models. It took significantly less time to train compared to both ResNet50 and XCeption. This could be due to several factors, including the architecture and complexity of the models. The ResNet50 model took more than double the training time of the custom CNN model, indicating that ResNet50 is a more complex and computationally intensive model to train. Conversely, the XCeption model had the longest training time, nearly four times that of the custom CNN model. XCeption is known for its deeper architecture and increased number of parameters, which could explain the longer training time. It is worth noting that training time alone does not necessarily determine the quality or effectiveness of a model. The choice of model should also consider factors such as accuracy, performance on the specific task, and available computational resources.

5 Conclusion

The key goal of this study was to analyze the application of state-of-the-art deep learning algorithms in the domain of Alzheimer's disease identification. Impressively, the XCeption model demonstrated exemplary performance, yielding extraordinary precision, recall, $F1$-score, and accuracy metrics of 98%. These findings unequivocally validate the model's remarkable prowess in discerning intricate patterns and discerning vital biomarkers associated with the pathogenesis of this complex neurological disorder. However, the utilization of deep learning for Alzheimer's disease detection necessitates addressing formidable challenges that impede its practical implementation. These challenges include the scarcity of available data, the heterogeneous nature of datasets, and the inherent difficulty of interpreting the intricate workings of the deployed models. Nevertheless, the integration of deep learning techniques in the realm of Alzheimer's disease detection holds immense potential for catalyzing transformative advancements in the field. By enabling early and accurate diagnosis, these approaches can facilitate timely intervention, enhance treatment efficacy, and augment our fundamental understanding of the multifaceted etiology underlying this debilitating ailment. To overcome these barriers and drive progress in this area, it is imperative to engage in continuous and interdisciplinary research, foster collaborative endeavors, and establish robust standards for data collection, harmonization, and analysis.

References

1. Islam J, Zhang Y (2017) A novel deep learning based multi-class classification method for Alzheimer's disease detection using brain MRI data. In: Brain informatics: international conference, BI 2017, Beijing, China, November 16–18, 2017, proceedings. Springer International Publishing, pp 213–222
2. Association A (2016) 2016 Alzheimer's disease facts and figures. Alzheimers Dement 12(4):459–509
3. Bekris LM, Lutz F, Yu CE (2012) Functional analysis of APOE locus genetic variation implicates regional enhancers in the regulation of both TOMM40 and APOE. J Hum Genet 57(1):18–25
4. Valenzuela O, San Román B, Guzman FMO, Villamor JLB, Saéz-Lara MJ, Rojas F, Rojas I (2014) Development of soft-computing techniques capable of diagnosing Alzheimers disease in its pre-clinical stage combining MRI and FDG-PET images. In: IWBBIO, pp 1644–1650
5. Chitradevi D, Prabha S (2020) Analysis of brain sub regions using optimization techniques and deep learning method in Alzheimer disease. Appl Soft Comput 86:105857
6. Liu S, Liu S, Cai W, Pujol S, Kikinis R, Feng D (2014) Early diagnosis of Alzheimer's disease with deep learning. In: 2014 IEEE 11th international symposium on biomedical imaging (ISBI). IEEE, pp 1015–1018
7. Luo S, Li X, Li J (2017) Automatic Alzheimer's disease recognition from MRI data using deep learning method. J Appl Math Phys 5(9):1892–1898
8. Ji H, Liu Z, Yan WQ, Klette R (2019) Early diagnosis of Alzheimer's disease using deep learning. In: Proceedings of the 2nd international conference on control and computer vision, pp 87–91

9. Bi X, Wang H (2019) Early Alzheimer's disease diagnosis based on EEG spectral images using deep learning. Neural Netw 114:119–135
10. Islam J, Zhang Y (2018) Early diagnosis of Alzheimer's disease: a neuroimaging study with deep learning architectures. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops, pp 1881–1883
11. Al-Shoukry S, Rassem TH, Makbol NM (2020) Alzheimer's diseases detection by using deep learning algorithms: a mini review. IEEE Access 8:77131–77141
12. Gulhare KK, Shukla SP, Sharma LK (2017) Deep neural network classification method to Alzheimer's disease detection. Int J Adv Res Comput Sci Softw Eng 7(6):1–4
13. Ding Y, Sohn JH, Kawczynski MG, Trivedi H, Harnish R, Jenkins NW, Lituiev D, Copeland TP, Aboian MS, Mari Aparici C, Behr SC, Franc BL (2019). A deep learning model to predict a diagnosis of Alzheimer disease by using 18F-FDG PET of the brain. Radiology 290(2):456–464
14. Zhao Z, Chuah JH, Lai KW, Chow CO, Gochoo M, Dhanalakshmi S, Wang N, Bao W, Wu X (2023). Conventional machine learning and deep learning in Alzheimer's disease diagnosis using neuroimaging: a review. Front Comput Neurosci17:10
15. Afzal S, Maqsood M, Khan U, Mehmood I, Nawaz H, Aadil F, Song OY, Yunyoung N (2021) Alzheimer disease detection techniques and methods: a review
16. Zhang F, Pan B, Shao P, Liu P, Shen S, Yao P, Xu RX, Alzheimer's Disease Neuroimaging Initiative (2022). A single model deep learning approach for Alzheimer's disease diagnosis. Neuroscience 491:200–214
17. https://www.kaggle.com/datasets/tourist55/alzheimers-dataset-4-class-of-images
18. Huang YJ, Powers R, Montelione GT (2005) Protein NMR recall, precision, and F-measure scores (RPF scores): structure quality assessment measures based on information retrieval statistics. J Am Chem Soc 127(6):1665–1674

Segmentation of White Matter Lesions in MRI Images Using Optimization-Based Deep Neural Network

Puranam Revanth Kumar, Rajesh Kumar Jha, and P. Akhendra Kumar

Abstract White matter lesions (WML) cause numerous brain diseases, and automatic WML segmentation is essential for evaluating the natural disease course and the efficacy of clinical interventions, especially drug discovery. This work presents a novel metaheuristic Harris hawks' optimization (HHO) to train the deep convolutional neural network (DCNN) as a robust model for WML segmentation. The proposed method was tested on MICCAI 2019 dataset. However, the WML regions would appear blurry and global inconsistency. In order to solve the fuzzy problem, we improved the network architecture by incorporating skip connections across the mirrored layers within encoder and decoder stacks. As a result, the segmented WML output appears more realistic and coherent with its surrounding contexts. The proposed optimization method, skip connections, has increased the segmentation performance of Dice score to 79.92%, precision 82.25%, recall 83.90%, and $F1$-score 82.94% for MRI dataset which is superior compared to other approaches. The proposed technique outperformed existing state-of-the-art methodologies in both qualitative and quantitative measurements across a wide range of medical modalities.

Keywords Brain MRI images · Deep learning · Residual connection · Segmentation · WML

P. R. Kumar (✉) · R. K. Jha · P. Akhendra Kumar
Department of Electronics and Communication Engineering, Faculty of Science and Technology (IcfaiTech), ICFAI Foundation for Higher Education, Hyderabad, India
e-mail: revanth123451.rk@gmail.com

R. K. Jha
e-mail: rajeshjha@ifheindia.org

P. Akhendra Kumar
e-mail: akhendra.p@ifheindia.org

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2023

S. Shakya et al. (eds.), *Fourth International Conference on Image Processing and Capsule Networks*, Lecture Notes in Networks and Systems 798,
https://doi.org/10.1007/978-981-99-7093-3_17

1 Introduction

White matter lesions (WMLs) are regions of aberrant myelination in the brain. These abnormalities are most easily seen on magnetic resonance imaging as hyperintensities, namely on T2 weighted and fluid-attenuated inversion recovery (FLAIR) sequences. Due to their strong link to a higher chance of cognitive damage, ischemic stroke, and mood and gait disorders, WMH are primarily considered to be established markers of cerebral small vessel disease (CSVD) [1]. In elderly people (95%), WML has been recognized as an established sign of cerebral small vascular disease (CSVD). Numerous degenerative processes and etiologies contribute to CSVD, which is a collection of illnesses affecting the brain's microvessels (arterioles, venules, and capillaries). Brain shrinkage, perivascular spaces, microbleeds, and recent small subcortical infarcts are all neuroimaging findings associated with CSVD [2]. When segmenting WMLs near cerebrospinal fluid (CSF) spaces, FLAIR sequence demonstrates a superior effect than T1-w and T2-w images. This is because FLAIR sequence inhibits high CSF signals by using a long inversion time [3]. Additionally, for the elderly population, FLAIR images show a decrease in contrast between white matter (WM) and gray matter (GM), leading to a uniformly low background signal and making WMLs prominent [4].

Brain lesions or white matter hyperintensities (WMH) are caused by the autoimmune and demyelinating disease [5, 6]. As a result, the probability of survival can be increased through a more accurate and earlier diagnosis of the WML. However, it is hard to get a correct diagnosis of different WML. A significant step toward facilitating efficient WML detection is the generation of an appropriate WML representation via optimization method. First, the primary objective of this research is to propose a Harris hawk's optimization (HHO)-based deep convolutional neural network method for WML segmentation. The major contribution is to detect the WML utilizing a combination of statistical and textural features. The DCNN is trained using the proposed HHO algorithm in such a way that the model parameters are learned in the most effective manner. The proposed algorithm inherits the HHO method for high global convergence feature. Thus, the proposed algorithm effectively renders precision while facilitating WML segmentation.

The remainder of the paper is structured as follows: Sect. 2 describes in detail the typical WMLs segmentation methodologies used in the literature and the difficulties encountered, which served as research problem for the development of the suggested method. Methodology is discussed in Sect. 3. Section 4 displays the results and comparative outcomes of the suggested method in contrast to other state-of-the-art methods, and Sect. 5 summarizes and proposes directions for further research.

2 Related Work

In recent times, deep learning [7] has drawn significant interest due to its powerful simulation abilities and the capacity to automatically acquire sophisticated features from the given data. DL has also proven significant success in image classification [8], target recognition [9], and image segmentation [10]. Convolutional neural networks (CNNs) [11], the lesion prediction algorithm [12], and U-Net [13] are only a few of the DL techniques that have been studied in the context of WML segmentation. The adaptability of CNN is demonstrated over their ability to learn features independently using a given dataset. CNNs show outstanding results in medical image segmentation and demonstrate generalizability by training on publicly available medical images [14], particularly in three-dimensional space. In general, it can be broken down into the three categories below. The segmentation problem is first reduced to a classification problem when using the voxel-based lesion segmentation framework. The core voxel's lesion status is predicted, and the lesion's location is identified. In order to segregate multiple sclerosis lesions, [15] created a unique architecture based on a cascade of two six-layer CNNs. The first was developed to detect voxels in the image that could contain a lesion, while the second network was developed to minimize the number of voxels that were incorrectly labeled [16]. Second, an up-sampling procedure helps remove duplicate data while maintaining the original resolution of the input in the patch-based lesion segmentation architecture. Multiple deep neural network architectures that take into account explicit location characteristics or multi-scale patches were integrated in [17] to segment WML lesions. A three-dimensional conditional random field (CRF) that takes into account both local and global context was presented in [18] as a means of reducing false positives in the post-processing stage. Finally, an encoder–decoder network representing a volume-based segmentation architecture is developed for enabling full-pipe WML segmentation output from MRI Images. A basic investigation of deep 3D convolutional encoder networks using Boltzmann machine theory was performed by [4]. To detect WML with multi-modality MRIs, [19, 20] used deep fully convolutional networks. Some proposed algorithms to solve ISP are listed in Table 1.

2.1 Research Problem

All the listed algorithms in the literature were proposed to overcome the ISP of normal images and medical images of the CT and MRI image types, but the main drawbacks of the above listed algorithms are that it is easy to fall into local optima and suffers from exploitation and exploration. There are now methods that can be combined to improve WML categorization performance. First, the high capacity of HHO can be used to balance between exploitation and exploration. Second, the suggested optimization method will train the tissue segmentation neural network weights. To get over the ISP problem in WML segmentation, we were inspired to

Table 1 Some proposed algorithms to solve ISP

References	Procedure
Ramya et al. [5]	In order to achieve automated detectability of punctate white matter lesions (PWMLs) in babies, deep learning models were constructed using a convolutional neural network called You Only Look Once v3 (YOLOv3). As a consequence of this, a threshold of detection probability between 20 and 30% was established for all deep learning models
Asghar et al. [21]	The threshold levels' bounds were explored using the cluster validity measure to locate the boundaries that may contain the ideal threshold values. Then, GA is applied to look for the best possible threshold values within the acquired boundaries
Guardia et al. [4]	The U-Net that was enhanced by highlighting foregrounds (HF) showed a significant improvement in the identification of WML voxels at the boundary of WMLs or in tiny WML clusters, both quantitatively and subjectively
Gilsoon et al. [14]	The focus of this work is on using WML volume to evaluate functional differences in people with moderate cognitive impairment (MCI). In this case, the ISP problem was resolved by employing a data-driven method of assessing the groups' respective power spectra
Thawkar et al. [16]	A particle swarm optimization (PSO) with the added benefits of collaborative and all-encompassing learning has been designed to take on the ISP. Together, PSO and cooperative learning can help mitigate the negative effects of the dimensionality curse and delay or eliminate the onset of early convergence

suggest a novel model that combines HHO and CNN. In order to segment WML from MRI images, this method is compared to numerous state-of-the-art methods.

3 Methodology

In this section, the DCNN weights was trained using the Harris hawk's optimization (HHO) algorithm as a robust model for the WML segmentation task. Also, we enhanced the network model by introducing skip connections between mirrored layers in the encoder and decoder stacks [7]. In general, skip connections tend to work well for deeper networks and tasks that require high levels of abstraction, such as semantic segmentation and image synthesis.

3.1 Harris hawk's Optimization (HHO)

The HHO algorithm [21] replicates Harris hawks' cooperative foraging and pursuing technique. To pursue fleeing prey, the hawks employ a separate strategy known as 'surprise pounce'. The fitness metric is dependent on the targeted prey for each repetition. The HHO [21] algorithm is divided into three steps: exploration, transition, and

exploitation. First, the target function and solution space, together with the relevant parameters, are specified.

3.1.1 Exploration Phase

In this phase, hawks perch randomly and wait for prey depending on two approaches stated in Eq. (1).

$$H_i(t+1) = \begin{cases} H_r(t) - r_1 \left| H_r(t) - 2_{r_2} H_i(t) \right| h \geq 0.5 \\ \left(H_{\text{pery}}(t) - H_m(t) \right) - r_3(lb + r_4(lb - ub)) h < 0.5 \end{cases} \quad (1)$$

where $H_i(t+1)$ represents the new position of the hawk, $H_r(t)$ represents the hawk that was picked randomly in the current iteration, $H_{\text{pery}}(t)$ represents the location of the prey, and $H_i(t)$ represents the position of each hawk in the system. Here, $r_1, r_2, r_3, r_4, ,$ and h are the randomized values in the range [0,1], and the lower and upper limit variables are *lb* and *ub*, respectively. When h is less than 0.5, the hawks perch in random places. When h is greater than 0.5, they stay at a place with other members of their population. Here, $H_m(t)$ denotes the current population's average location for hawks, and it is calculated using Eq. (2).

$$H_m(t) = \frac{1}{P_{\text{size}}} \sum_{i=1}^{P_{\text{size}}} H_i(t) \quad (2)$$

where P_{size} denotes the size of the current population. The transition between exploration and exploitation. At this stage, the following Eq. 3 is used to calculate the prey's energy level;

$$E = 2E_0 \left(1 - \frac{t}{T} \right) \quad (3)$$

where E, T, and E_0 represent the target's escape energy, total generations, and beginning energy state, respectively.

3.1.2 Exploitation Phase

In this phase, the hawks make a 'surprise pounce', striking the prey seen during the exploration phase.

Soft Besiege ($R \geq 0.5$ and $|E_{\text{esc}}| \geq 0.5$)—The hawks gently circle their prey while trying to get away and making jumps that are hard to predict. This behavior is specified by Eqs. (4) and (5).

$$H_i(t+1) = \Delta H_i(t) - E_{\text{esc}} \left| JS.H_{\text{pery}}(t) - H_i(t) \right|, \tag{4}$$

$$\Delta H_i(t) = H_{\text{pery}}(t) - H_i(t) \tag{5}$$

The xth iteration difference between the rabbit's position vector and its current location is denoted by $\Delta H_i(t)$. Once the hawks have begun their gentle besiege with dives, the final strategy to improve their positions may be carried out using Eq. (6).

$$H_i(t+1) = \begin{cases} X \text{ if fit}(X) < \text{fit}(H_i(t)) \\ Y \text{ if fit}(Y) < \text{fit}(H_i(t)) \end{cases} \tag{6}$$

Hard Besiege ($R \geq 0.5$ and $|E_{\text{esc}}| \geq 0.5$): The hawks are unable to completely encircle the prey that is attempting to flee since the prey does not have sufficient energy to do so, and this behavior is represented in Eq. (7).

$$H_i(t+1) = H_{\text{pery}}(t) - E_{\text{esc}} |\Delta H_i(t)|. \tag{7}$$

The position of the hawks can be updated using Eq. (8).

$$H_i(t+1) = \begin{cases} X \text{ if fit}(X) < \text{fit}(H_i(t)) \\ Y \text{ if fit}(Y) < \text{fit}(H_i(t)) \end{cases} \tag{8}$$

where fit(X) and fit(Y) denote the X and Y's fitness values, respectively. Equations (9) and (16) are used to determine the X and Y values in Eq. (10).

$$X = H_{\text{pery}}(t) - E_{\text{esc}} |JSH_{\text{pery}}(t) - H_i(t)| \tag{9}$$

$$Y = X + B \times \text{LF(Dim)} \tag{10}$$

Here, LF is the Levy flight function, B is a vector of size $1 \times$ Dim selected at random, $JS = 2(1 - r_5)$ prey escaping procedure which changed randomly in each iteration. r_5 represents random number inside (0, 1), and Dim is the dimension that describes the problem. Using the LF method, the hawk's motions may be replicated in the play. It is shown by Eq. (11).

$$\text{LF}(x) = 0.01 \times \frac{u \times \sigma}{|v|^{\frac{1}{\beta}}}, \sigma = \left[\frac{\urcorner\left(1 + \beta \times \sin\frac{\pi\beta}{2}\right)}{\urcorner\left(\frac{1+\beta}{2}\right) \times \beta \times 2\left(\frac{\beta-1}{2}\right)} \right]^{\frac{1}{\beta}} \tag{11}$$

In this case, where u, v are random normal distribution vector with the size of $1 \times$ Dim, β is a constant with a value of 1.5, and $\urcorner$ is a standard gamma function. To test the algorithm's performance, the proposed solution is examined for the maximum

number of iterations. In the meantime, the fitness measure is evaluated by taking into account the error rate in addition to a number of attributes using Eq. (12).

$$\text{Fitness} = \alpha\psi(A) + \eta\frac{|C|}{|F|} \tag{12}$$

Here, $\psi(A)$ stands for the error rate. The number of characteristics that will be used is denoted by the $|C|$, while the total number of features is indicated by $|F|$. $\alpha\epsilon(0, 1)$ and $\eta = (1 - \alpha)$ are the parameters that, respectively, characterize the classification performance and the duration of the feature selection. The Algorithm 1 [21] provides the detailed steps of HHO as follows:

Algorithm 1: HHO Pseudo Code

Inputs: The population size N and maximum number of iterations T
Output: The location of prey and its fitness value
Initialize the random population $X_i(i = 1, 2, 3, \ldots, N)$
while (stopping condition is not met) **do**
Calculate the fitness value of hawks
Set H_{prey} as the location of prey (best location)
for (each hawk (H_i)) **do**
Update the initial energy E_0 and jump strength J
$E_0 = 2rand() - 1, J = 2(1 - rand())$
Update the E using Eq. (3)
if (|E| ≥ 1) **then go to** ---------- Exploration phase
Update the location vector using Eq. (1)
if (|E| < 1) **then go to** ---------- Exploitation phase
if (r ≥ 0.5 and |E| ≥ 0.5) **then go to** ———— Soft besiege
Update the location vector using Eq. (4)
else if (r ≥ 0.5 and |E| < 0.5) **then go to** ---------- Hard besiege
Update the location vector using Eq. (7)
else if (r ≥ 0.5 and |E| < 0.5) **then got to** ---------- Soft besiege with progressive rapid dives
Update the location vector using Eq. (6)
else if (r < 0.5 and |E| < 0.5) **then go to** ---------- Hard besiege with progressive rapid dives
Update the location vector using Eq. (8)
Return H_{prey}

3.2 Proposed HHO-DCNN for WML Segmentation

This section presents the proposed HHO-DCNN approach for WML segmentation and describes how the feature vector is used to refine the segmentation. Segmentation is performed with a DCNN [18], and weights of DCNN are trained with the HHO. To identify hyperintensities, the proposed HHO helps to select the features from an

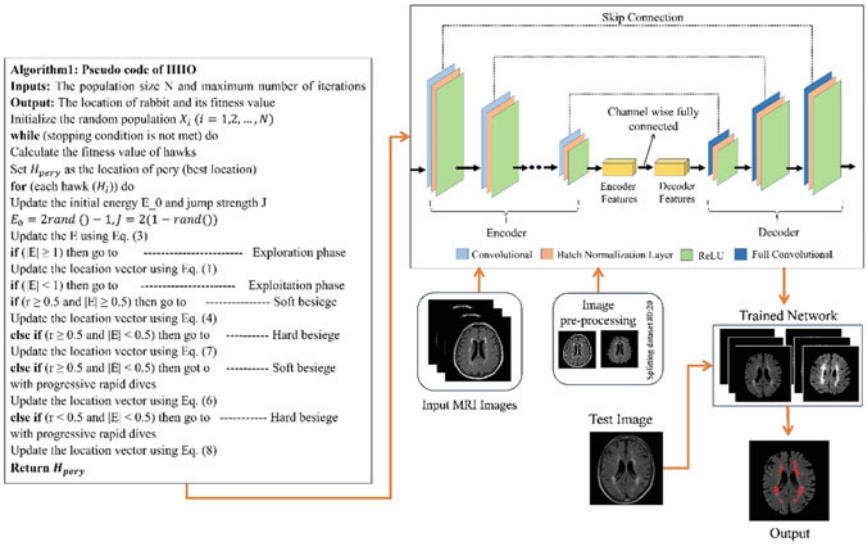


Fig. 1 Schematic view of proposed HHO-based DCNN for WML segmentation

input image. To reset the randomization problem, the HHO uses a pulsing mechanism, where the particles in the search space are randomly reset to their initial positions periodically. HHO can be applied to problems in the real world, even if the search space is unknown or complex. On the other side, HHO [21] draws its inspiration from the compliant manner and chasing style of Harris' hawks. In addition, HHO can deal with multi-modality difficulties, local optimal solutions, and deceptive optima in the search space. Therefore, a better solution is achieved with enhanced performance, and the rate of convergence is controlled by incorporating HHO with DCNN. Figure 1 depicts the HHO-based DCNN architecture, which makes use of a powerful encoder–decoder framework for feature embedding, residual connections for data flow, and bottleneck layers for model compression. We used ROBEX [34] for preprocessing to eliminate non-anatomical objects, which reduced the number of false positives.

3.3 Architecture of CNN

Here, Fig. 2 represents the schematic view of DCNN and Fig. 3 illustrates the segmented output of the proposed algorithm. Layers in a DCNN are responsible for certain tasks. These layers include the pooling layer, the convolutional layers, and the fully connected layers (FC). Features are generated by the convolutional layer, which also enables pattern extraction via a feature vector generated from the objects' segmentations.

$$T = \{T_1, T_2, \ldots, T_h, \ldots, T_i\} \tag{13}$$

where i represent the total number of convolution layers and T_h represents the hth convolution layer used in DCNN. The output can be derived from the units that are located at (p, w), and it is represented as:

$$(T_{\vartheta}^{h})_{p,w} = (v_{\vartheta}^{h})_{p,w} + \sum_{p=1}^{W_1^{p-1}} \sum_{z=-r_1^u}^{r_1^u} \sum_{s=r_2^u}^{r_2^u} (X_{f,p}^{u})_{z,s} * (T_{\vartheta}^{h-1})_{p+z,w+s} \tag{14}$$

where * designates the conv operator that allows for pattern extraction from the outputs of consecutive conv layers, $(T_p^{u-1})_{m+z,r+s}$ represents the fixed feature maps, W_1^{p-1} represents the total number of feature maps, and $(X_{f,p}^{u})_{z,s}$ represents the weights.

Features maps are built by convolving the inputs with the training weights associated with the neurons. Nonlinear activation functions are used to simplify functional mappings between input and output variables once the result has been obtained. Number of convolution layers (conv layers) in DCNN is used to describe the complexity of the network's ability to process data, with the input to the convolution layer being a segmented image produced by the sparking process.

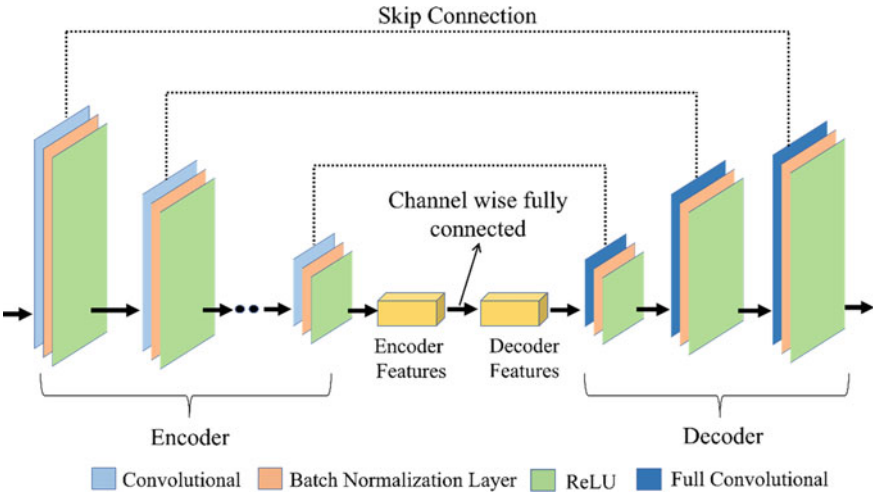


Fig. 2 Schematic view of proposed deep CNN

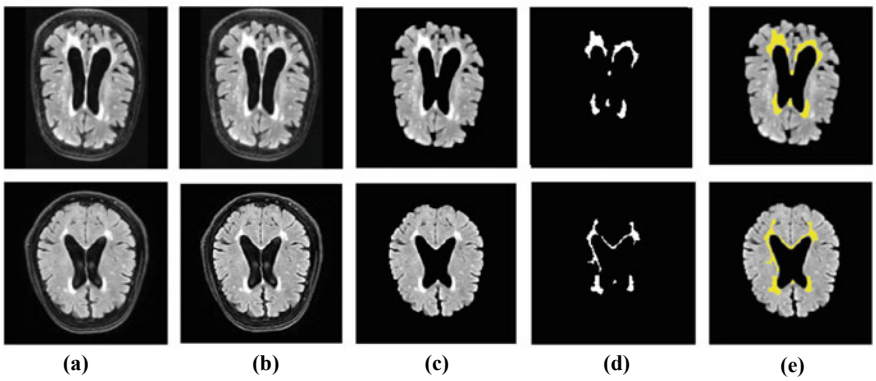


Fig. 3 Segmentation of WML on high-resolution MICCAI MRI images. **a** Original image, **b** enhanced image, **c** skull stripping, **d** binary mask, **e** segmented output

4 Results and Discussion

In this section, we present the results of extensive experiments intended to test the efficacy of the suggested algorithms and to compare them to some of the state-of-the-art techniques for dealing with the WML segmentation. In this work, we proposed a novel automated method, HHO-DCNN, and tested it on a large dataset consisting of 220 individuals from a variety of demographic backgrounds to segment WML from multimodal MRI images. In a quantitative evaluation of WML segmentation of 2D MRI images using a wide range of imaging procedures and clinical diagnoses, the proposed methodology outperformed earlier state-of-the-art approaches due to its advanced labeling techniques. In addition to detecting and segmenting other commonly existing cerebral lesions of vascular origin, this WML segmentation approach was created utilizing the 2D thick-slice protocol, which is more routinely used in regular clinical acquisition with numerous scanners.

4.1 Dataset

As given in Table 2, three datasets (Utrecht, Singapore, GE3T) were used in this study acquired from three different scanners.[1] There are 20 people in each dataset. To conduct the experiment's validation, we randomly assigned 80% of the subjects to the training group and 20% to the test set. Forty participants from two scanners were utilized as the training data for the cross-scanner trials, while the remaining twenty subjects were used as the testing data. Please take into account that we did not utilize complete 3D imagery but rather 2D images taken from the axial view.

[1] https://liwang.web.unc.edu/.

Table 2 Overview of the MICCAI WML images available for training and testing

Institute	Scanner	Training	Testing
UMC Uterche	3T Philips	25	35
NUHS Singapore	3T Siemens	35	45
VU Amsterdam	3T GE Signa	35	45

The suggested approach was successfully hick-validated using a dataset consisting of slice MRI images. MATLAB DL tool box was used to implement the proposed algorithm; the loss function used was weighted Dice loss. Images and annotated regions were uniformly down sampled in the x and y directions to a pixel spacing of 0.5 in order to account for differences in pixel spacing between scanners. There was no change to the z-axis pixel spacing. When training, 256 × 256 random 2D patches were sampled from the image volume as input. Training was performed on NVIDIA GE Force 8 GB GPU, with a learning rate of 0.001, the batch size was set to 128. Training required only 45 min, whereas testing and segmenting lesions from an unseen image took on average just 10 s.

The most significant contribution that the proposed technique makes is the establishment of a clinically usable and competitive method for the automated segmentation of WML in a wide range of application situations. These scenarios include the identification of WML across the presence of other intracranial lesions and the utilization of MRI images obtained using a variety of imaging protocols. The qualitative results of our proposed model WML segmentation are displayed in Fig. 3.

It is important to note that our proposed fully automatic approach outperformed human delineations using semi-automatic tools by an experienced observer working alone to segment WML (Table 3). Therefore, we believe that our method would be a secure and workable solution for segmenting WML in clinical settings.

Table 3 Performance of different quantitative evaluation matrices using automatic methods for WML segmentation

References	Dice score %	Precision %	Recall %	$F1$-score %
Sole et al. [4]	75.00	×	62.00	63.00
Todea et al. [3]	79.00	55.00	×	85.01
Huang et al. [2]	80.00	×	82.00	76.00
Zhang et al. [9]	88.00	×	76.00	80.00
Zhu et al. [10]	78.90	×	79.20	76.40
This work	79.92	82.25	83.90	82.94

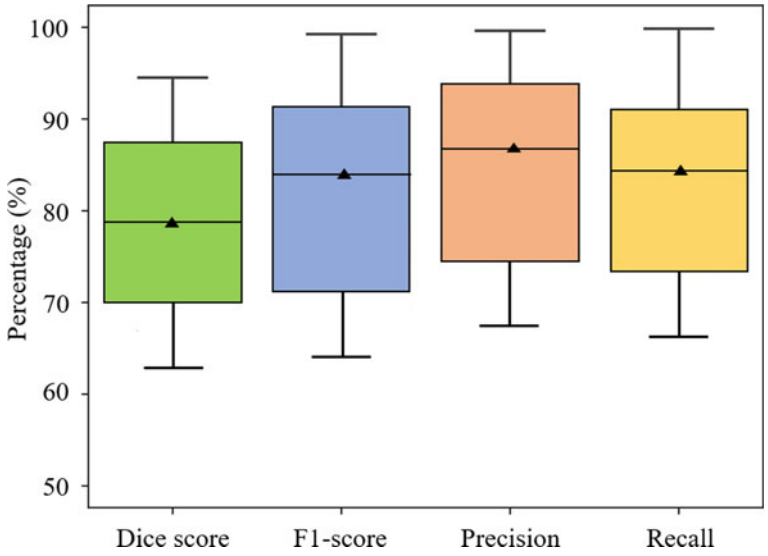


Fig. 4 Quantitative evaluation of proposed algorithm for WML segmentation

4.2 Quantitative Evaluation

The suggested HHO-DCNN method is tested quantitatively in terms of its performance with respect to Dice loss, $F1$-score, precision, and recall (see Fig. 4). Quantitative research involves manipulating the training data. In addition, the suggested HHO-DCNN uses the MACCAI 2019 dataset, demonstrating the effectiveness of the proposed technique.

Recall: described the portion of ground truth voxels that were corrected segmented as:

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{15}$$

Precision: The recall was calculated for both the combined set of labeled WML regions and the individual set of validated WML regions. The proportion of segmented voxels that are considered to be part of the ground truth regions is referred to as "precision."

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{16}$$

The metrics of interest in the aforementioned methods are the proportions of true positives (TP), true negatives (FN), and false positives (FP). Along with voxel-level recall and accuracy, we also developed the $F1$-score as a metric to evaluate

performance. For 3D linked component lesions, the recall formula was the same as in equation [15, 22, 23], but TP and FN were read as the percentage of lesions correctly segmented and the percentage of lesions missed.

The F1-score was the harmonic mean of recall and precision and was calculated as:

$$F1 = \frac{2 * \text{recall} * \text{precision}}{\text{recall} + \text{precision}} \tag{17}$$

Lesion recall and $F1$-score were created to better identify recall and precision characteristics at the voxel level.

As a result, we applied the Dice loss to prevent training from settling into a series of local minima, as shown in Eq. (17).

$$\text{Dice loss} = 1 - \frac{1}{C}\sum_{1}^{C}\left(\frac{2\sum_{i}^{N} P_i^c g_i^c}{\sum_{i}^{N}(P_i^c)^2 + \sum_{i}^{N}(g_i^c)^2}\right) \tag{18}$$

where $P_i \in P$ is the output of the algorithm, $g_i \in G$ is the real-world value, and C is the total number of classes. Definite WML and suspected WML were defined previously. The Dice loss was adjusted since the definitive WML was given greater consideration.

$$\text{Dice loss} = 1 - \frac{1}{C}\sum_{1}^{C}(w(x))^c\left(\frac{2\sum_{i}^{N} P_i^c g_i^c}{\sum_{i}^{N}(P_i^c)^2 + \sum_{i}^{N}(g_i^c)^2}\right) \tag{19}$$

where $(w(x))^c$ is the weight map.

The weight was set to 2, for regions with confirmed WML as $1 + f(\text{dis(Dis)})$, where (dis(Dis)) was the distance to the center of the closest region with confirmed WML. The value of $f(\text{dis(Dis)})$ was computed to be between 0 and 1.

$$f(\text{dis(Def)}) = 1 - \frac{\text{dis(Def)})}{\text{dis(Def)}_{\max}} \tag{20}$$

Dice Similarity Coefficient (DSC): measures the overlap percentage between G and P. It is defined as:

$$\text{DSC} = \frac{2 \times |G \cap P|}{|G| + |P|} \tag{21}$$

Our study has several limitations. First, WMLs may exist alongside and mix with other diseases, such as stroke lesions in T2-FLAIR MR imaging [1, 9, 24, 25]. The goal of this research was to develop an automated system for segmenting WMLs; as a result, the suggested method did not take into account labeling other illnesses. When we were training the model, it was assigning the same label to both

the background and other disorders. False positive findings for WMLs segmentation were seen, however, since other disease characteristics, such as stroke lesions, also present as hyperintense regions.

5 Conclusion

The proposed automated white matter hyperintensities segmentation algorithm, HHO-DCNN, has demonstrated its reliability and effectiveness as an extension to traditional state-of-the-art methods for WML patients. The method automatically segments WML in brain MRI images from MICCAI datasets, with a reasonable processing time of 45 min. Evaluation on a diverse set of 220 individuals, acquired from different MRI scanners and displaying various diagnoses and lesion loads, has shown that HHO-DCNN outperforms other existing methods in terms of segmentation accuracy and spatial agreement. This suggests that proposed method is a fast, reliable, and easy-to-use method with optimization for the automated segmentation of WML. Since the method includes a preprocessing stage, it is flexible enough to be used to enhance image quality, allowing for better results to be obtained even with low-resolution input. Maximum precision of 82.25%, maximum recall of 83.90%, dice score 79.92%, and maximum $F1$-score of 82.94% were achieved by the suggested HHO-DCNN, which was superior to all other approaches.

The proposed method's effectiveness in addressing various problem domains is notably constrained by its reliance on a fixed hierarchical structure and its deficiency in adaptive mechanisms. These limitations render it potentially less suitable for tackling dynamic or intricate optimization scenarios. Additionally, HHO's performance may be sensitive to parameter settings, and it can easily fall into a local optimum, requiring careful tuning to achieve optimal results. To overcome these issues in future, we employ a hybrid optimization based deep CNN to increase the accuracy by escaping from local optima.

References

1. Chandra A, Verma S, Raghuvanshi A et al (2023) PCcS-RAU-Net: automated parcellated corpus callosum segmentation from brain MRI images using modified residual attention U-Net. Biocybern Biomed Eng 43(2):403–427
2. Huang F, Xia P, Vardhanabhuti V et al (2023) Semisupervised white matter hyperintensities segmentation on MRI. Hum Brain Mapp 44(4):1344–1358
3. Todea AR, Melie-Garcia L, Barakovic M et al (2023) A multicenter longitudinal MRI study assessing LeMan-PV software accuracy in the detection of white matter lesions in multiple sclerosis patients. J Magn Reson Imaging
4. Solé-Guardia G, Custers E, de Lange A et al (2023) Association between hypertension and neurovascular inflammation in both normal-appearing white matter and white matter hyperintensities. Acta Neuropathol Commun 11(1):2

5. Balakrishnan R, Valdés Hernández MC, Farrall AJ (2021) Automatic segmentation of white matter hyperintensities from brain magnetic resonance images in the era of deep learning and big data—a systematic review. Comput Med Imaging Graph 88:101867
6. Gauber M, Lange C, Zimmermann I et al (2023) Performance evaluation of automated white matter hyperintensity segmentation algorithms in a multicenter cohort on cognitive impairment and dementia. Front Psychiatry 13:1010273
7. Shehata LAO, Ibrahim O, El-Kammash TH et al (2023) Gad. Brain volumetric and white matter structural connectivity alterations in autistic children: case–control study. Egypt J Radiol Nucl Med 54:36
8. Kuroda T, Ono K, Honma M et al (2023) Cerebral white matter lesions and regional blood flow are associated with reduced cognitive function in early-stage cognitive impairment. Front Aging Neurosci 16(15):1126618
9. Zhang H, Zhu C, Lian X, Hua F (2023) A Nested attention guided UNet++ architecture for white matter hyperintensity segmentation. IEEE Access 1–11 (2023)
10. Zhu W, Huang H, Zhou Y et al (2022) Automatic segmentation of white matter hyperintensities in routine clinical brain MRI by 2D VB-Net: a large-scale study. Front. Aging Neurosci 14:915009
11. Li X, Zhao Y, Jiang J et al (2022) White matter hyperintensities segmentation using an ensemble of neural networks. Hum Brain Mapp 43(3):929–939
12. Zhang Y, Duan Y, Wang X et al (2022) A deep learning algorithm for white matter hyperintensity lesion detection and segmentation. Neuroradiology 64:727–734
13. Pious AE, Sridevi UK (2023) A novel segment white matter hyperintensities approach for detecting Alzheimer. Comput Syst Sci Eng 44(3):2715–2726
14. Park G, Hong J, Duffy BA et al (2021) White matter hyperintensities segmentation using the ensemble U-Net with multi-scale highlighting foregrounds. Neuroimage 237:118140
15. Li H, Jiang G, Zhang J et al (2018) Fully convolutional network ensembles for white matter hyperintensities segmentation in MR images. Neuroimage 183:650–665
16. Thawkar S (2022) Feature selection and classification in mammography using hybrid crow search algorithm with Harris hawk's optimization. Biocybern Biomed Eng 42(4):1094–1111
17. Tran P, Thoprakarn U, Gourieux E et al (2022) Automatic segmentation of white matter hyperintensities: validation and comparison with state-of-the-art methods on both multiple sclerosis and elderly subjects. NeuroImage Clin 33:102940
18. Yang D, Masurkar A (2022) Early-stage MRI volumetric differences in white matter hyperintensity and temporal lobe volumes between autopsy-confirmed Alzheimer's disease, cerebral small vessel disease, and mixed pathologies. Dement Geriatr Cogn Dis Extra 12(1):69–75
19. Moeskops P, de Bresser J, Kuijf HJ et al (2018) Evaluation of a deep learning approach for the segmentation of brain tissues and white matter hyperintensities of presumed vascular origin in MRI. Neuroimage Clin 17:251–262
20. Dadar M, Manera AL, Ducharme S et al (2022) White matter hyperintensities are associated with grey matter atrophy and cognitive decline in Alzheimer's disease and frontotemporal dementia. Neurobiol Aging 111:54–63
21. Heidari AA, Mirjalili S, Faris H et al (2019) Harris hawks optimization: algorithm and applications. Future Gener Comput Syst 97:849–872
22. Sandrone S, Aiello M, Cavaliere C et al (2023) Mapping myelin in white matter with T1-weighted/T2-weighted maps: discrepancy with histology and other myelin MRI measures. Brain Struct Funct 228:525–535
23. Rieu Z, Kim J, Kim RE et al (2021) Semi-supervised learning in medical MRI segmentation: brain tissue with white matter hyperintensity segmentation using FLAIR MRI. Brain Sci 11:720
24. Rehan Afzal HM, Luo S, Ramadan S et al (2021) Automatic and robust segmentation of multiple sclerosis lesions with convolutional neural networks. Comput Mater Continua 66(1):977–991
25. Guo X, Ye C, Yang Y et al (2022) Ensemble learning via supervision augmentation for white matter hyperintensity segmentation. Front Neurosci 16:946343

A New Multi-level Hazy Image and Video Dataset for Benchmark of Dehazing Methods



Bedrettin Çetinkaya, Yücel Çimtay, Fatma Nazli Günay, and Gökçe Nur Yılmaz

Abstract The changing level of haze is one of the main factors which affects the success of the proposed dehazing methods. However, there is a lack of controlled multi-level hazy dataset in the literature. Therefore, in this study, a new multi-level hazy color image dataset is presented. Color video data is captured for two real scenes with a controlled level of haze. The distance of the scene objects from the camera, haze level, and ground truth (clear image) is available so that different dehazing methods and models can be benchmarked. In this study, the dehazing performance of five different dehazing methods/models is compared on the dataset based on SSIM, PSNR, VSI, and DISTS image quality metrics. Results show that traditional methods can generalize the dehazing problem better than many deep learning-based methods. The performance of deep models depends mostly on the scene and is generally poor on cross-dataset dehazing.

Keywords Hazy imagery · Reconstruction of images · Image enhancement · Poor visibility

B. Çetinkaya (✉) · Y. Çimtay · F. N. Günay · G. N. Yılmaz
Computer Engineering, Ted University, Ankara, Turkey
e-mail: bedrettin.cetinkaya@tedu.edu.tr

Y. Çimtay
e-mail: yucel.cimtay@tedu.edu.tr

F. N. Günay
e-mail: fnazli.gunay@tedu.edu.tr

G. N. Yılmaz
e-mail: gokce.yilmaz@tedu.edu.tr

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2023

S. Shakya et al. (eds.), *Fourth International Conference on Image Processing and Capsule Networks*, Lecture Notes in Networks and Systems 798,
https://doi.org/10.1007/978-981-99-7093-3_18

1 Introduction

Different conditions that affect the scattering of the light such as haze, fog, and smoke can significantly reduce information contained in the captured scene. This loss of information causes poor performance in many computer vision problems such as object detection/tracking/classification [1–3], depth estimation [4], and remote sensing [5]. Therefore, reconstructing hazy-free data have the potential to improve the performance of other computer vision problems.

Traditional approaches try to generate hazy-free images using hand-crafted features/priors, such as Dark Channel Prior (DCP) [6], Color Attenuation Prior (CAP) [7], and color-lines [8]. On the other hand, the modern methods [9–13] are based on deep learning and they require both hazy-free and hazy data for training. Therefore, the datasets have a big impact on the model performance. On the other hand, evaluating the cross-dataset performance of models gives information about their generalization capabilities. To this end, we published a new foggy image dataset, Indoor-Multi-Level Fog dataset (IMFD), for image dehazing problem and compare several state-of-the-art methods on IMFD.

2 Related Work

Image dehazing is one of the most attractive problems in computer vision. With the rise of deep learning, neural networks-based solutions have become more dominant rather than traditional methods such as DCP [6] and CAP [7]. Therefore, we divided previous studies into two main groups: (i) Traditional and (ii) Deep Learning-based methods.

2.1 Traditional Methods

Most of the traditional methods rely on prior information such as image saturation/attenuation and pixel values in the RGB channels and they try to model haze under the assumption of homogeneous distribution of haze, unlike the real-world scenario. He et al. [6] proposed one of the most important traditional methods named Dark Channel Prior (DCP). This method dehazes hazy images estimating transmission maps and atmospheric light. Due to the simplicity and efficiency of DCP algorithm, there are many improved versions [14–18]. Also, Zhe et al. [7] use Color Attenuation Prior (CAP). They use CAP to estimate the depth map of a hazy image and hazes are removed using this depth information. Moreover, Fattal et al. [8] exploit color-lines that come from 1D distribution of RGB pixel values. This color-line can successfully estimate transmission offsets of hazy images. Also, Berman et al. [19] remove haze

using haze-lines which enables estimating the distance map and the atmospheric light.

2.2 Deep Learning-Based Methods

Deep learning-based techniques have gained increasing popularity in recent times due to their successful performance. Cai et al. [9] introduced a CNN, DehazeNet that estimates a medium transmission map to reconstruct haze-free images. Also, Ullah et al. propose a lightweight version of DehazeNet. Unlike these works, Li et al. [20] directly estimated haze-free images without finding the output of any intermediate step such as transmission/distance map. Zhang et al. [10] proposed a GAN to jointly learn the atmosphere light and the transmission map. Singh et al. [21] employed GAN-based architecture which aims to learn multi-level complexity and multi-scale spatial features. Also, Wu et al. [11] successfully applied contrastive learning for image dehazing problem. Jin et al. [22] estimated uncertainty maps that are used as attention coefficients for the density of haze. Tran et al. [23] proposed a GAN model with DCP. Song et al. [12] used a vision transformer-based architecture to remove haze. Jiang et al. [13] proposed a hybrid model which aims to de- hazes hazy images and refines the outputs of dehazing step.

3 Datasets

This section reviews popular datasets used in the image dehazing problem.

Chic: Color Hazy Image for Comparison dataset, CHIC, [24] provides fog-free RGB images and their hazed versions with nine different haze levels in a well-controlled indoor environment. Also, it contains three different scenes and two different camera positions which are static and dynamic. 9 different images with different haze levels are presented. Scenes are cropped where full images size is 6000×4000. Scene-1_ cropped images size is 1566×2244 and Scene-2 is 2010×3528.

DenseHaze: DenseHaze dataset [25] contains 33 paired hazy and fog-free ground-truth images. It contains both indoor and outdoor scenes. Hazy images were created using professional haze machines and they have homogeneous haze distributions. The foggy sceneries were captured by using actual haze that was produced by expert haze generators. The identical visual information was recorded under the same lighting conditions in the respective haze-free and haze-filled scenarios. The goal of the DenseHaze dataset is to greatly advance single-image dehazing state of the art by pushing robust approaches for realistic and diverse hazy scenarios.

D-Hazy: D-Hazy dataset [26] contains 1409 hazy and corresponding fog-free images. It used NYU Depth [27] and Middelbury [28] datasets because hazy images were

generated synthetically using depth information in these datasets. Only indoor scene is available in this dataset. Due to the fact that in a hazy media, the scene brightness attenuates with distance, researchers are able to produce a comparable hazy scene with high accuracy based on the depth information and utilizing the physical model of a hazy medium.

IHaze: IHaze dataset [29] contains 35 paired hazy and haze-free ground-truth images. All scene contains a *MacBeth* color checker to simplify the color calibration process. Also, the same illumination conditions were used for all data.

NHHaze: NHHaze dataset [30] contains 55 paired hazy and fog-free ground-truth images and these hazy images are created using a professional haze generator. Moreover, hazy images reflect real-world scenarios because these images have non-homogeneous haze distributions. Also, real hazy data without ground-truths are available for the evaluation process. All images were collected from the outdoor environment.

OHaze: OHaze dataset [31] contains 45 paired hazy and fog-free ground-truth images from the outdoor environment. Also, it contains different illumination conditions because data were collected under different sunlight levels. Using traditional image quality measurements like PSNR, SSIM, and CIEDE2000, OHAZE is used to compare a sample collection of cutting-edge dehazing approaches.

Revide: Unlike the previous image dehazing datasets, Revide dataset [32] contains paired videos with hazy and haze-free data. Data were collected using 47 different scenes and each scene contains different haze levels and illumination conditions. By capturing the same scene (with or without haze) twice using a well-designed video acquisition system, they collect matched real-world hazy and haze-free movies that are precisely aligned. They also create a Confidence Guided and Improved Deformable Network (CG-IDN) for video dehazing in order to address the difficulty of utilizing temporal redundancy among the hazy frames.

4 IMF Dataset (IMFD)

Previously stated datasets are mostly based on static images and there is a lack of foggy video data. Another shortcoming is that they do not provide foggy scenes with a controlled level of fog except [24, 33]. To overcome that issues and enrich the foggy dataset literature, in this study, we present a new indoor foggy imagery dataset (Indoor-Multi-Level Fog Dataset) which was created in a light and fog-controlled environment. We present two different scenes which include objects with various distances from the camera. In addition, we provide the ground truth as both image and video data.

Hanwha Wisenet QNO-7082R Network camera [34] was used to capture the imagery of the scenes. Before recording, the camera was focused on the scenes. Video resolution was set as 1080p (1920×1080) with a ratio of 16:9. Frame rate and



Fig. 1 Scenes 1 and 2, and the depth of the objects with respect to the camera, respectively

video coding standard were set as 60 fps and H.264, respectively. IR filter was turned off. The camera was connected remotely, and during recording, all the conditions including the position of the camera were the same except the level of the fog. First of all, the ground-truth scene was captured and then the room was filled with fog up to the scene that cannot be seen by the eye. Then recording was restarted and maintained up to the fog density reduces and the scene approaches the ground truth.

Quenlite QL-2000RGB Fog Machine [35] was used for the purpose of fog production. This machine can be controlled remotely. Its fog emittance capacity and spraying distance are 560 m^3/min and 8 m, respectively. In one cycle, it can continuously emit for 50 s. Then it cools down and is ready to be triggered again. We repeated the cycle up to the fog becomes an opaque layer. We recorded the foggy scene during the fog was being evacuated from the room.

We captured the imagery in a rectangular room that is isolated from the environment so that no daylight comes in. The lamps on the ceiling were used to enlighten the room. In this way, the amount of light in the room was kept stable. We created 2 different scenes with different levels of fog. The ground-truth images and the depth of the objects are shown in Fig. 1 for scenes 1 and 2. We positioned different objects at different depths which is a crucial concept to measure the success of any dehazing method in terms of improving the visual quality and increasing the visibility distance.

The imagery of the scenes was captured while the fog was evacuating the room. The evacuation of the fog is controlled by a ventilation system. Therefore, evacuation speed was constant with respect to time. After saving the recording, we split the total video duration to 10 equal-length windows and snap a video frame corresponding to the ends of each interval. Figure 2 shows the frame selection process, where fr is the frame rate of the video and L is the duration of the video. Therefore, the frame number (FN) corresponding to each fog level can be found by using Eq. 1, where i and NL hold the fog level and the number of fog levels, respectively.

$$\mathrm{FN}_i = (\mathrm{NL} - 1) * \mathrm{fr} * L, i = 1, 2, 3, \ldots, \mathrm{NL} \tag{1}$$

Finally, we classified the video into 10 fog levels which are shown in Fig. 3. Therefore for each scene, we provide 10-level foggy images (with ground truth) and

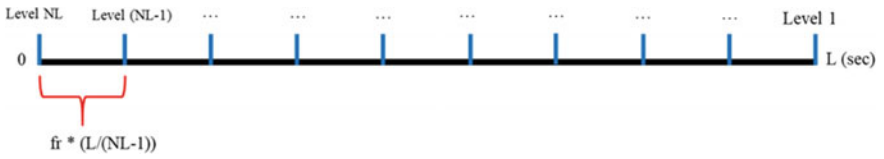


Fig. 2 Frame rate selection process

also the labeled foggy videos with respect to the starting and ending times of different fog levels. Our dataset is available at https://github.com/ycimtay/ITUFD.

5 Experiment

We evaluate our dataset using state-of-the-art methods published with official pre-trained weights for deep learning-based models. These methods are: EDN-GTM [23], AECR-Net [11], Light-DehazeNet [36], DehazeFormer [12]. Moreover, we used different pre-trained models of each of these studies to see their generalization performance. We also evaluate the results of DCP [6] and ω is set to 0.45. DCP is one of the fundamental dehazing methods which uses atmospheric light scattering model. Air light is estimated by using the dark channel pixel information. We use peak signal-to-noise ratio (PSNR), structural similarity index measure (SSIM), visual saliency-induced index (VSI) [37], and deep image structure and texture similarity (DISTS) [38] metrics for the evaluation process as given in Table 1. Peak signal-to-noise ratio (PSNR) is an expression for the ratio of the greatest potential value (power) of a signal to the power of the noise that distorts it and lowers the quality of its representation. The SSIM index is used to calculate how similar two images are to one another. A starting uncompressed or distortion-free image is used as the basis for the SSIM's prediction of image quality.

According to Table 1, EDN-GTM [23] trained with OHaze dataset performs better than all other methods for scene 1 in general. For scene 2, DehazeFormer [12] and EDN-GTM [23] have the best performances. On the hand, DCP [6] has considerable performance for both scenes 1 and 2. Other deep learning-based methods except for EDN-GTM and DehazeFormer have low generalization performance.

Moreover, we provide visual results for each method presented in Table 1, as shown in Fig. 4. We consider only the best results for each method among the pre-trained weights.

6 Results and Discussion

Level of the haze is one of the most important effects which determines the success of any dehazing method.



Fig. 3 Fog levels from 1 to 10 for scene 1

Table 1 Results for state-of-the-art methods in terms of SSIM, PSNR, VIS, and DISTS metrics for both ITUFD scenes 1 and 2

Method	Training datasets	ITUFD scene	Level 2				Level 5				Level 8			
			SSIM	PSNR	VSI	DISTS	SSIM	PSNR	VSI	DISTS	SSIM	PSNR	VSI	DISTS
DCP (CVPR 2009) [6]	–	1	**0.85**	16.35	0.90	0.58	0.83	15.10	0.90	0.46	0.84	15.47	0.90	0.40
AECR-Net (CVPR 2021) [11]	DenseHaze	1	0.42	12.08	0.83	0.37	0.49	11.61	0.85	0.37	0.50	12.22	0.84	0.35
	ITS	1	0.74	14.25	0.89	0.50	0.74	14.16	0.90	0.42	0.71	12.93	0.91	0.36
	NHHaze	1	0.44	10.15	0.81	0.39	0.43	9.32	0.86	0.37	0.46	9.89	0.88	0.37
Light-D.Net (IEEE TIP 2021) [36]	*	1	0.49	11.15	0.88	0.48	0.51	11.41	0.88	0.42	0.48	11.24	0.89	0.40
EDN-GTM (ISCSI 2022) [23]	DenseHaze	1	0.74	13.09	0.89	0.36	0.78	13.43	**0.91**	**0.33**	0.75	12.99	0.91	**0.31**
	IHaze	1	0.82	14.92	0.90	0.43	0.80	13.96	**0.91**	0.36	0.81	14.35	**0.92**	0.32
	NHHaze	1	0.80	13.76	0.89	0.44	**0.84**	14.13	0.90	0.41	**0.85**	14.56	0.91	0.38
	OHaze	1	**0.85**	**16.38**	0.88	0.52	**0.84**	**15.49**	0.90	0.43	**0.85**	**16.20**	0.91	0.39
DehazeFormer (IEEE TIP 2023) [12]	Indoor	1	0.73	12.36	0.89	0.50	0.65	10.50	0.90	0.42	0.70	11.26	0.91	0.39
	Reside6k	1	0.75	12.23	**0.93**	**0.31**	0.66	10.60	0.90	0.43	0.70	11.53	0.91	0.37
	RSHaze	1	0.83	15.06	0.86	0.51	0.82	14.92	0.86	0.41	0.83	15.22	0.87	0.39
DCP (CVPR 2009) [6]	–	2	**0.83**	17.39	0.89	0.60	0.81	16.72	0.92	0.41	0.84	17.93	0.93	0.32
AECR-Net (CVPR 2021) [11]	DenseHaze	2	0.42	13.50	0.84	0.40	0.48	12.34	0.84	0.36	0.56	15.19	0.89	0.29
	ITS	2	0.68	15.09	0.89	0.52	0.60	13.06	0.91	0.38	0.70	15.66	0.92	0.29
	NHHaze	2	0.33	11.64	0.81	0.41	0.46	10.01	0.85	0.35	0.51	11.56	0.88	0.31

(continued)

Table 1 (continued)

Method	Training datasets	ITUFD scene	Level 2				Level 5				Level 8			
			SSIM	PSNR	VSI	DISTS	SSIM	PSNR	VSI	DISTS	SSIM	PSNR	VSI	DISTS
Light-D.Net (IEEE TIP 2021) [36]	*	2	0.55	9.12	0.88	0.49	0.55	9.25	0.90	0.41	0.56	9.33	0.91	0.35
EDN-GTM (ISCSI 2022) [23]	DenseHaze	2	0.76	16.45	0.89	0.40	0.79	16.56	**0.93**	**0.30**	0.79	17.34	0.93	0.26
	IHaze	2	0.76	15.90	0.90	0.47	**0.82**	17.19	**0.93**	0.33	0.80	17.57	**0.95**	**0.24**
	NHHaze	2	0.70	14.11	0.86	0.45	0.79	14.53	0.91	0.39	**0.85**	18.18	0.94	0.29
	OHaze	2	**0.83**	**17.52**	0.89	0.54	**0.82**	16.62	0.91	0.41	**0.85**	**19.05**	0.92	0.32
DehazeFormer (IEEE TIP 2023) [12]	Indoor	2	0.74	15.07	0.89	0.54	0.74	14.76	0.92	0.36	0.81	17.08	0.94	0.28
	Reside6k	2	0.73	14.40	**0.93**	0.27	0.56	12.18	0.91	0.43	0.68	13.73	0.92	0.34
	RSHaze	2	**0.83**	17.02	0.88	0.54	**0.82**	**17.38**	0.89	0.39	**0.85**	18.91	0.91	0.30

While bold numbers show the best results for the corresponding level, metric, and scene, red numbers represent the second-best results. Also, ∗ means that the corresponding dataset was collected with public hazy images from Internet resources

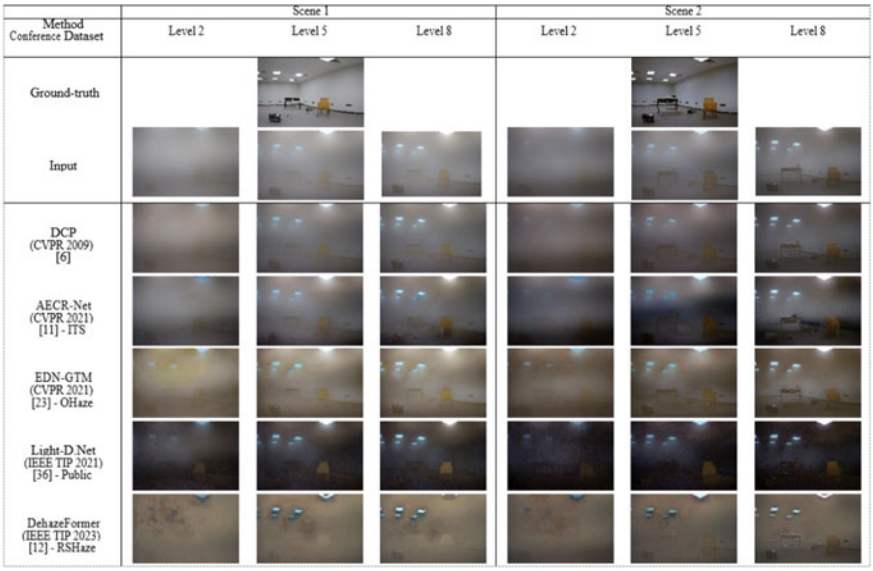


Fig. 4 Visual results for state-of-the-art methods. We visualize only the best result presented in Table 1 for each method

Most of the methods perform well when haze level is low. However, when the level is increased, the success drops dramatically. Therefore, in this study, we present a new multi-level color hazy image dataset which includes two scenes with different object distances from the camera. 5 different methods are compared on the dataset and results are presented. Performance of the dehazing methods can be represented by their generalization capability and adaptation to new hazy scenes. Compared to traditional methods, deep models perform better when tested on the same dataset they are trained with. However, cross-dataset dehazing performance is generally lower than the traditional methods. In the future, more effort should be given to develop deep models with high capacity of generalization [39].

7 Conclusion

In this study, a new multi-level hazy image and video dataset is proposed. By controlling the speed of the evacuation of the fog from the room, the fog level is adjusted. There are 10 level of foggy images of two scenes. DCP and the state-of-the art deep learning-based dehazing methods are tested on this new dataset. According to the results, DCP and EDN-GTM achieve promising results. Another important result is that deep learning-based methods have limited generalization capability when tested on a new dataset. This study is important and valuable since it brings a new

multi-level hazy dataset to dehazing literature. The state-of-the-art methods can be benchmarked on this dataset.

Acknowledgements This study is supported by The Scientific and Technological Research Council of Turkey (TUBITAK) within the Project number: 122E333.

References

1. Sindagi VA, Oza P, Yasarla R, Patel VM (2020) Prior-based domain adaptive object detection for hazy and rainy conditions. In: Computer vision—ECCV 2020: 16th European conference, Glasgow, UK, August 23–28, 2020, proceedings, part XIV 16. Springer, pp 763–780
2. Li C, Liang X, Lu Y, Zhao N, Tang J (2019) rgb-t object tracking: benchmark and base-line. Pattern Recogn 96:106977
3. Pei Y, Huang Y, Zou Q, Lu Y, Wang S (2018) Does haze removal help CNN-based image classification? In: Proceedings of the European conference on computer vision (ECCV), pp 682–697
4. Hambarde P, Murala S (2020) S2dnet: depth estimation from single image and sparse samples. IEEE Trans Comput Imaging 6:806–817
5. Kaplan NH (2018) Remote sensing image enhancement using hazy image model. Optik 155:139–148
6. He K, Sun J, Tang X (2010) Single image haze removal using dark channel prior. IEEE Trans Pattern Anal Mach Intell 33(12):2341–2353
7. Zhu Q, Mai J, Shao L (2015) A fast single image haze removal algorithm using color attenuation prior. IEEE Trans Image Process 24(11):3522–3533
8. Fattal R (2014) Dehazing using color-lines. ACM Trans Graph (TOG) 34(1):1–14
9. Cai B, Xu X, Jia K, Qing C, Tao D (2016) Dehazenet: an end-to-end system for single image haze removal. IEEE Trans Image Process 25(11):5187–5198
10. Zhang He, Patel VM (2018) Densely connected pyramid dehazing network. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 3194–3203
11. Wu H, Qu Y, Lin S, Zhou J, Qiao R, Zhang Z, Xie Y, Ma L (2021) Contrastive learning for compact single image dehazing. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 10551–10560
12. Song Y, He Z, Qian H, Du X (2023) Vision transformers for single image dehazing. IEEE Trans Image Process
13. Jiang N, Hu K, Zhang T, Chen W, Xu Y, Zhao T (2023) Deep hybrid model for single image dehazing and detail refinement. Pattern Recogn 136:109227
14. Xu H, Guo J, Liu Q, Ye L (2012) Fast image dehazing using improved dark channel prior. In: 2012 IEEE international conference on information science and technology. IEEE, pp 663–667
15. Wang J-B, He N, Zhang L-L, Lu K (2015) Single image dehazing with a physical model and dark channel prior. Neurocomputing 149:718–728
16. Wang Y, Wu B (2010) Improved single image dehazing using dark channel prior. In: 2010 IEEE international conference on intelligent computing and intelligent systems, vol 2. IEEE, pp 789–792
17. Zhu M, He B, Wu Q (2017) Single image dehazing based on dark channel prior and energy minimization. IEEE Signal Process Lett 25(2):174–178
18. Pei S-C, Lee T-Y (2012) Nighttime haze removal using color transfer pre-processing and dark channel prior. In: 2012 19th IEEE international conference on image processing. IEEE, pp 957–960
19. Berman D, Treibitz T, Avidan S (2018) Single image dehazing using haze-lines. IEEE Trans Pattern Anal Mach Intell 42(3):720–734

20. Li B, Peng X, Wang Z, Xu J, Feng D (2017) Aod-net: all-in-one dehazing network. In: Proceedings of the IEEE international conference on computer vision, pp 4770–4778
21. Singh A, Bhave A, Prasad DK (2020) Single image dehazing for a variety of haze scenarios using back projected pyramid network. In: Computer vision—ECCV 2020 workshops: Glasgow, UK, August 23–28, 2020, proceedings, part IV 16. Springer, pp 166–181
22. Jin Y, Yan W, Yang W, Tan RT (2022) Structure representation network and uncertainty feedback learning for dense non-uniform fog removal. In: Proceedings of the Asian conference on computer vision (ACCV), Dec 2022, pp 2041–2058
23. Tran L-A, Moon S, Park D-C (2022) A novel encoder-decoder network with guided transmission map for single image dehazing. Procedia Comput Sci 204:682–689
24. El Khoury J, Thomas J-B, Mansouri A (2018) A database with reference for image dehazing evaluation. J Imaging Sci Technol 62(1):10503–10511
25. Ancuti CO, Ancuti C, Timofte R, Gool LV, Zhang L, Yang M-H (2019) Ntire 2019 image dehazing challenge report. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops, IEEE CVPR
26. De Vleeschouwer Cosmin Ancuti C, Ancuti CO (2016) D-hazy: a dataset to evaluate quantitatively dehazing algorithms. In: IEEE international conference on image processing (ICIP), ICIP'16
27. Silberman N, Hoiem D, Kohli P, Fergus R (2012) Indoor segmentation and support inference from RGBD images. ECCV 5(7576):746–760
28. Scharstein D, Hirschmüller H, Kitajima Y, Krathwohl G, Nešić N, Wang X, Westling P (2014) High-resolution stereo datasets with subpixel- accurate ground truth. In: Pattern recognition: 36th german conference, GCPR 2014, Münster, Germany, 2–5 Sept 2014, proceedings 36. Springer, pp 31–42
29. Ancuti CO, Ancuti C, Timofte R, De Vleeschouwer C (2018) I-haze: a dehazing benchmark with real hazy and haze-free indoor images. arXiv:1804.05091v1
30. Ancuti CO, Ancuti C, Timofte R (2020) NH-HAZE: an image dehazing benchmark with non-homogeneous hazy and haze-free images. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops, IEEE CVPR
31. Ancuti CO, Ancuti C, Timofte R, De Vleeschouwer C (2018) O-haze: a dehazing benchmark with real hazy and haze-free outdoor images. In: IEEE conference on computer vision and pattern recognition, NTIRE workshop, NTIRE CVPR'18
32. Zhang X, Dong H, Pan J, Zhu C, Tai Y, Wang C, Li J, Huang F, Wang F (2021) Learning to restore hazy video: a new real-world dataset and a new method. In: CVPR, pp 9239–9248
33. El Khoury J, Thomas J-B, Mansouri A (2016) A color image database for haze model and dehazing methods evaluation. In: Image and signal processing: 7th international conference, ICISP 2016, Trois-Rivières, QC, Canada, May 30–June 1, 2016, proceedings 7. Springer, pp 109–117
34. Datasheet. Hhanwha wisenet qno-7082r network camera: https://hanwhavision.eu/product/qno-7082r/. Tech. Rep.
35. https://www.yonkamuzikmarket.com/quenlite-ql-2000-sis-makinasi-2000-watt-uzaktan-kumandali. Tech. Rep.
36. Ullah H, Muhammad K, Irfan M, Anwar S, Sajjad M, Imran AS, de Albuquerque VHC (2021) Light-dehazenet: a novel lightweight CNN architecture for single image dehazing. IEEE Trans Image Process 30:8968–8982
37. Zhang L, Shen Y, Li H (2014) VSI: a visual saliency-induced index for perceptual image quality assessment. IEEE Trans Image Process 23(10):4270–4281
38. Ding K, Ma K, Wang S, Simoncelli EP (2020) Image quality assessment: unifying structure and texture similarity. IEEE Trans Pattern Anal Mach Intell 44(5):2567–2581
39. Cetinkaya B, Cimtay Y, Gunay FN, Yilmaz GN (2023) A new multi-level hazy image and video dataset for benchmark of dehazing methods. arXiv:2307.16050

Creative AI Using DeepDream



Rakhi Bhardwaj, Tanmay Kadam, Shubham Waghule, Sahil Shendurkar, and Bhakti Sarag

Abstract The term "creative AI" refers to a branch of computer science that automates the creation of unique pieces of art, music, and literature. It allows technology to replicate and help human beings in the creation of unique and imaginative items. DeepDream, a computer vision tool, uses a deep neural network to find and improve patterns in images. The approach works by submitting an image to a trained neural network, which recognizes specific elements or patterns in the image and amplifies and enhances them. The current DeepDream algorithm's usage of a single image has the drawback of potentially producing overfitting, limiting the algorithm's ability to produce images that differ from the input image. As a result, the variety and originality of the developed graphics may be constrained. To get around this problem, DeepDream algorithm was trained to use multiple images or a combination of images and noise, resulting in more varied and inventive outputs. This allows us to create imaginative films as well as numerous visuals using just two images.

Keywords DeepDream · Inception-v3 · Convolutional neural network (CNN)

1 Introduction

A neural network is a machine learning (ML) technique inspired and similar to the human nervous system and brain structure. It is made up of processing units that are divided into input, hidden, and output layers. Each layer's nodes or units is linked to nodes in neighboring layers. A weight value is assigned to each link. At each unit, the inputs are multiplied by the weights and summed.

Recent advances in image categorization and speech recognition have been fueled by artificial neural networks.

R. Bhardwaj (✉) · T. Kadam · S. Waghule · S. Shendurkar · B. Sarag
Department of Computer Engineering, Vishwakarma Institute of Technology, Pune, India
e-mail: rakhi.bharadwaj@vit.edu

T. Kadam
e-mail: tanmay.kadam21@vit.edu

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2023

S. Shakya et al. (eds.), *Fourth International Conference on Image Processing and Capsule Networks*, Lecture Notes in Networks and Systems 798,
https://doi.org/10.1007/978-981-99-7093-3_19

An artificial neural network can be trained by exposing it to millions of training samples and gradually tweaking network parameters until the required classifications are produced. The network is typically composed of 10–30 stacked layers of artificial neurons. Each image gets loaded into the input layer, which communicates with the following layer until it reaches the "output" layer. This final output layer provides the "answer" for the network.

Convolutional Neural Networks (CNNs) have emerged as a powerful tool for image analysis and understanding. Among the various CNN architectures.

One of the issues with neural networks is determining what happens at each layer. Each layer gradually pulls more complicated information from the image, through training, until the final layer effectively decides what the image portrays. For example, the first layer can look for corners or edges. Intermediate layers examine the fundamental properties to find general forms or elements, such as a door or a leaf. In response to extraordinarily complex objects, such as entire houses or trees, these neurons light up. The final several layers integrate those interpretations into comprehensive ones.

1.1 Convolution Neural Network

There are many problems that can be solved with neural networks, such as classification, pattern recognition, dimensionality reduction, clustering, natural language processing (NLP), computer vision, predictive analysis, regression, and others.

Convolutional neural network, sometimes known as CNN, is a subtype of artificial neural network used largely for image processing and recognition. It is composed of numerous layers of interconnected neurons that process and transform visual data, and it is based on the biological structure of the visual cortex of the human brain.

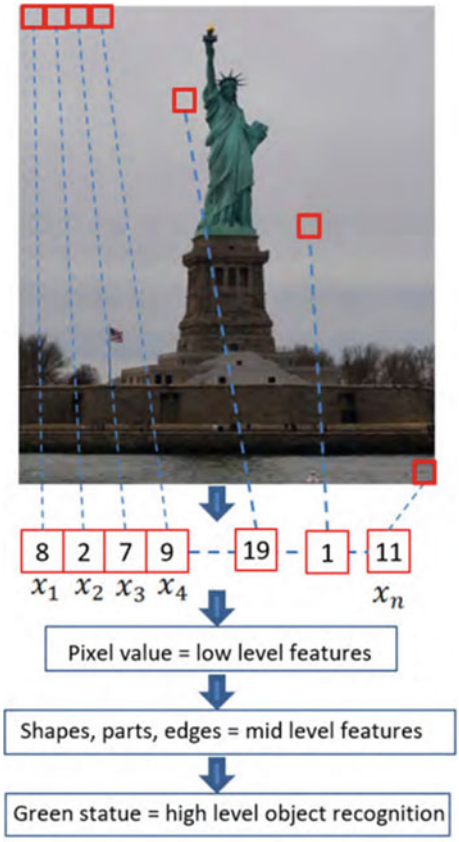
Convolutional layers, which apply a set of learned filters to the input data, are used in CNNs to automatically and adaptively learn spatial hierarchies of features from raw input data, such as pictures. These filters are taught to the network by training it on certain features such as edges or textures.

CNNs are designed to automatically and adaptively learn spatial hierarchies of features from raw input data, such as images, by applying convolutional layers that apply a set of learned filters to the input data. These filters, which are trained on a massive collection of labeled photographs, may recognize certain elements such as edges or textures.

CNNs have been used in a variety of applications, including self-driving cars, medical imaging, and robotic computer vision systems. CNNs have been demonstrated to be very effective at a variety of image recognition tasks, including object detection, facial recognition, and picture classification.

Figure 1 demonstrates how a convolution neural network (CNN) deep neural network can successfully recognize a higher-level object by learning hierarchical layers of representation from a low-level input vector. Only a broad generalization of the pixel values in the highlighted area may be seen in the red squares of the figure.

Fig. 1 Working of CNN



CNNs are able to extract higher-level representations of a picture after each layer, leading to image recognition.

1.2 What is DeepDream?

DeepDream is a computer vision approach that uses a neural network to generate attractive and frequently baffling visuals. Google created it in 2015 as a tool to experiment with deep learning and make artistic and abstract visuals.

Here, the input image is analyzed with a pre-trained neural network, which is made up of numerous layers of interconnected nodes to recognize various properties such as edges, textures, and various forms from the image. Based on the output of the previous layer, each network layer learns to recognize the complex features.

The DeepDream approach takes advantage of the neural network's hierarchical structure by inducing an artificial "dream" state, in which the network is urged to

enhance specific aspects present in the image. This is accomplished by altering the input image in such a way that the activation of specific neurons in the network is maximized. The resultant image is then transmitted back into the network, and the process is repeated, resulting in an iterative feedback loop that intensifies and exaggerates specific image patterns.

By changing the algorithm's settings, such as the layer of the network to focus on, the amount of zoom, and the severity of the effect, users can regulate the types of features that are amplified. This opens a vast array of creative opportunities and has produced a number of visually spectacular photographs.

DeepDream has been employed in different fields, including art, design, and advertising. It has also been utilized in scientific studies to better understand the inner workings of deep learning algorithms and to investigate the behavior of neural networks.

DeepDream has been employed in many different fields, including art, design, and advertising. It has also been utilized in scientific studies to better understand the inner workings of deep learning algorithms and investigate the behavior of neural networks.

1.3 Motivation

The DeepDream algorithm is a well-known picture generating approach that is frequently utilized in computer vision and deep learning. It is known for delivering distinctive and creative visual outputs by optimizing deep neural network activations for a particular input image. However, the usual DeepDream algorithm implementation employs a single input image, limiting the diversity of the generated outputs.

To address this restriction, DeepDream algorithm can be implemented with two input images. The goal of this method is to create more diversified and fascinating visual outputs by merging elements from two different photographs. This can result in the formation of novel and surprising visual combinations that would not otherwise exist in either of the input images alone.

Furthermore, DeepDream algorithm implementation using two images enables the generation of more sophisticated and layered visual outputs visual representations that can capture the essence of both input images at different levels of abstraction by optimizing the activations of multiple layers of a deep neural network. This can result in the creation of rich and detailed visual outputs suitable for several applications such as art and design, visual narrative, and scientific visualization.

Overall, the aim for DeepDream algorithm implementation combining two photographs is to push the boundaries of what is possible with this powerful image generating technology, as well as to explore new and creative ways to create visually attractive outputs.

2 Literature Survey

A deep neural network is made up of nodes arranged in layers. Numerous designs have been developed to address issues in various fields or use-cases. RNN is frequently used in time series problems/forecasting, while CNN is frequently used in computer vision and image recognition, for instance [1].

Since CNN is based on the human visual cortex, it is the preferred neural network for computer vision (image recognition) and video recognition. As the series of several convolution layers moves from the input to the output layers, each layer collects features with a finer resolution. Fully connected layers that perform categorization come after the convolution layers. A sub-sampling or pooling layer follows almost every convolution layer. A 2D, n-pixel picture, is provided to CNN. Every layer is made up of filters or kernels, which are collections of 2D neurons [2].

According to this study, combining these two approaches encourages the creative process and enhances the images that drug users and patients in mental health facilities employ to create hallucinations. The neural style transfer (NST) and deep dream models from this work are combined to produce a new image that incorporates the two approaches. Pre-trained networks for NST and deep dream are VGG-19 and Inception-v3, respectively. The style transfer method includes a crucial stage called the gradient matrix. With gradient fall in the first case and gradient rise in the second, the loss in style transmission is minimized in the first case and maximized in the latter. Based on the degree of clarity in each image, different loss values produced by various photographs are discovered. DeepDreams exhibited lower loss values than distortions, but NST had larger loss values. The clear pictures that lacked blending lines, circles, colors, or other shapes had the opposite effect [3].

AI algorithm Deep Inceptionism is probably the most peculiar one to date. This artificial intelligence was programmed to classify images. It enters the network, unlocks certain neurons, reveals some hidden layers, and attempts to determine the various features that are prefer to understand. AI algorithms, especially the DeepDream algorithms, have a variety of qualities that can be used to generate many kinds of art.

The paper explores the interpretability of deep learning-enabled image recognition algorithms in computer vision research in relation to theories from art history and cognitive psychology on how people perceive things related to vision and related to vision-related perceptual abilities. It is found that there are a surprising number of similarities between what can be learned about the machine-learned image and humanistic theories of visual perception, especially art historian Erwin Panofsky's image analysis technique and psychologist Eleanor Rosch's theory of graded categorization according to prototypes. This suggests that researchers in the arts and sciences would greatly benefit from closer cooperation. This paper argues that, in light of the quick development of image recognition technology, a resurgence of iconographic and formalist art historical research is required to influence future navigation and interpretation of all machine-learned images. This is accomplished by employing the programs Grad-CAM: Gradient-weighted Class Activation Mapping

created by the Machine Learning and Perception Lab at Georgia Tech, and Google's DeepDream as examples.

When you input an image, the neurons of a trained ANN become active and emit activations. The deep dream algorithm tries to (boost the neurons firing or activations) by trying to change the input image in a way that would lead some of these neurons to fire more frequently. You can decide which neurons in a given layer should fire more frequently. You may get a feel of how close organic neurons—which are what our human level intelligence was initially looking for—are by looking at these kinds of hallucinogenic results. The procedure is then repeated until all of the features that a particular layer was initially looking for are present in the input image. Strange characteristics can be seen in the original photograph. Using the GPU of an Intel Core i3-7100U CPU, I evaluated the Deep Inceptionism learning vision performance and obtained a lot of smoothen images with a type of trippy effect. The algorithm functions by producing dream Like Effects.

Deep learning has captured the interest of academics recently. Deep learning methods like convolutional neural networks (CNN) are commonly used to solve complex problems. It avoids the limitations of traditional machine learning methods. The objective of this study was to inform readers about a variety of CNN-related issues. In addition to outlining the three most popular topologies and learning techniques for CNN, this page also explains the network conceptually. By providing students with a comprehensive understanding of CNN, this study will inspire academics to continue their studies in this field. This study will be a source and handy guide for anybody who are interested in the subject.

Image style conversion is a technique for altering an image's background and general tone (style) without altering the source image's overall composition. This creates a new, aesthetically pleasing image that seamlessly combines the local characteristics of the source image with the style. The typical style conversion process can only synthesize images in the abstract painting style due to the limitations of the style conversion method, which prevent it from extracting the target image's high-level abstract components. German researchers like Gatys and others promoted the use of convolutional neural networks to achieve visual style conversion as a result of the rapid development of artificial intelligence algorithms and the deployment of funding in experimental facilities [1].

On the basis of this discovery, they were able to separate the style and content of the image and then combine the overall content and style of the image to achieve style conversion. They discovered the convolutional neural network through research and divided it into deep convolutional layers and shallow convolutional layers. The former obtained the overall framework of the image, while the latter obtained the style characteristics of the image. The gradient descent method must be used in the implementation to repeatedly update the pixel values of the source image, which significantly increases resource occupation and cost and introduces a number of issues with slow generation progress. However, the method successfully implements style conversion. Researchers have made many advancements as a result of Gatys' work, with Justin et al. proposing a rapid stylized conversion algorithm [2] that can transform the source image into the desired style after just one optimization

by establishing a feed-forward network beforehand, greatly enhancing generation speed.

3 Methodology

3.1 Tools Used

a. **TensorFlow**: Google's TensorFlow is a prominent open-source machine learning framework. One of its applications is in computer vision, particularly in the development of art using neural networks, termed "DeepDream". "TensorFlow" is used to implement the image generating process in addition to training the model. Users can produce highly customized and unique images by putting an image through the trained model and altering specific parameters, such as the "dreaminess" of the image.
b. **DeepDream**: is an image enhancement and generation approach that uses a convolutional neural network (CNN). It operates by passing an input image through the network's layers and amplifying specific features discovered at each layer. The generated image is then put back into the network, and the process is iteratively repeated to produce highly detailed and abstract images.
c. **DeepDream**: is implemented using TensorFlow, which provides an efficient way to define and train the CNN model. TensorFlow's computational graph design enables efficient parallel processing on GPUs, making it perfect for training large-scale neural networks such as DeepDreams.
d. **ImageNet**: In the DeepDream approach, ImageNet was used as a source of pre-trained neural network models. For this purpose, the Inception model, developed by Google researchers and winner of the 2014 ImageNet Large Scale Visual Recognition Challenge (ILSVRC), has been used extensively. The pre-trained Inception model is used as a starting point in the DeepDream technique to generate new images by optimizing the activation of certain neurons within the network. These activations are interpreted as patterns or properties recognized by the network, such as edges, textures, or forms.
e. **Inception-v3**: Inception-v3 is a deep convolutional neural network architecture developed by Google researchers as an extension to the previously developed Inception-v1 and Inception-v2 models. It was created to improve the accuracy of image classification tasks, particularly those requiring fine-grained object detection.

The Inception-v3 architecture is made up of convolutional layers, pooling layers, and fully linked layers. It also includes some innovative features, such as the use of factorized convolutions and batch normalization, which reduces computing costs while enhancing model accuracy (Table 1).

A number of network models are given in the table along with their top-1 error, top-5 error, and cost in billion operations (Bn Ops). The top-1 error is the likelihood

Table 1 Inception-v3 trained on ImageNet and compared with other contemporary models

Network	Top-1 error (%)	Top-5 error (%)	Cost Bn ops
GoogLeNet	29	9.2	1.2
BN-GoogLeNet	26.8	2013	1.5
BN-Inception [7]	25.2	7.8	2.0
Inception-v3-basic	23.4	–	3.8
Inception-v3-rmsprop RMSProp	23.1	6.3	3.8
Inception-v3-smooth LabelSmoothing	22.8	6.1	3.8
Inception-v3-fact Factorized 7 * 7	21.6	5.8	4.8
Inception-v3-BN-auxiliary	21.8	5.6	4.8

that the network will make an incorrect prediction, whereas the top-5 error is the likelihood that the network will make an incorrect prediction among the top-5 most likely outcomes. The price, which is expressed in billion operations, is what it costs to run the network computationally. The table demonstrates that as models get more sophisticated, their error rates go down while their computing costs go up.

f. **The PIL module** (short for Python Imaging Library): is a third-party Python library that allows you to view, manipulate, and save several image file types. The Image module is one of PIL's sub-modules that provides a convenient and user-friendly interface for working with images in Python.

3.2 DeepDream Implementation Using Tensorflow

Given a pre-trained convolutional neural network with parameters θ, the DeepDream algorithm aims to find an input image x that maximizes the activation of a set of target neurons. The activation of a neuron in a specific layer is represented as a function $f(x; \theta)$, where x denotes the input image and θ represents the network parameters. To quantify the activation of the target neurons, a loss function $L(x; \theta)$ is defined, providing a measure of how changes to the input image affect the target neuron activation. Common choices for the loss function include the sum or average activation of the target neurons.

To compute the gradients $\nabla L(x; \theta)$ of the loss function with respect to the input image x, TensorFlow's automatic differentiation capabilities are utilized. This computation enables the determination of the direction and magnitude of changes required in the input image to maximize the loss, thus emphasizing the activation of the target neurons. The gradients provide valuable information on how to modify the input image to amplify the desired neuron activations.

Based on the computed gradients, the DeepDream algorithm updates the input image x by iteratively applying incremental modifications. These modifications are determined by a step size α and are proportional to the gradients $\nabla L(x; \theta)$. The

iterative process involves adjusting the input image to progressively amplify the activation of the target neurons.

To enhance the visual quality and generate intricate patterns, the DeepDream algorithm further employs iterative refinement. At each iteration, the modified image from the previous step becomes the new input, and the gradient computation and image modification steps are repeated. This iterative process allows the algorithm to capture more detailed and visually interesting patterns over multiple iterations.

In addition to the core algorithm, DeepDream offers optional techniques to improve the generated images. These techniques may include smoothing or regularization, which help reduce noise or artifacts that may arise during the DeepDream process. By applying these techniques, the algorithm can produce visually appealing and aesthetically pleasing results.

3.3 Proposed System

3.3.1 Working of DeepDream

Convolutional neural networks (CNNs) have emerged as a powerful tool for image analysis and understanding. Among the various CNN architectures, Inception-v3 has gained significant attention for its exceptional performance in visual recognition tasks. In this research project, we leverage the capabilities of Inception-v3 to extract and amplify specific patterns present in images.

The initial layers of the Inception-v3 network, typically consisting of 2–3 layers, serve as the foundation for pattern extraction. These layers are responsible for detecting low-level visual features such as edges, corners, and textures. As we progress deeper into the network, the subsequent layers become increasingly adept at recognizing complex visual patterns, thanks to the inclusion of thousands of layers.

To amplify specific patterns in the images, we focus on three specific layers: “mixed-3”, “mixed-5”, and “mixed-7”. These layers represent deeper stages within the Inception-v3 architecture, where the network has learned to capture higher-level visual features. By analyzing the activations in these layers, we can emphasize and enhance the presence of specific patterns, enabling us to gain deeper insights into the visual content of the images.

Research aims to exploit the power of Inception-v3 by harnessing the discriminative capabilities of its intermediate and deep layers. By leveraging the network’s ability to recognize intricate visual patterns, we seek to uncover hidden structures, distinguish subtle details, and enhance the interpretability of the images under study (Fig. 2).

Based on the concepts of pattern recognition and image processing, the technique employs a pre-trained neural network to generate beautiful and hallucinogenic visuals by augmenting patterns and features in an image. DeepDream’s premise is like how human’s analyses shapes and patterns in ordinary items. Shapes that appear like animals, objects, or even humans are noticeable. Similarly, the DeepDream algorithm

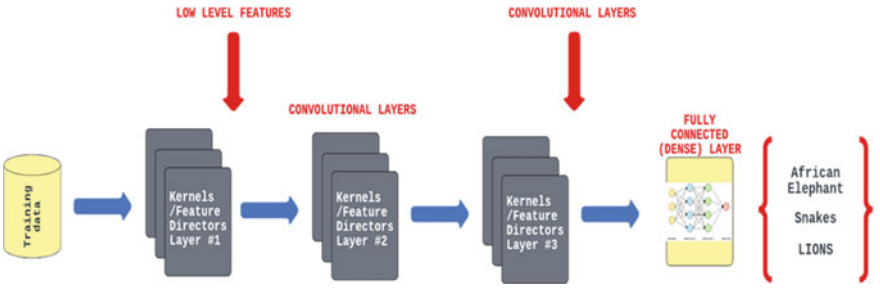


Fig. 2 Working of DeepDream algorithm

takes an image and attempts to analyse it by amplifying specific patterns and features learned by the neural network during its training phase.

From a mathematical aspect, the DeepDream algorithm selects specific layers within a pre-trained neural network to focus on. These layers have been trained to recognize forms, colors, and textures, among other things. The algorithm then improves on these qualities by increasing the layer's activations. As a result, the original image has been converted into a bizarre and hallucinogenic piece of art, with the patterns and characteristics enlarged and twisted in a unique and visually appealing way.

3.3.2 Mathematical Process Behind DeepDream

1. Convolutional Neural Network: DeepDream's first stage is to train a convolutional neural network (CNN) on a huge image dataset. By applying a collection of filters to the input image and sending the outputs through a sequence of nonlinear activation functions, the CNN learns to recognize patterns in images such as edges, corners, and textures.
2. Select a layer: Once the CNN has been trained, a layer of the network is selected to be used for feature visualization. The layer selected affects which features are highlighted in the final image. The network's deeper layers capture more abstract features, such as object sections or entire objects, and the network's earlier levels catch more low-level features, such as edges and textures.
3. Choose an input image: An input image is chosen to serve as the DeepDream algorithm's starting point. A photograph, a painting, or a computer-generated image are all examples of images.
4. Forward pass: The input image is processed by CNN's chosen layer to build a feature map. The activations of the neurons in the layer for the input image are represented by the feature map.

   Each activation in the feature map corresponds to a single feature recognized by the network.
5. Amplify the activations: Gradient ascent is used to amplify the activations in the feature map. Gradient ascent entails computing the gradient of the feature map

in relation to the input image and then modifying the input image to enhance the activations of the neurons that correspond to the desired features. This method is performed several times to get an output image that maximizes the activation of the selected characteristics.

6. Repetition: The preceding processes are repeated for different layers and features to generate a succession of output images that emphasize more abstract features in the source image.

$$y = b + m * x$$

This equation represents the relationship between the input image x and the network's activations or loss y. The trained model is represented as a linear function with two parameters: the line's intercept b (also known as the bias) and slope m (also known as the weight or coefficient). When the input image x is zero, the intercept b represents the baseline level of activations or loss, whereas the slope m reflects the change in activations or loss for a unit change in the input picture x.

As a result, the trained model generates a prediction for the activations or loss y based on the observed correlation between x and y during the training phase when it is applied to a fresh input image x. Through optimization methods like gradient descent, where the objective is to minimize the gap between the expected result of the model and the true output for a given input, the model parameters b and m are learned. The trained model can be trained to correctly forecast the activations or loss for new input images by changing the model's parameters.

$$\begin{aligned} \text{Gradient} &= \mathrm{d}y/\mathrm{d}x \\ &= d(\text{activation/loss})/d(\text{input image}) \end{aligned}$$

In this equation, dy/dx represents the gradient of the output variable y with respect to the input variable x, also known as the derivative of the output variable y with respect to the input variable x. In this case, y stands for the activation or loss function and x for the input image.

Figure 3 Activations with respect to the pixel of input image the activation or loss function's change in response to a minor modification in the input image can be observed by the derivative d(activation/loss)/d(input image). The gradient can be viewed as a vector pointing in the direction of the activation or loss function's steepest increase. Changes can be figured out from the input image to maximize or minimize the activation or loss by computing the gradient with respect to the input image

$$x|_{\text{new}} = x|_{\text{old}} + \mathrm{d}y/\mathrm{d}x * \text{learning rate(step size)}$$

is an update approach that is applied to iteratively modify the input picture x in optimization methods like gradient descent to reduce the loss function of a convolutional neural network (CNN).

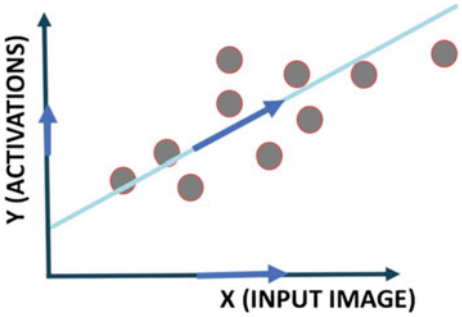


Fig. 3 Activations analysis

In this equation, $x|_{\text{old}}$ stands for the input picture's current value, dy/dx is the gradient of the loss function relative to the input image, and learning rate is a hyperparameter that controls the update's step size.

The direction in which the loss function descends most rapidly, or in which it is minimized, is indicated by the gradient dy/dx derived a step size that specifies how much the input picture should be modified in the direction of the negative gradient by multiplying the gradient by the learning rate.

The rate of learning can be used to regulate the strength of the update while updating the input picture in the direction that reduces the loss function, as shown by the equation $x|_{\text{new}} = x|_{\text{old}} + \mathrm{d}y/\mathrm{d}x * \text{learning rate}$. This update rule can be applied iteratively to improve the prediction by gradually modifying the input image.

Integrated each DeepDream image (blended image) into a video frame to generate a DeepDream video.

There are several ways to accomplish this, but one effective method is to use "zooming," which involves gradually zooming in on the image to create a video-like effect. Begin with the original DeepDream image and crop and resize it to generate a series of frames that gradually zoom in on the image. The DeepDream algorithm is then applied to each frame to create a new blended image that can be used as the next frame in the video.



3.3.3 Implementation

To create an amazing DeepDream video, a few steps need to be followed. First, a special place called the "dream location" is set up to store all the frames generated for the video. This dedicated spot keeps things organized and facilitates easy access to the frames as the project progresses.

Next, the dimensions of the blended image are determined. The choice of dimensions is crucial as larger images require more processing time. Experimenting with different dimensions helps find the right balance between quality and processing time.

Once the dimensions are set, the number of frames for the video is decided. Starting with a smaller value, such as 50 frames, is recommended to ensure the process is working correctly. As confidence builds, the frame count can be increased to create longer and visually stunning DeepDream videos. Having more than 500 frames provides ample material to work with.

To add dynamic movement and engagement, the blended image is zoomed in. This involves cropping the edges and resizing the image to its original size. By focusing on different parts of the image, a sense of depth and motion is created, enhancing the appeal of the DeepDream video.

Sometimes, the zooming effect can darken the image over time. Adjusting the GB values can counter this issue. Increasing the RGB values to 2 or above produces brighter DeepDream images, while reducing them below 2 results in darker images.

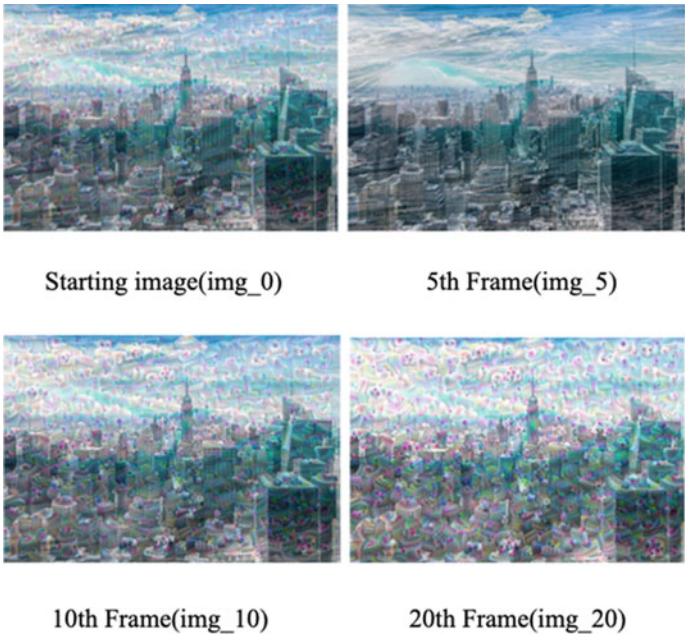
Experimenting with different values helps achieve the desired brightness and visual impact.

The deep dream model helper function can be utilized to customize the deep dream algorithm and achieve the desired image blend for the video.

To ensure visual consistency and coherence, the image needs to be clipped to fit within the constraints of the neural network. This step involves transforming the datatype of the clipped image result before converting it back to a regular picture. This maintains visual appeal and consistency in the DeepDream video.

All the generated frames are stored in the dream location, making them easily accessible and organized for use in the video.

Finally, the video can be created by specifying a name for the dream, providing the path of the frames to include, defining the codec, and creating a VideoWriter object. This object merges all the frames to produce the final DeepDream video. Following these steps allows the creation of a unique and visually stunning DeepDream video that brings the individual dream to life.



Here, the shape of the losses is (3), and the sum of all the losses is 0.55770755.

4 Result

Successfully integrating DeepDream with two photographs and making a DeepDream video involved the mixing of two images and applying the DeepDream algorithm. A unique and visually appealing video was created by experimenting with factors such as image dimensions, step size, and the zooming effect. The necessary frames were generated and a video was created by following the procedures mentioned in the approach.

These were the results of experimentation on Steps:

DeepDream model	Inception V3
Step start	0
Step size	0.001
Gradient step	[605, 910, 3]
Gradient at $x = 5$	3625
Inception layers	Mixed-3, mixed-5, mixed-7
Total steps	4000
Loss	0.5804304480552673
Params in Inception-v3	23,851,784

Overall, the outcomes of DeepDream implementation with two photographs and are eager to explore new possibilities with this fascinating method.

5 Conclusion

This case study focuses on the implementation of the DeepDream algorithm to mix two photographs and make a video. To integrate the two images, the Python Image Library (PIL) is utilized, and each DeepDream image is treated as a frame in the video. The image is zoomed in to make the film by chopping off the edges and resizing it to its original shape. RGB values greater than or equal to 2 produce brighter DeepDream images, whereas values less than 2 produce darker images. The DeepDream model helper function is used to change the number of steps and step size to match the intended dream. The image is trimmed to ensure that it fits within the neural network's constraints, and the frames are saved in a dream place for easy retrieval.

Finally, by defining the codec and creating a VideoWriter object, the frames are used to generate the video. Following these procedures will result in a visually spectacular DeepDream film that is tailored to the user's individual dream.

References

1. Fernández-Delgado M, Cernadas E, Barro S, Amorim D (2014) Do we need hundreds of classifiers to solve real-world classification problems? J Mach Learn Res 15(1):3133–3181
2. LeCun Y, Bengio Y (1998) Convolutional networks for images, speech, and time series. In: Michael AA (ed) The handbook of brain theory and neural networks. MIT Press, Cambridge, MA, USA, pp 255–258
3. Al-Khazraji LR, Abbas AR (2022) Employing neural style transfer for generating deep dream images. ARO-Sci J Koya Univ 10(2):134–141
4. Suzuki K, Roseboom W, Schwartzman DJ, Seth AK (2017) A deep-dream virtual reality platform for studying altered perceptual phenomenology. Sci Rep
5. Naul B, Bloom JS, Pérez F, van der Walt S (2018) A recurrent neural network for classification of unevenly sampled variable stars. Nat Astron 2(2):151–155
6. Najafabadi MM, Villanustre F, Khoshgoftaar TM, Seliya N, Wald R, Muharemagic E (2015) Deep learning applications and challenges in big data analytics. J Big Data 2(1):1
7. Goodfellow I, Bengio Y, Courville A (2016) Deep learning. In: Adaptive computation and machine learning. MIT Press, Cambridge, MA, USA, p 775
8. Gavin HP (2016) The Levenberg-Marquardt method for nonlinear least squares curve-fitting problems. Tech Rep
9. Glorot X, Bengio Y (2010) Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of 13th international conference on artificial intelligence statistics, pp 249–256
10. Chien S, Choo S, Schnabel MA, Nakapan W, Kim MJ, Roudavski S (2016) Artificial imagination of architecture with deep convolutional neural network. In: 21st international conference of the Association for Computer-Aided Architectural Design Research in Asia CAADRIA
11. Spratt L. Dream formulations: humanistic themes in the iconology of the machine-learned image. Kunsttexte.de Art Historical J
12. Boden MA (2004) The creative mind: myths and mechanisms. Psychology Press
13. Jennings KE (2010) Developing creativity: artificial barriers in artificial intelligence. Mind Mach 20(4):489–501
14. DiPaola S, Gabora L (2009) Incorporating characteristics of human creativity into an evolutionary art algorithm. Genet Program Evolvable Mach 10(2):97–110
15. Wiggins G (2006) A preliminary framework for description, analysis and comparison of creative systems. J Knowl Based Syst 19(7):449–458
16. Ritchie D (2007) Some empirical criteria for attributing creativity to a computer program. Mind Mach 17:67–99

Tuberculosis Bacteria Detection Using Deep Learning Techniques



Sharmin Akther Rima, Meftahul Zannat, Samia Shameem Haque, Al Kawsar, Sanjeda Sara Jennifer, and Ahmed Wasif Reza

Abstract Tuberculosis Bacteria (TB) is a bacterial infection that kills and affects millions of people annually. For the disease to be effectively treated and to be stopped from spreading further, early detection is essential. In this paper, an image processing-based deep learning approach is proposed for the detection of TB germs. The effectiveness of the suggested method is assessed using a sizable dataset of TB images, and the outcomes show that it successfully recognizes TB. With the potential to improve patient outcomes and slow the spread of the disease, the suggested method presents a promising approach for the early detection and diagnosis of TB. For the detection of TB, convolutional neural networks (CNNs) InceptionV3, DenseNet201and EfficientNetB3 have been used on a dataset of X-ray images, and the EfficientNetB3 model achieved the highest test accuracy, which is 100%.

Keywords Tuberculosis bacteria · Deep learning · InceptionV3 · DenseNet201 · EfficientNetB3

1 Introduction

Tuberculosis Bacteria (TB) is the world's second worst infectious disease, trailing only Human Immunodeficiency Virus (HIV) with an estimated 1.4 million fatalities in 2019 [1]. The tuberculosis-causing mycobacterium primarily affects the lungs but can also harm other vital organs. If not adequately controlled, it can be lethal and spreads through the air. Pulmonary Tuberculosis Bacteria (PTB) is a contagious type of tuberculosis that impacts the lungs. After settling in the lungs, the germs travel through the blood to several organs, including the heart, kidneys, and others [2]. If TB is not properly treated, it may get worse. Despite the possibility of drug resistance,

S. A. Rima · M. Zannat · S. S. Haque · A. Kawsar · S. S. Jennifer · A. W. Reza (✉)
Department of Computer Science and Engineering, East West University, Dhaka 1212, Bangladesh
e-mail: wasif@ewubd.edu

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2023

S. Shakya et al. (eds.), *Fourth International Conference on Image Processing and Capsule Networks*, Lecture Notes in Networks and Systems 798,
https://doi.org/10.1007/978-981-99-7093-3_20

therapy can successfully treat the illness with antibiotics for a period of six months [3]. For ease of use and safety, chest X-rays are preferred for detecting tuberculosis [4]. Another option is to have a doctor examine chest radiographs but this requires extensive clinical work and takes time [5].

Transfer learning can diagnose tuberculosis quickly and accurately. Clinical testing, physical examinations, medical histories and imaging methods are all used in the diagnosis process. The goal of the study is to suggest an enhanced model for quickly diagnosing bacterial lung infections. In order to do that, four deep learning models, namely CNN, InceptionV3, DenseNet201 and EfficientNetB3 have been implemented. EfficientNetB3 displayed the best performance with 100% accuracy. Some of the major challenges associated with manually diagnosing tuberculosis are:

- Difficulty in diagnosing due to its variety of symptoms and severity.
- The symptoms of tuberculosis and other respiratory illnesses overlap, making a diagnosis difficult.
- In cases of extrapulmonary tuberculosis, conventional diagnostic techniques may not be sensitive enough.
- Testing for tuberculosis, particularly molecular testing, can be costly and not always easily accessible, especially in low-resource healthcare settings.

The major body of this paper is structured as follows: Our literature review is covered in Sect. 2 and the materials and methods for implementation are covered in Sect. 3. Results and discussion are covered in Sect. 4, and our research is wrapped up with a conclusion in Sect. 5.

2 Literature Review

Numerous techniques, including sputum microscopy, molecular assays, and skin tests, are used in clinical practice to diagnose tuberculosis bacteria (TB), taking into account risk factors, medical history and symptoms. Rapid diagnostic instruments like GeneXpert MTB/RIF (Mycobacterium tuberculosis complex and resistance to rifampin) can identify both TB and resistance. The accuracy of diagnostics is improved by advances in machine learning and artificial intelligence (AI). Serological testing, nucleic acid amplification tests, imaging, computed tomography and positron emission tomography (CT and PET) and non-invasive tools like breath analysis and biosensors are a few of the alternative methods under investigation. These methods seek to enhance early detection, the start of therapy and effective TB management in medical settings [6].

Large datasets of clinical data, genetic data and medical imaging are analyzed by machine learning algorithms to demonstrate their potential for TB identification. In order to detect and diagnose TB early, these algorithms recognize its patterns. Clinical data evaluation and prediction models frequently use random forests and support vector machines (SVMs). Additionally, ML can use genetic information to identify TB strains, anticipate medication resistance and forecast treatment outcomes.

The detection of TB is improved when machine learning (ML) and nuclear acid amplification test (NAATs) are used together. For the development and application of ML models in TB diagnosis, it is essential to overcome issues with data quality, biases and ethics [7].

Convolution neural networks (CNNs) are quite good at identifying TB in X-ray and CT scan pictures used for diagnosis. Transfer learning with pre-trained models improves performance, while deep learning algorithms extract features and spot TB-related anomalies. The use of trained algorithms and picture masks is crucial. Deep learning algorithms must be used in TB diagnosis, and this requires overcoming biases, assuring high-quality datasets and addressing ethical issues [8].

InceptionV3 with transfer learning was used by Karaca and Güney to identify tuberculosis with 98.90% accuracy. The research's crucial contribution was identifying the constraints placed on radiologist availability and expertise, as well as the challenges associated with computer training [9].

To discriminate between photos of tuberculosis and healthy images, Rahman and Khandakar used a variety of deep neural networks, with ChexNet attaining an accuracy of 96.47%. The paper's key contribution was faster TB diagnosis by computer usage, and its limitations included the possibility that CNN model performance may be enhanced even further [10].

The statistics show that Stefanus Kieu Tao Hwa's suggested ensemble technique, when utilized to diagnose tuberculosis using chest X-rays, achieved a maximum accuracy of 89.77%, a sensitivity of 90.91% and a specificity of 88.64%. The work's significance was the improvement of TB detection through the use of ensemble classifiers, although additional components may further enhance TB detection [11].

3 Materials and Methods

3.1 Dataset Description

In order to proceed with the TB detection experimentation, we have utilized the Tuberculosis Bacteria (TB) Chest X-ray Database, which is publicly available in Kaggle [12]. A database of chest X-ray images for Tuberculosis Bacteria (TB) positive cases as well as normal images has been developed by a group of researchers from Qatar University, Doha, Qatar and Dhaka University, Bangladesh, along with their collaborators from Malaysia and medical professionals from Hamad Medical Corporation and Bangladesh. With an accuracy of 98.3%, the research team was able to distinguish between TB and normal chest X-ray images. There are 4200 X-ray images that are divided into two classes. The first class represents normal lungs, while the second class represents lungs affected by tuberculosis bacteria. However, the dataset is imbalanced, where there are 3500 images of normal chest X-ray and 700 images are Tuberculosis Bacteria affected. The image height and width are 512 and 512, respectively. Figure 1 shows the raw images of the dataset.

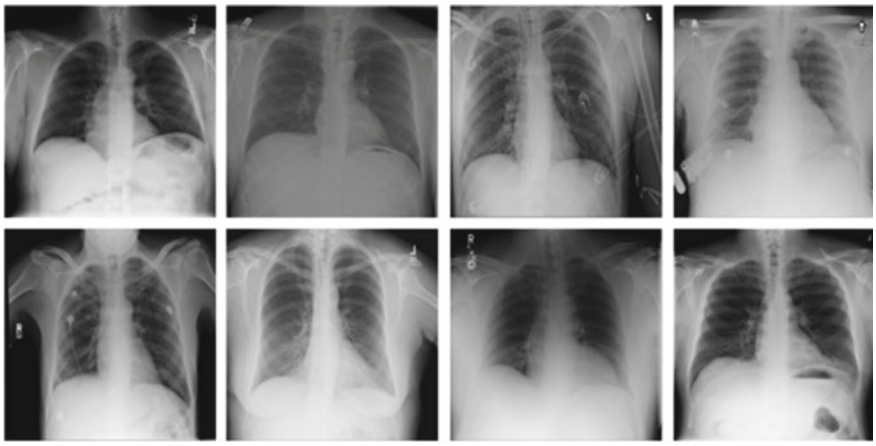


Fig. 1 Exemplary dataset inputs

3.2 Dataset Pre-processing

The pre-processing of image input in our classification is crucial for improved model performance and accuracy. The image height and width are 512 and 512, respectively, before pre-processing. In the three models, including InceptionV3, Densenet201 and CNN, the data was resized by 200*200, and in EfficientNetB3, by 224 * 224. The dataset has been split into three parts: training, testing and validation. Training sets make up 70% of the dataset, followed by testing sets at 15%, and validation sets at 15%. After splitting, the training dataset contains 2940 images, testing contains 630 images and validation contains 630 images.

3.3 System Architecture and Implementation

The process of the recommended framework is shown in Fig. 2. The suggested system covers model training, data pre-processing and results for a deep neural network that classifies chest images. The experimental model and data pre-processing are the two essential steps in the research process. The images have been resized in preparation for processing. After that, training, testing and validation were conducted on the dataset. The dataset and results have been examined using four deep learning models: CNN, DenseNet201, Inception V3 and EfficientNetB3.

This study employed four well-established deep learning models, namely InceptionV3, DenseNet201, CNN and EfficientNetB3. The selection of algorithms was guided by several critical criteria aimed at achieving superior model performance and efficient training. The primary consideration was the model's performance in

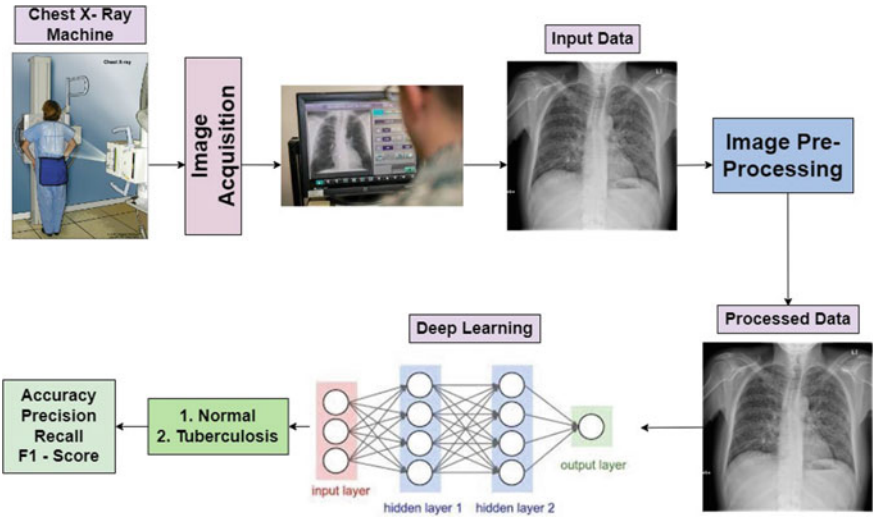


Fig. 2 Hierarchical overview of the system architecture

the specific classification task. These models have demonstrated impressive capabilities in diverse computer vision applications. We applied consistent hyperparameters and resized the images to 200 × 200 pixels for computational efficiency and improved performance. For InceptionV3, DenseNet201 and CNN, feature extraction was performed with a fixed image size of 200 × 200 pixels but for EfficientNetB3, we resized it into 224 × 224. The training of the deep learning model begins by employing a data frame, which provides a structured and efficient way to organize and manage the dataset, ensuring easy accessibility and proper data formatting during the training process. For feature extraction, the output layers of a pre-trained base model are modified to suit the specific classification task. Instead of using the original output layer, a custom output layer is created specifically for this research. Subsequently, the entire model is trained, incorporating the base model with the modified output layer. To enhance the model's performance, batch normalization is applied to the output of the base model. Batch normalization helps improve the convergence of the training process and reduces the negative impact of internal covariate shifts. Additionally, dropout regularization is introduced on the dense layer of the model to prevent overfitting during training, thereby increasing the model's generalization capability. Learning Rate Scheduling and Early Stopping strategies are also incorporated into the training process. With the modified callback, user interaction is enabled during training, allowing dynamic adjustment of the learning rate. This flexibility helps fine-tune the learning rate as the model progresses, and the dwell option further facilitates automatic learning rate adjustments. As a result, the model's convergence can potentially be accelerated, as the learning rate adapts to the changing characteristics of the training process. To optimize the model's learning process, it is compiled with the Adamax optimizer, which is well-suited for handling sparse

gradients and has shown to be effective in various deep learning tasks. As for the loss function, categorical cross-entropy is used, which is appropriate for classification problems. The categorical cross-entropy loss measures the dissimilarity between the predicted probability distribution and the true label distribution. By combining these techniques, the proposed model benefits from improved convergence, reduced overfitting, and a better ability to handle classification problems, ultimately leading to enhanced performance and more accurate predictions.

EfficientNetB3 (proposed model): This paper aimed to optimize the performance of the EfficientNetB3 model by exploring different parameters. We resized the images to 224 × 224 pixels. To prevent overfitting, we used a batch size of 30, batch normalization and a dropout rate of 40%. The learning rate was set to 0.001, and both L1 and L2 regularization were employed. We utilized the "Adamax" optimizer and employed the "categorical_crossentropy" loss function. Figure 3 presents the architecture of the EfficientNetB3 model and the proposed EfficientNetB3 model, providing a visual representation of its structure and design.

4 Results and Discussion

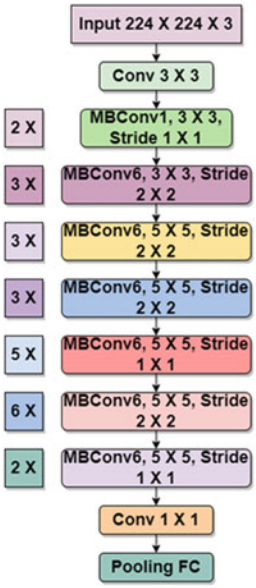
4.1 Experimental Results

The experimental result analysis of InceptionV3, DenseNet201, CNN and EfficientNetB3 is given in Table 1, where EfficientNetB3 has the greatest accuracy compared to the other models.

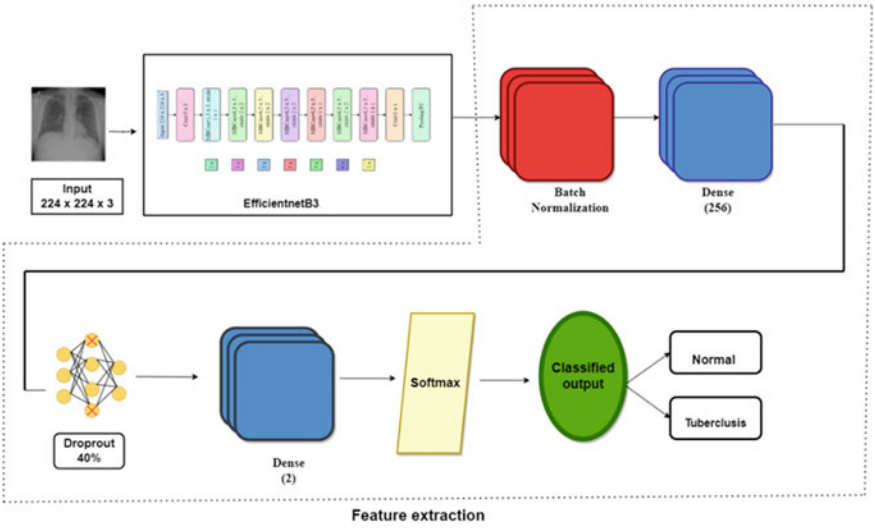
The high accuracy of the EfficientNetB3 model stems from a synergistic combination of architectural innovations, transfer learning, and learning rate adjustment. The EfficientNet architecture, utilizing compound scaling, optimizes the model's depth, width and resolution to achieve a balance between complexity and computational efficiency. Leveraging pre-trained weights from large-scale datasets such as ImageNet facilitates efficient transfer learning, providing a strong foundation for fine-tuning the target dataset with limited labeled examples.

Moreover, our decision to adopt larger input image sizes, enabling the model to capture intricate details and patterns, significantly enhances its recognition capabilities. To further fine-tune the model, we efficiently use computational resources and incorporate a dynamic learning rate adjustment strategy. This strategy ensures the model converges faster and prevents overfitting, ultimately leading to superior accuracy on the image classification task.

In order to evaluate the models and their performances, precision, recall, $F1$-score and confusion metrics were used. The mathematical notations are given in Eqs. (1), (2), and (3), Fig. 4 gives a visual representation of the test accuracy of all the models, and the evaluation outcome is presented in Table 2.



a) EfficientnetB3 Model



b) Modification of EfficientnetB3 Architecture

Fig. 3 Architecture and the modified version of the EfficientNetB3 model

Table 1 Result analysis

	Model	Training accuracy (%)	Validation accuracy (%)	Test accuracy (%)
1	InceptionV3	95.17	95.32	95
2	DenseNet201	98.18	98.69	92
3	CNN	96.88	96.29	96
4	EfficientNetB3	99.90	99.84	100

$$\text{Precision} = \frac{\text{true positive}}{(\text{true positive} + \text{false positive})} \tag{1}$$

$$\text{Recall} = \frac{\text{true positive}}{(\text{true positive} + \text{false negative})} \tag{2}$$

$$F1\text{Score} = \frac{2 \times (\text{precision} \times \text{recall})}{(\text{presion} + \text{recall})} \tag{3}$$

As given in Table 2, DenseNet201, InceptionV3 and CNN had 92%, 95% and 96% accuracy, respectively, whereas EfficientNetB3 outperformed all the models achieving 100% test accuracy. Figure 5 shows the accuracy and loss of EfficientNetB3 during training and validation, and the confusion matrix is shown in Fig. 6.

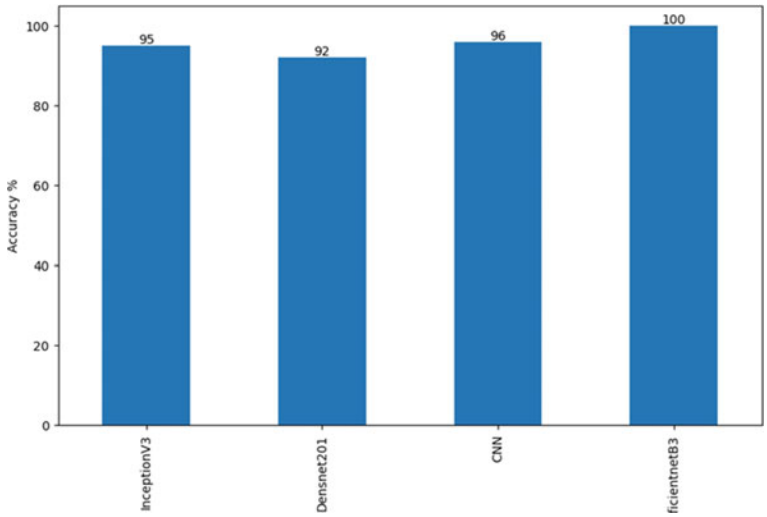


Fig. 4 All models' test accuracy

Table 2 Evaluation outcome

Model	Class	Precision	Recall	*F*1-score	Accuracy
InceptionV3	Normal	0.98	0.96	0.97	0.95
	Tuberculosis affected	0.83	0.89	0.86	
DenseNet201	Normal	0.96	0.95	0.95	0.92
	Tuberculosis affected	0.77	0.78	0.77	
CNN	Normal	0.97	0.98	0.98	0.96
	Tuberculosis affected	0.91	0.87	0.89	
EfficientNetB3	Normal	1.00	1.00	1.00	1.00
	Tuberculosis affected	1.00	1.00	1.00	

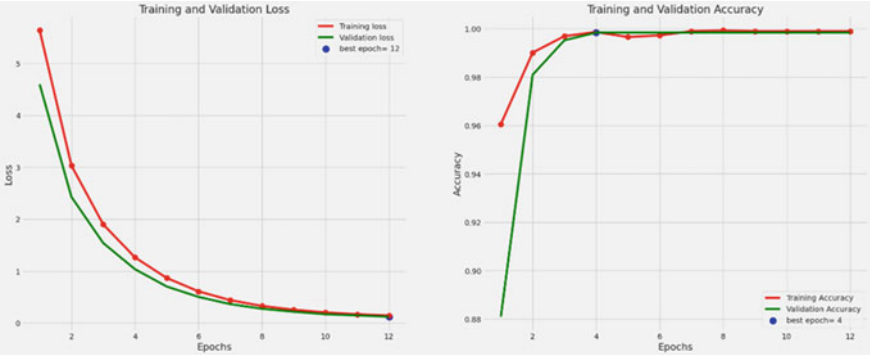


Fig. 5 Training and validation accuracy/loss (EfficientNetB3)

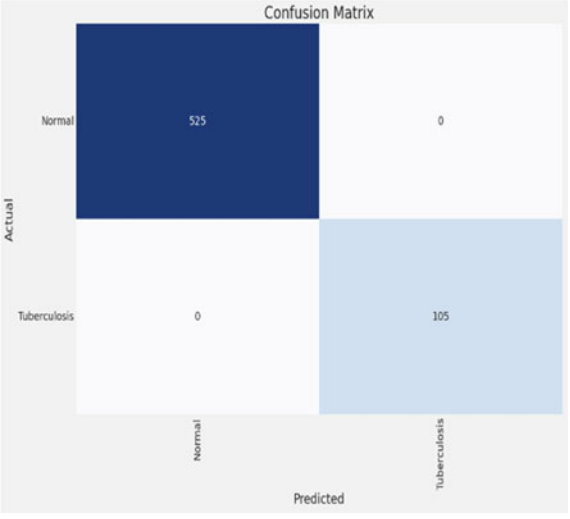


Fig. 6 EfficientNetB3 (confusion matrix)

4.2 Performance Evaluation

Software Specification: In order to execute this research, the Windows operating system and Google Colab were utilized. The Windows operating system was used for all of the project's experiments, which is equipped with an Intel(R) Core (TM) i7-10700 CPU clocked at 2.90 GHz. There are 8.0 GB of installed RAM on hand. The system type is a 64-bit operating system, and the CPU is an x64-based processor. On the other hand, Google Colaboratory is a powerful Python programming platform that provides Zero configuration, Free access to GPUs and Easy sharing. Users may create Python programs utilizing all the libraries on a platform provided by Google named Colab.

In order to do a proper evaluation, the output of this study has been compared with several state-of-the-art works. Table 3 gives the earlier studies that have used the same dataset, but different models for detecting TB, presenting a comparative analysis of these works.

Table 3 provides a comprehensive summary of various studies that have used deep learning models for medical image classification tasks. Mwendo [4] utilized deep transfer learning techniques for X-ray image classification. However, the provided information only states that the achieved accuracy was 92%, without specifying the model or approach used. Karaca [9] used the InceptionV3 and CNN models for X-ray image classification. The reported accuracy was 98.90%. Rahman [10] employed the DenseNet model for X-ray image classification and achieved an accuracy of 98.6%. Kieu [11] utilized VGG16 and InceptionV3 models for X-ray image classification. The reported accuracy was 89.77%. Acharya [13] employed Normalizing Flow Networks (NFNets) for X-ray image classification and achieved an accuracy of 96.91%. Tasci [14] utilized the CNN-VoPreCNNFT model for X-ray image classification and reported an accuracy of 97.69%. Huy [15] used a new deep learning model CBAMWDNet to Detect Tuberculosis, and they achieved 98.80% accuracy in their research. Table 3 gives that our model offers the maximum accuracy when compared to other models after comparing the suggested model with them. It achieved 100% accuracy which full filled our goal.

Table 3 Comparison with state-of-the-artwork

References	Model	Dataset	Accuracy (%)
Mwendo [4]	Deep transfer learning	X-ray images	92
Karaca [9]	InceptionV3, CNN	X-ray images	98.90
Rahman [10]	DenseNet	X-ray images	98.6
Kieu [11]	VGG 16, InceptionV3	X-ray images	89.77
Acharya [13]	NFNets	X-ray images	96.91
Tasci [14]	CNN-VoPre CNNFT	X-ray images	97.69
Huy [15]	CBAMWDnet	X-ray images	98.80
This paper	EfficientNetB3	X-ray images	100

5 Conclusion

This research on Tuberculosis Bacteria detection using deep learning techniques and image pre-processing concludes that deep learning techniques, combined with image pre-processing, can provide an accurate and effective way of detecting Tuberculosis Bacteria. In this research, four different deep learning models were used, and the outcomes demonstrated that EfficientNetB3 provides a maximum test accuracy of 100%, outperforming all the models that were implemented. Even when performing a comparative analysis with other state-of-the-art works, it was observed that EfficientNetB3 had the best performance. To increase the generalizability of the suggested approach, the paper recommends further study using larger datasets while being aware of the dataset's limitations.

References

1. Ramya R, Babu PS (2015) Automatic tuberculosis screening using canny edge detection method. In: Proceedings of the 2nd international conference on electronics and communication systems (ICECS), India, Feb 2015
2. Brooks JV, Orme IM (1998) Evaluation of once-weekly therapy for tuberculosis using isoniazid plus rifamycins in the mouse aerosol infection model. Antimicrob Agents Chemother 42(11):3047–3048
3. Overview Tuberculosis (TB) (2019) [online]. Available https://www.nhs.uk/conditions/tuberculosis-tb/NHS
4. Mwendo I, Gikunda K (2023) Deep transfer learning for detecting Covid-19, Pneumonia and Tuberculosis using CXR images—a review. arXiv preprint arXiv:2303.16754
5. Silverman C (1949) An appraisal of the contribution of mass radiography in the discovery of pulmonary Tuberculosis. Am Rev Tuberculosis: 466–482
6. Xpert MTB/RIF on stool is useful for the rapid diagnosis of Tuberculosis in young children with severe pulmonary disease [online]. Available https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5558052
7. Tuberculosis disease diagnosis based on an optimized machine learning model [online]. Available https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9041161/
8. Liu C, Alcantara M (2017) TX-CNN: detecting Tuberculosis in chest X-ray images using convolutional neural network. In: 2017 IEEE international conference on image processing (ICIP)
9. Karaca BK, Güney S (2021) Comparative study for Tuberculosis detection by using deep learning. In: 44th international conference on telecommunications and signal processing (TSP), pp 88–91
10. Rahman T, Khandakar A (2020) Reliable Tuberculosis detection using chest X-ray with deep learning, segmentation and visualization. IEEE Access: 191586–191601
11. Kieu S, Hwa T, Hijazi MHA (2019) Ensemble deep learning for tuberculosis detection using chest X-ray and canny edge detected images. IAES Int J Artif Intell: 429–435
12. Tuberculosis (TB) chest X-ray database [online]. Available https://www.kaggle.com/datasets/tawsifurrahman/tuberculosis-tb-chest-xray-dataset
13. Acharya V, Dhiman G (2022) AI-assisted tuberculosis detection and classification from chest X-rays using a deep learning normalization-free network model. Computat Intell Neurosci

14. Tasci E, Uluturk C (2021) A voting-based ensemble deep learning method focusing on image augmentation and preprocessing variations for tuberculosis detection. Neural Comput Appl 33(22):15541–15555
15. Huy VTQ, Lin C-M (2023) An improved Densenet deep neural network model for tuberculosis detection using chest X-Ray images. In: 2023 3rd international conference on intelligent communication and computational techniques (ICCT), vol 11, pp 42839–42849

An Enhanced Real-Time System for Wrong-Way and Over Speed Violation Detection Using Deep Learning



A. Manasa and S. M. Renuka Devi

Abstract Nowadays in many of the developing countries, traffic rule violation has become a critical issue. Due to the increase in population, the use of vehicles is also increasing exponentially, leading to an increase in road accidents. Major reasons for these accidents are traffic rule violations, such as over speed and wrong direction driving. To monitor these violations, an enhanced real-time system is needed which has the ability of traffic rule violation detections from nearby CCTV cameras. The proposed traffic signal violation detection system performs tasks like speed violation detection and wrong-way direction detection, without human intervention. The proposed system mainly performs three tasks: vehicle detection, vehicle tracking and violation detection. The YOLOv3 deep learning method is used for locating vehicles moving on road, and then the Kalman filter is used for tracking of detected vehicles. After tracking each vehicle is assigned with an unique id, then speed violation detection algorithm and wrong-way direction detection algorithm vehicles are used to identify the vehicles which violated the traffic rules. The system is very efficient and accurate to identify vehicles moving in the wrong direction and over speed.

Keywords YOLOv3 · Kalman filter · Speed violation · Wrong direction violation

1 Introduction

Due to rapid growth of population, usage of vehicles has also increased exponentially. This is a serious threat to the lives of people, due to the increase in road accidents. The number of such road accidents is equally proportional to traffic rule violations. A 2018 report of the Road Transport Ministry [1] shows that in India, the total number of road accidents has increased when compared with previous years. In the world record statistics [2] report, India is ranked in 1st place with the highest deaths

A. Manasa · S. . M. Renuka Devi (✉)
Department of ECE, G. Narayanamma Institute of Technology and Science (for women), Hyderabad, Telangana, India
e-mail: renuka.devi.sm@gnits.ac.in

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2023

S. Shakya et al. (eds.), *Fourth International Conference on Image Processing and Capsule Networks*, Lecture Notes in Networks and Systems 798,
https://doi.org/10.1007/978-981-99-7093-3_21

caused by road accidents. As per [3] 'Road Accidents in India 2021', there were about 410 k incidents of road accidents that took place in just a single year, with 150 k deaths and 380 k injured persons. Therefore, road accidents have become one of the main causes of death. This is worsened due to the increase in total vehicles on roads. Accidents mostly occur due to over speed violation and wrong direction of vehicles. To overcome this problem and to reduce such accidents, an enhanced real-time system is very important which uses deep learning algorithms for analyzing and detecting these violations effectively to decrease road accidents.

The main contribution of this paper is, the YOLOv3 model is trained on real images to classify and detect cars, bikes and humans. Unlike other research works, where multiple systems are needed for detection of multiple violations, this paper proposes a single system to detect multiple violations at once. The paper is organized in the following way: Sect. 1 has the introduction of the proposed model, Sect. 2 discusses literature survey and its summary. Section 3 discusses algorithms used in the proposed model. Section 4 shows results of proposed algorithms with respective figures. Section 5 briefly discusses the conclusion and future scope of this paper.

2 Literature Survey

Over many years, researchers have been working to develop algorithms that can detect various traffic rule violations. Mampilayil et al. [4] have proposed an automated video surveillance system which has the capability of detecting traffic rule violations automatically. The main focus is only on three wheeler vehicles. For detection Tensorflow with OpenCV is used and the detection of traffic violation is done by deep learning.

Bay dadaev et al. designed a model which detects vehicles violating the traffic rule [5]. The proposed model is mainly based on video processing. The deep learning method YOLOv3 is used for detection of cars. After estimating the location of a vehicle, Kalman filtering also known as linear quadratic estimation (LQE) is implemented for vehicle tracking. A unique method called 'entry-exit' algorithm is used which has 91.98% accuracy in wrong-way detection of vehicles.

Suttiponpisarn et al. introduced a system that detects and tracks vehicles from CCTVs. Using the RLB-CCTV algorithm, road lane boundaries are obtained with 91.6% accuracy, then vehicle detection is done by YOLOv4-Tiny, for tracking, Deep SORT is used [6]. It automatically detects vehicles driving in wrong direction using the MBCDD algorithm with 96.61% accuracy.

Rahman et al. designed a wrong-way vehicle detection system [7]. It mainly consists of three stages: YOLO, centroid tracking algorithm and the last stage is violation detection. The proposed system is effective in different lighting and different environmental conditions. The main limitation is the centroid tracking algorithm which causes switching of vehicle IDs due to overlapping of vehicles.

Suttiponpisarn et al. [8] introduced a model which detects and tracks the direction of vehicles moving on two-way roads, with the help of YOLOv4-tiny and Deep SORT.

The system first draws the region of interest and validates the vehicle direction giving very accurate results for detection of wrong direction vehicles.

R. J. Franklin et al. developed a system to detect traffic rule violations such as high vehicle speed and signal jump. It uses the YOLOv3 object detection technique. This system then improves accuracy by using the vehicle location and region of interest in between frames determining signal jump. This counts detected vehicles with accuracy of 97.67% and detects speed violations with 89.24% accuracy [9].

A. Sayar et al. developed a system which will detect the wrong-way violation [10]. For vehicle detection, support vector machines (SVMs) are used. The MedianFlow algorithm is used to track vehicles due to its high speed. After vehicle detection and tracking, center points of two consecutive frames are taken from the bounding box and are compared for violation detection.

Mallela et al. [11] presented a system which detects triple riding and the violation of speed limit. The YOLOv3 is trained on the COCO dataset to detect bike riders, and then it checks whether the rule is violated or not. The GSM module is used for data transferring. It is connected to the NodeMCU module for data extraction and transfers data to the cloud. This model has 91.7% accuracy.

All previous research paper works and the methodologies used are summarized into a single table for better understanding. It is given in Table 1.

Table 1 Summary for literature survey on different traffic rule violations

Authors and year of publication	Methodology	Type of violation detection	Type of vehicles	Accuracy (%)
Mampilayil et al. [4], 2019	Tensorflow with OpenCV	Wrong direction detection	Three wheelers	_
Sayar et al. [10], 2019	SVM	Security lanes and wrong-way violation	Cars	90
Rahman et al. [7], 2020	YOLO and centroid tracker	Wrong-way violation detection	Cars	_
Franklin et al. [9], 2020	YOLOv3	Speed violation and signal jump detection	All vehicles	89.24
Mallela et al. [11], 2020	YOLOv3	Speed violation, triple riding detection	Bikes	91.7
Bay dadaev et al. [5], 2020	YOLOv3, Kalman filter	Wrong-way detection	Cars, persons	91.98
Suttiponpisarn et al. [8], 2021	Yolov4-tiny and deepsort	Wrong-way detection	Bikes	96.1
Suttiponpisarn et al. [6], 2022	RLB-CCTV, MBCDD	Road lane, wrong-way detection	Cars	96.61

3 Project Methodology

In this proposed system first, each vehicle present in a given frame is detected and classified using the YOLOv3. For each detection of a vehicle, a bounding box is generated and fitted over it. Then, the information from these bounding boxes is given to the Kalman filter. Then, the Kalman filter predicts and estimates the location of each vehicle in its next frames and tracks are updated. After tracking, each vehicle is assigned with an unique id. The proposed system is further developed for speed violation detection and wrong-way direction detection using over speed violation detection (OSVD) algorithm and wrong-way traffic violation detection (WTVD) algorithm as shown in the block diagram in Fig. 1. The proposed system can detect these different violations using a single system.

3.1 YOLOv3 Algorithm

The YOLOv3 algorithm is a convolutional neural network (CNN), which is used for the detection of various different objects in real-time situations. It is used for the identification of particular objects in images and videos. To detect an object, it uses features of input images learned by using CNN [12]. For vehicle detection, YOLOv3 is implemented using the deep learning libraries OpenCV and keras. The Darknet-53 is used in YOLOv3 [13].

Object detection can be done using background subtraction and optical flow, but they show false positive detections when there are several vehicles moving very close to each other. Also these methods are very sensitive in low light conditions resulting in inaccurate models. To overcome these drawbacks, YOLOv3-416 algorithm is used which has processing speed of 29 ms and mean Average Precision (mAP) of 55.3% on COCO dataset.

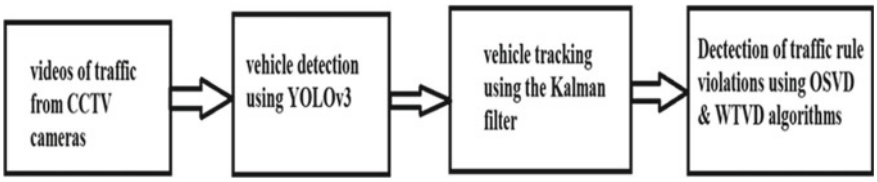


Fig. 1 Block diagram of proposed model

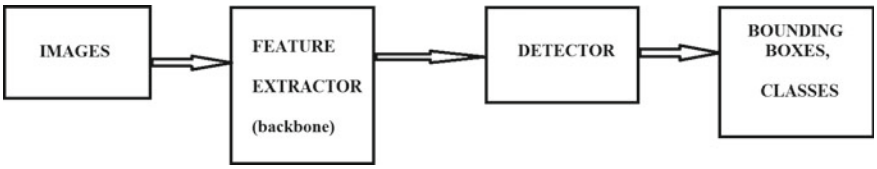


Fig. 2 Basic block diagram of YOLOv3 deep learning network

3.2 Working of YOLOv3

The convolutional layers of YOLOv3 architecture learn various important features of given input images and are then given to a regression for detection. These learned features have the bounding boxes coordinates, class label, bounding boxes sizes, dimensions and many more. In YOLOv3, each cell of the prediction map predicts only a certain number of bounding boxes. YOLOv3 has three anchors in particular and it predicts three bounding boxes per neuron. Sometimes, more than one bounding box detects the object as positive detection, which leads to a single object to be detected multiple times. To avoid this situation, non-maximum suppression is used which will pass detections only if they haven't been detected previously. Non-maximum suppression is an imperative part of utilizing the YOLOv3 algorithm effectively. Anchor boxes and non-maximum suppression values are few of the features that make the predictions of an object possible. Every bounding box prediction has five components, namely: (a, b, w, h, c). The (a, b) are centroid coordinates of the box, w and h are width and height and c is confidence score. Block diagram of YOLOv3 is given in Fig. 2.

3.3 YOLOv3 Network Architecture

The YOLOv3 network architecture consists of elements such as residual blocks, skip connections and upsampling. The CNN layers are followed by batch normalization and leaky ReLU. For detection, YOLOv3 uses Darknet-53 along with additional 53 layers stacked onto it, giving a total of 106 layers of fully convolutional architecture as shown in Fig. 3. It does not have any pooling layers (max pooling loses small features from the input). Instead of pooling, additional CNN layers with stride 2 are used to downsample the input image, which prevents loss of low level features that are useful in detecting small objects.

The input size for YOLOv3 is 416×416 as it has a better map and better processing speed. YOLO version3 makes predictions at 3 different layers 82, 94 and 106 layers, which are used for detecting large, medium and small-sized objects, respectively.

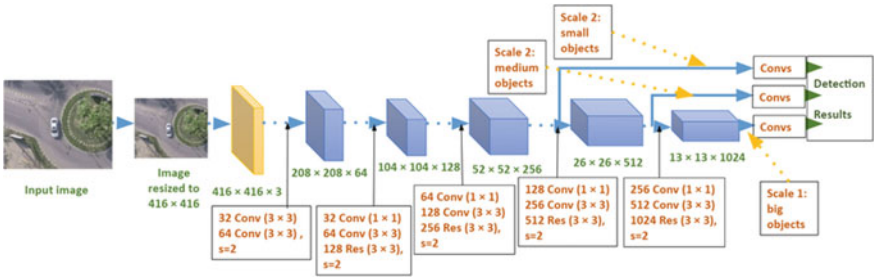


Fig. 3 YOLOv3 network architecture [14]

3.4 Kalman Filter

In order to be able to recognize traffic violations, after vehicle detection and drawing bounding boxes, tracking of moving vehicles must be done within a certain period of time. The process of vehicle tracking has two stages: In the first stage, every moving car is detected from each frame then in the second stage, new detections of the same car are associated with it over time. To track each detected vehicle, the Kalman filter is being used. The Kalman filter is powerful and fast compared to other tracking algorithms. Therefore, it is used for tracking the detected vehicles. It predicts the vehicle location in each frame and assigns them to its respective track.

The YOLOv3 model gives the bounding box information and from these the centroids of each bounding box are extracted and given to Kalman filter which is primarily based on centroid tracking algorithm. For every detection an unique ID is assigned. The Hungarian algorithm [15] is used to assign each detection to its corresponding unique ID. In the next step, Kalman filter equations are used to predict multiple vehicles. The Kalman filter has mainly two phases: (1) Prediction phase and (2) Correction phase. In the prediction phase, the Kalman filter predicts the state at the next step. And after receiving measurement, it corrects or updates the prediction of the current state. It runs in loop for further frames and updates the assigned tracks and deletes lost tracks till the end of the video. The formulas for both phases are given below.

1. Prediction phase

State estimation:

$$e_k = Se_{k-1} + Cu_k \tag{1}$$

Error covariance:

$$q_k = Sq_{k-1}S^T + p \tag{2}$$

2. Correction phase

Gain:

$$G_k = q_k O^T \left(O q_k O^T + R\right)^{-1} \tag{3}$$

State estimation update:

$$E_k = e_k + G_k(M_k - O e_k) \tag{4}$$

Error covariance update:

$$Q_k = (1 - G_k O) q_k \tag{5}$$

where e_k and E_k are state estimations in prediction and correction phase respectively, S is state transition matrix, C is the control matrix, u_k is the variable input, q_k and Q_k are the error covariance in prediction and correction phase, respectively, p is process noise covariance, O is observation matrix, R is measurement covariance, M_k is measurement vector and G_k is the Kalman gain.

3.5 Wrong-Way Traffic Violation Detection Algorithm

Wrong-way direction is one of the leading causes of deaths in India. If a vehicle is moving in the opposite direction of traffic flow, then it is called a wrong-way traffic violation. The algorithm used to detect such violation is named as wrong-way traffic violation detection (WTVD). After vehicle detection and tracking, direction of each vehicle is determined. In order to determine the direction of each vehicle first, the pixel difference of that particular vehicle is calculated once for every 5 frames. To calculate this difference, the centroids from each detected bounding box are extracted. Initial centroids of a vehicle are taken from the frame F_1 and are denoted as (A_1, B_1). Centroids from frame F_5 are taken as (A_5, B_5), here for every 5th frame, the difference is calculated for faster direction estimation. Figure 4 shows the movement of cars throughout the video frames.

The pixel difference is calculated as $\mathrm{d}A = A_5 - A_1$. If $\mathrm{d}A$ is negative and A_5 is smaller than A_1, the car is moving from right to left and has direction as WEST. For $\mathrm{d}B = B_5 - B_1$, if $\mathrm{d}B$ is negative and B_5 is smaller than B_1, the car is moving from top to bottom and has direction as NORTH. In case of horizontal roads, WEST is considered positive and in case vertical roads NORTH is considered as positive. Otherwise, it is considered as a traffic rule violation as they are moving in opposite directions of intended traffic flow of vehicles. The vehicles violating the traffic flow are detected by the algorithm which then highlights those vehicles as WRONG DIRECTION.

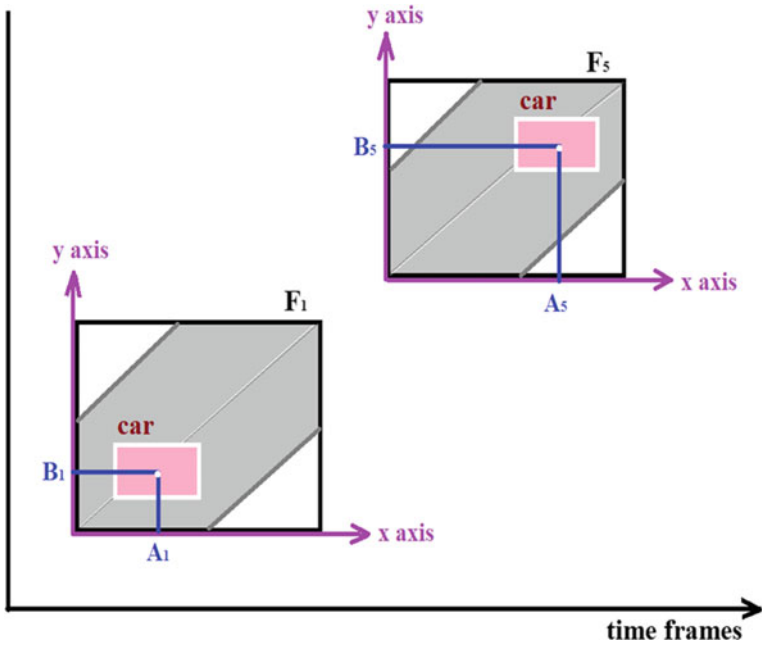


Fig. 4 Cars moving through the frames with centroids (A_1, B_1) and (A_5, B_5)

3.6 Over Speed Violation Detection Algorithm

Another main cause of deaths due to road accidents is vehicles moving with very high speed. The velocity is calculated by using the centroids of the vehicles. After detecting and tracking, each vehicle is fitted with a bounding box. Then, the centroids of each vehicle (A_1, B_1) and (A_2, B_2) are extracted and used to calculate the velocity using below mentioned formulae. Initially, the velocity is calculated in terms of pixels which is then converted into km/h. This is done during object tracking. The over speed violation detection (OSVD) algorithm is used to calculate and estimate the speed of every detected vehicle in the current frame. Therefore, the OSVD algorithm calculates velocity of each vehicle and checks whether the speed is below speed limit Th_1 or not. The traffic rule violation is detected when any vehicle is moving with speed exceeding predefined speed limit Th_1 which is 60 km/h, Th_1 is assumed to be 60 km/h in this paper. The velocity is calculated using the below formulae.

$$\text{Velocity} = \text{d_met} * 3.6 * \text{fps} \tag{6}$$

$$\text{d_pix} = \text{sqrt}\left((A_2 - A_1)^2\right) + \left((B_2 - B_1)^2\right) \tag{7}$$

$$\text{d_met} = \text{d_pix/ppm} \tag{8}$$

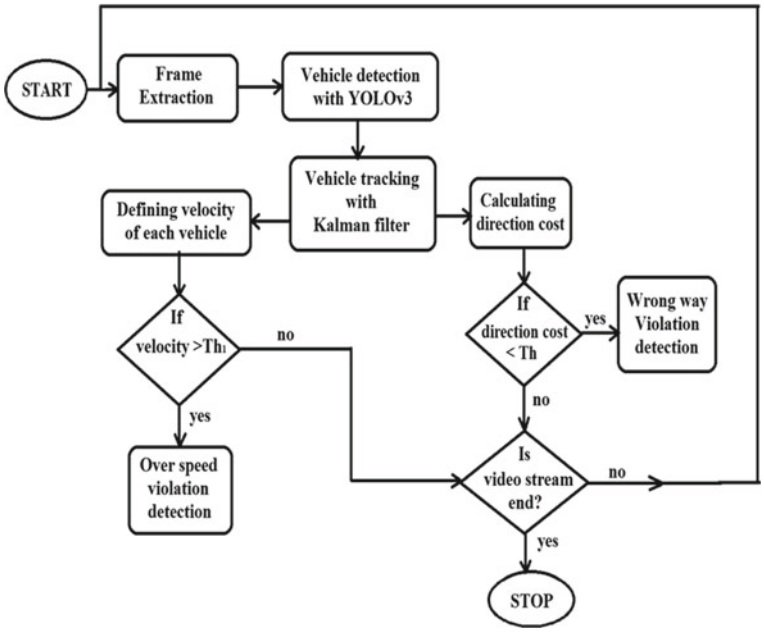


Fig. 5 Flowchart of traffic violation detection system

where ppm is pixel per meter, fps is frames per second, 3.6 is the factor used to convert speed into km/h, d_met is pixel to meter conversion metric, (A_1, B_1) and (A_2, B_2) are centroids.

The flowchart of the proposed model is as shown in following Fig. 5., which starts with extraction of frames from real-time input videos after which an array of frames is given to the YOLOv3 algorithm which detects the 3 classes and fits bounding boxes. Then, the Kalman filter uses this information and tracks the vehicles, and for every vehicle, direction is calculated and velocity is calculated. If the direction of vehicle is opposite to intended traffic flow or if velocity is more than threshold, then they are considered as violations and such vehicles are detected at that input frame. The system then repeats itself until the end of all frames.

4 Experimental Results

To obtain a dataset of traffic videos, different traffic videos are taken from different locations. These videos are of resolution 1920 * 1080 with 30 fps. Then, all frames are extracted from every video and split into a 7:3 ratio and used as training dataset and testing dataset. The YOLOv3-416 model is then trained with real world data from the created dataset. It is mainly trained to detect and classify persons, cars and bikes, in which the input frames are resized to image pixels 416 * 416 with 120

epochs. It took around six hours in total to finish training the custom model with 3 classes. The python software with Tensorflow and OpenCV libraries are used to implement the proposed model.

4.1 Vehicle Detection and Tracking

For vehicle detection, YOLOv3-416 is used which detects and classifies cars, persons and bikes. The algorithm then uses bounding boxes to be fitted over each detection. The trained YOLOv3-416 model is used over new real-time videos which are test videos.

After the detection of vehicles, the information of bounding boxes is given to the Kalman filter which is used for tracking of vehicles. The results of vehicle detection and tracking are shown in Fig. 6, where persons, cars and bikes are detected and bounding boxes are fitted over it.

4.2 Wrong-Way Violation Detection

For wrong-way detection, initially the moving direction of each vehicle is estimated and then vehicles which are moving in the west direction and north direction, i.e., are considered correct and it is considered to be a violation, if it moves in the east direction (left to right) and south direction (top to bottom). Vehicles violating the rule are saved into the result folder. Figure 7 shows the vehicles with traffic violations detected and highlighted as WRONG DIRECTION by the algorithm, where the vehicles moving in the EAST direction and SOUTH direction are wrong.

4.3 Over Speed Violation Detection

The velocity of each vehicle is calculated by using the velocity formula. These estimated velocities of every vehicle are displayed as shown in Fig. 8. Since all vehicles are moving within speed of 60 km/h, there is no violation detected.

Any vehicle which is moving with velocity more than a predefined limit of 60 km/h is considered as violation and are highlighted as ‘overspeed’ as shown in Fig. 9. The video frames in which vehicles violating the traffic rule are detected will be saved automatically into the result folder.



Fig. 6 Detection and tracking of cars, bikes and persons

5 Conclusion

The proposed traffic signal violation detection system performs tasks like speed violation detection and wrong-way direction detection, without human intervention. The proposed system mainly consists of three stages: vehicle detection, vehicle tracking and violation detection. The YOLOv3 deep learning method is used for the detection of vehicle locations, and then the Kalman filter is used for tracking of detected vehicles. After tracking, each vehicle is assigned with an unique id, then speed violation detection algorithm and wrong-way direction detection algorithm vehicles are used to identify the vehicles which violated the traffic rules. The system is very efficient and accurate to identify vehicles moving over speed and in the wrong direction.

The proposed model has an accuracy of 88.23% in detecting vehicles moving in the wrong direction and accuracy of 91.42% in detecting vehicles with over speed.

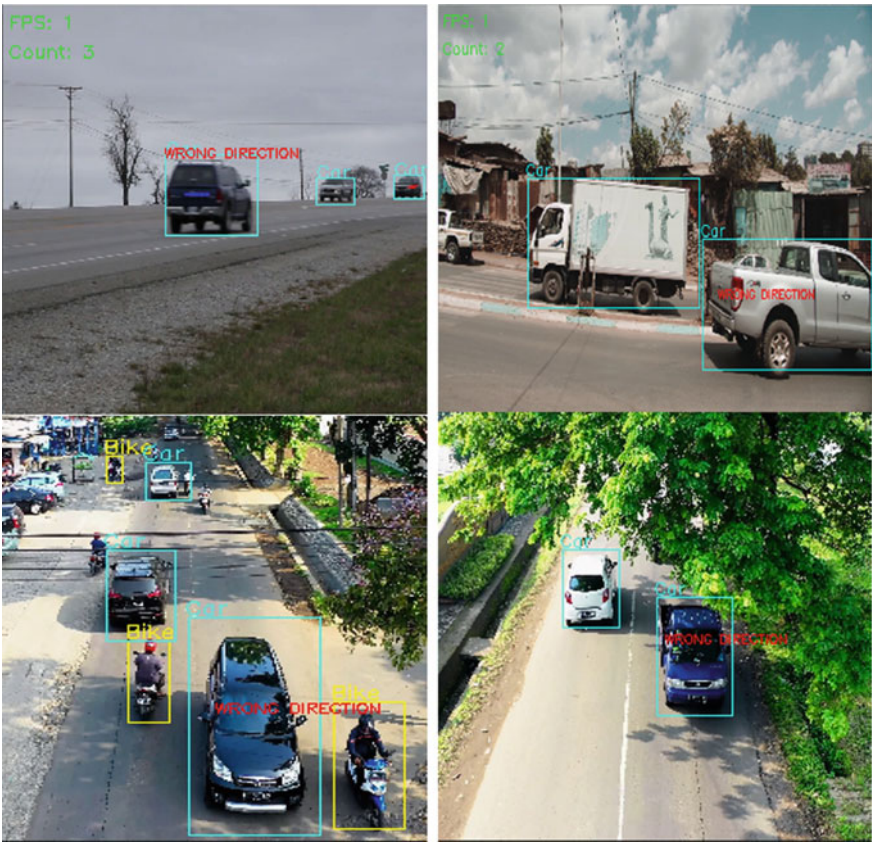


Fig. 7 Vehicles moving in wrong direction are detected using Kalman filter

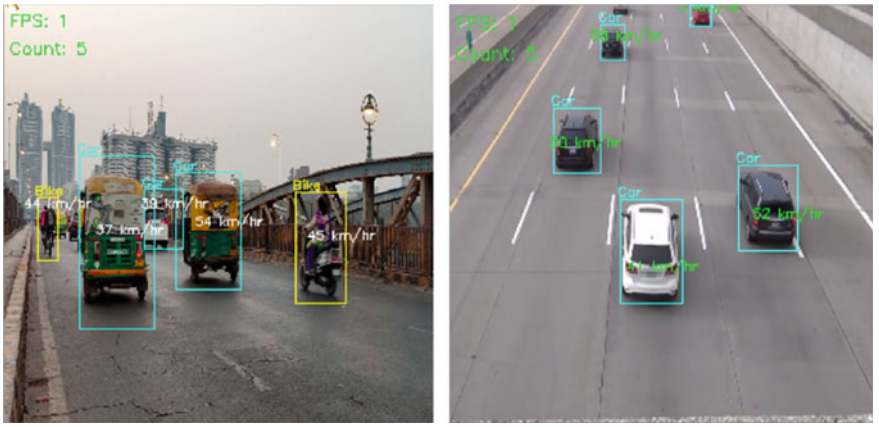


Fig. 8 Speed estimation of vehicles

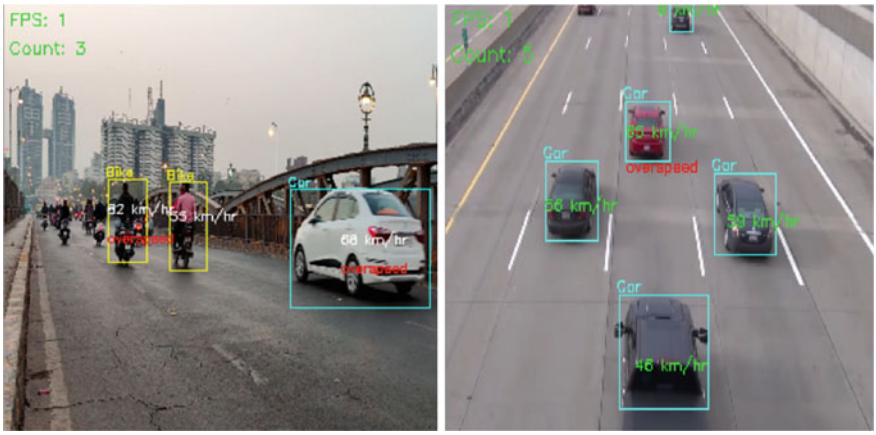


Fig. 9 Detection of vehicles moving with over speed using Kalman filter

The proposed system can detect only two types of traffic violations. This can further be improved by adding more traffic violations that lead to road accidents.

References

1. Ministry of Road Transport & Highways, Government of India. https://morth.nic.in/sites/default/files/Annual-Report-2010-2011.pdf
2. Global status report on road safety 2018. https://www.who.int/publications-detail-redirect/9789241565684
3. Ministry of Road Transport & Highways, Government of India. https://morth.nic.in/sites/default/files/RA_2021_Compressed.pdf
4. Mampilayil HR, Rahamathullah K (2019) Deep learning based detection of one way traffic rule violation of three wheeler vehicles. In: International conference on intelligent computing and control systems (ICCS), Madurai, India, pp 1453–1457
5. Usmankhujaev S, Bay Dadaev S, Woo KJ (2020) Real-time, deep learning based wrong direction detection. Appl Sci 10(7):2453
6. Suttiponpisarn PS, Charn Sripinyo C, Usanavasin S, Nakahara H (2022) An enhanced system for wrong-way driving vehicle detection with road boundary detection algorithm. Procedia Comput Sci 204:164–171
7. Rahman Z, Ami AM, Ullah MA (2020) A real-time wrong-way vehicle detection based on YOLO and centroid tracking. In: IEEE region 10 symposium (TENSYMP), Dhaka, Bangladesh, pp 916–920
8. Suttiponpisarn P, Charnsripinyo C, Usanavasin S, Nakahara H (2021) Detection of wrong direction vehicles on two-way traffic. In: 13th international conference on knowledge and systems engineering (KSE), Bangkok, Thailand, pp 1–6
9. Franklin RJ, Mohana (2020) Traffic signal violation detection using artificial intelligence and deep learning. In: 5th international conference on communication and electronics systems (ICCES), Coimbatore, India, pp 839–844
10. Şentaş A, Kul S, Sayar A (2019) Real-time traffic rules infringing determination over the video stream: wrong way and clearway violation detection. In: International artificial intelligence and data processing symposium (IDAP), Malatya, Turkey, pp 1–4

11. Mallela NC, Volety R, Srinivasa Perumal R et al (2021) Detection of the triple riding and speed violation on two-wheelers using deep learning algorithms. Multimed Tools Appl 80:8175–8187
12. Sumalatha R, Varaprasada Rao R, Renuka Devi SM (2021) Pothole detection using YOLOv2 object detection network and convolutional neural network. In: Advances in intelligent systems and computing applied information processing systems, pp 293–300
13. Redmon J, Farhadi A (2018) YOLOv3: an incremental improvement. arXiv:1804.02767v1
14. Ammar A, Koubaa A, Ahmed M, Saad A, Benjdira B (2021) Vehicle detection from aerial images using deep learning: a comparative study. Electronics 10:820. https://doi.org/10.3390/electronics10070820
15. Kuhn HW (1955) The Hungarian method for the assignment problem. Naval Res. Logist Q 2:83–97

U-Net-Based Denoising Autoencoder Network for De-Speckling in Fetal Ultrasound Images



S. Satish, N. Herald Anantha Rufus, M. Antony Freeda Rani, and R. Senthil Rama

Abstract Nowadays medical imaging plays an important role in the detection of defects before proceeding with any medical procedures. During pregnancy, medical imaging plays a major role in the detection of any abnormalities/improper development of fetal organs during the gestational period with the help of ultrasound images. As ultrasound images are feeble toward noise interference, it makes difficult for proper intervention for the physician. In this paper, we modeled a U-Net-based denoising autoencoder network to reduce the speckle noise in fetal ultrasound images. To study the performance analysis of U-Net-based denoising autoencoder network, peak signal-to-noise ratio and structural similarity index measure parameters are used. The parameters are analyzed based on different dropout values, different epoch values, and noise levels ($\sigma = 1, 2, 3, 5, 10$). It is inferred from the analysis that PSNR = 28.48 dB, SSIM = 90.87%, accuracy = 98.40%, and loss = 0.54 from the proposed network. The proposed network model performs well than other de-speckling techniques.

Keywords Ultrasound images · Speckle noise · Image restoration · Autoencoder

S. Satish (✉) · N. Herald Anantha Rufus
Department of Electronics and Communication Engineering, Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, Chennai, Tamil Nadu, India
e-mail: satishsaiece@gmail.com

N. Herald Anantha Rufus
e-mail: drrufus@veltech.edu.in

M. Antony Freeda Rani
Department of Electrical and Electronics Engineering, DMI College of Engineering, Chennai, Tamil Nadu, India

R. Senthil Rama
Department of Electrical and Electronics Engineering, DMI College of Engineering, Nagercoil, Tamil Nadu, India

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2023

S. Shakya et al. (eds.), *Fourth International Conference on Image Processing and Capsule Networks*, Lecture Notes in Networks and Systems 798,
https://doi.org/10.1007/978-981-99-7093-3_22

1 Introduction

According to the echo imaging concept, ultrasound scanners produce ultrasound images (UI), when body tissues are exposed to the scanner's pulsed acoustic pulses [1, 2]. Ultrasound is a medical imaging technique that uses high-frequency sound waves to create images of internal body structures. It is a valuable tool in medical diagnostics, and it is a non-invasive, safe, and provides real-time images. It does not use ionizing radiation, making it ideal for use in pregnancy and for diagnosing a range of medical conditions. The image produced by the ultrasound can provide information about the shape, size, outline, and consistency of organs and tissues. It can be used to diagnose a range of conditions, such as gallstones, kidney stones, liver disease, pregnancy, and fetal development.

UI contains multiplicative and Gaussian-free noise. UI was typically much more challenging to eliminate the incremental noise because the concentration of the noise differs with the strength of the image. Generally, the speckle image consists of image pixel and Gaussian-free noise. For the majority of noise multiplicative applications, it is believed that the noise level is stationary, with an unidentified noise variance and a unified mean σ^2. The speckle image undergoes a logarithmic transformation y_{ij} [3] noise that transforms from multiplicative to additive conversion. After that, adding zero mean Gaussian noise is an approximation for the noise component [4]. Even though the quality of images is decreased by speckle noise, denoising ultrasound images before the diagnosis is essential to prevent negative effects on the diagnosis process. A few years ago, there has been a lot of interest in de-speckling ultrasound images.

2 Existing Methods

Mean filter is the most effective at denoising Gaussian, speckle, and Poisson noise, but it has a high blurriness value. The mean filter produces a blurring effect. Koo et al. [5] helps to identify the proper speckle region, the homogeneous region growing mean filter, a method for adaptive image filtering that assumed a region that is more speckle than average. If the center point is determined, a homogeneous region based on similarity and homogeneity can be found. With any degradation, the edges are preserved here.

Lee Filter [6] uses a local average filter to minimize speckles while maintaining the overall composition of the image. In ultrasound imaging, the Lee filter is employed in reducing speckle noise and raising image quality. This filter employs a weighted average to reduce image noise by employing a low-pass filter (LPF). It is an adaptive filter, which means that depending on the local characteristics, different weights are applied to different parts of the image. The filter operates by selecting a window centered on a specific element and calculating a measure of the average elements in just that period. The weight of each pixel is then determined by a function that takes

the window's standard deviation into account. The amount of the central elements is then changed to all the weighted averages of which window. Each element in the image is subjected to the same procedure, yielding a filtered image with less speckle noise. An anisotropic diffusion algorithm was proposed by Perona et al. [7], which has since been improved by numerous research works [8]. The PMD filter reduces speckles by preserving image edges, lines, and regions. The PMD filter is comparable to Gaussian blurring. The PMD equation is reduced to the Heat equation if the diffusion coefficient is persistent. PMD filters can be used to detect image boundaries in some cases.

Kaur et al. [9] medical images typically have low noise and contrast because of the circumstances and situations in which they are recorded, and they have a very low level of clarity. Contrary to certain other medical images that are free of noise and objects, a wide-range ultrasound image is a challenging process because edge fuzziness can also be present in these images.

Speckle noise is persistently increasing and has irregular sound waves that constantly change, according to Benazarti et al. [10]. As a result, the authors decided that the logarithmic transformation was the best option for converting signal-dependent noise to additive noise rather than just multiplicative noise. The dispersion tensor is employed in this study to implement adipose dissemination to use the class in a structured way for boosting attributes of the image, acclimating flow dispersion in the way of the native orientation. The effectiveness of the algorithm has been amply demonstrated by the authors.

Prudhvi et al. [11, 12] implemented multi-scale transforms to create denoising image enhancement filtration. According to the results of the de-hazing process, spatial domain techniques smooth out boundary lines and line segments in ultrasound images. The softening process was reduced by the researchers using wavelet transform-based techniques, but the borders were unaltered. Although the wavelet transform had a good performance, its computation complexity was more. The couple complex tree curvelet morphs and surpass intensity both tree curvelets transform excel at stripping away erratic noise versus extracting features and multi-scale morphs.

De Fontes et al. [13] created a powerful real-time noise removal model for ultrasound images that was an advancement over the non-local standard-based NL-means technique, i.e., a pixel's restored intensity must be mean that is full because the length between the patches was adequate for all pixels in the image [14, 15]. The outcomes show that this technique can be applied to real-time medical prognosis and is efficient for noise removal. For intermediate images, this model's calculation time was under 20 ms.

Ranjani et al. [16] suggested denoising ultrasound images using the multi-tree complicated curvelet transform-based Levy Shrink algorithm (DTCWT). In their study, where the DTCWT coefficient of all thread are displayed using a considerably tracked Levy distribution, the authors used this algorithm to remove speckle noise. They determined the items of the Levy issuance using a shuttered expression by attempting to manipulate the noise-containing ultrasound image's minute observation-based moments, as well as the minute the Levy and Gaussian instants

distributions. Finally, the coefficients were determined using a Bayesian MAP estimation model.

Wen et al. [17] used a wavelet analysis and fuzzy theory-based technique for image enhancement and noise removal. Ultrasound images were subjected to the correlation transformation, conversion of nonlinear into noise emulsifiers, and inter wavelet transform to obtain higher and lower frequency wavelets. The curvelet soft limit denoising method improved the higher frequency coefficients, while the fuzzy technique improved the lower frequency correlation coefficient.

Hota et al. [18] introduced a noise reduction technique for noise contraction in ultrasound images that are based on strain computing. The strain computing technique makes use of image decorrelation in various conditions such as tissue movement. Speckle patterns differ from one another when a similar item is studied in different environments or under different conditions. Due to the even out of subsequent frames, speckle noise has been contained. The noise reduction will be robust, as the correlation between the two frames is small.

Adam et al. [19] demonstrate an argumentative region growing-based filtration besides the withdrawal of speckling noise in medical images. Here, correlation value is used to categorize each region. The computation mean filter defines the correlation sections, while the pixel edge-designated classes are defined by the computation mean filter. However, it has a major flaw in that it takes longer to compute because it detects different fields by using the province expanding method.

A filter for speckle noise reduction based on dimensional extracts was proposed by Jiang et al. [20]. The above-derived filter is constructed by determining each regional sub-mask's minimal native direction derivative. The filtration produces the best outcomes when the demographic variance on the directive's ward is the lowest. The poor de-speckling pointed picture corner requires more computational effort, which is the filter's only drawback.

A noise-reducing filter was introduced by Rui et al. [21] and operates on the ultrasound medical image's 2-D lowest mean square. The weighted local dynamics are specified by the 2-D least mean square method, which is based on the local arrangement of speckle noise. The error essentially controls the filter output within the weighted and ideal local dynamics. Speckle reduction using the 2-D minimum mean square speckle-noise filter preserves a lot of information. The above-derived filtration is constructed by determining each regional sub-mask's minimal native direction variant. When the demographic variance on the ward of the directive is the least, the filtration yields best results [22, 23]. The poor de-speckling pointed picture corner requires more computation time, which is the filter's only drawback.

Gajpal et al. [24] introduced a speckle noise-reducing algorithm on the non-local process. This is, in essence, a Gaussian filter. The filtering task was carried out based on the object's high degree of correlation. In the field, the filter works on pixel compression. The Gaussian noise model performs admirably. The Gaussian filter, on the other hand, does not explicitly apply to the speckle noise image.

Yu et al. [25] demonstrated the contiguous nonlinear filtering methods connected to produce a speckle noise-reducing filter. Anisotropic and Gaussian filters are used as nonlinear filters. The filters were analyzed separately in the first step, and then

by nonlinear filtering. In the second step, the filters are evaluated by running both methods. Using this filter will improve the quality of denoised images.

3 Proposed Method

3.1 U-Net-Based Denoising Network

A U-Net-based denoising network is a type of neural network that can be used for image denoising are shown in Fig. 1. It consists of two parts: an encoder and a decoder. The encoder takes the noisy input image and maps it to a lower-dimensional representation, while the decoder maps this representation back to the original image. The encoder and decoder are connected through a bottleneck layer, which is the compressed representation of the input image. The encoder section consists of two convolution layers: a relu layer and a dropout layer, and a normalization layer which comprises a block along with a max pooling layer. The decoder sections consist of blocks and transpose convolution layers. In this section during the decoding process, convolution transpose is used before each block to perform denoising of the image from noisy image.

3.2 U-shaped Dilated Convolution Denoising Autoencoder Network

U-Net-based dilated convolution denoising autoencoder network combines the function of U-Net-based denoising and dilated convolution-based denoising autoencoder network as shown in Fig. 2. It consists of two parts: an encoder and a decoder. The encoder takes the noisy input image and maps it to a lower-dimensional representation, while the decoder maps this representation back to the original image. The encoder sections consist of dilated convolutional layer, dropout layer, normalization layer, and max pooling layer, whereas the decoder section consists of dilated convolutional layers and max pooling layers.

3.3 U-Net-Based Denoising Autoencoder Network

Modified U-Net-based dilated convolution denoising autoencoder network gives rise to a network called as U-Net-based denoising autoencoder network which consists of two parts: an encoder and a decoder as shown in Fig. 3. The encoder section consists of a block which comprises of a convolutional layer, relu, dropout layer, and normalization layer takes the noisy input image and maps it to a lower-dimensional

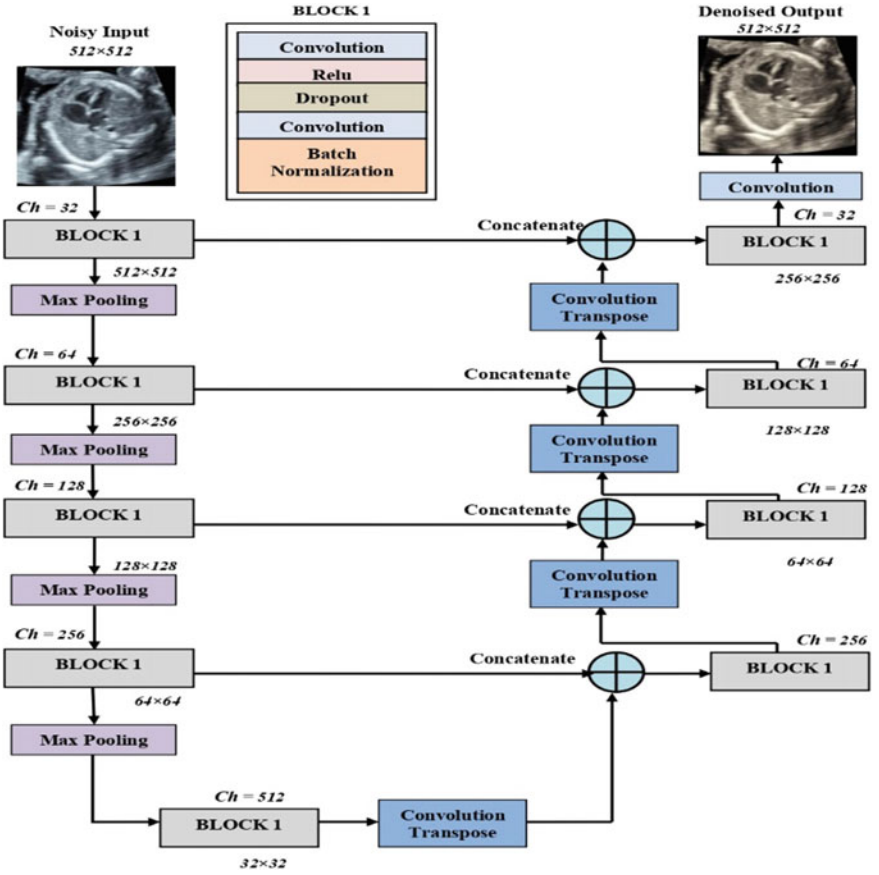


Fig. 1 U-Net-based denoising network

representation. The convolution layer is used for feature extraction operations. This procedure causes the image size to be reduced by the convolution window. The decoder section consists of the same blocks as used in the encoder section along with the up-pooling layer.

By pairing the shrinking image, the up-pooling layer restores its dimensions. In this network structure, transpose convolution is used only once. The structural characteristics of the layer's shrinking image size are restored with little distortion thanks to the transposed convolution process, which is the reverse of the convolution process.

Subsampling layers are used in CNN-based networks to form processes for the image content. Down sampling layers disregard the image resolution while allowing the receptive field to grow. Despite being used to maintain invariance, manage to overfit, and perform feature extraction, pooling layers, a type of down-sampling

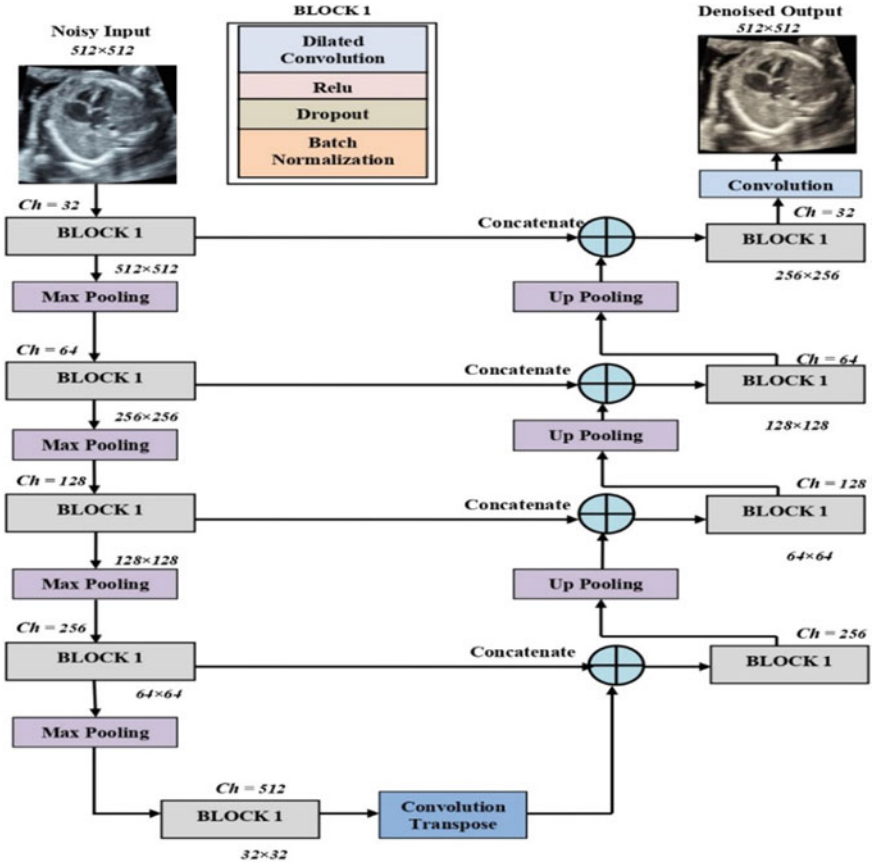


Fig. 2 U-shaped dilated convolution denoising autoencoder network

layer, also result in a reduction in the resolution of the images. However, down-sampling layers cause the resolution of features across a network's layers to gradually deteriorate. Small objects that are challenging to recover even with efforts involving skip connections or other techniques may be overlooked by the resulting coarse features. The resolution cannot be fixed by up-sampling layers, which are used to restore the images [26]. For applications involving image denoising, the resolution is a crucial consideration.

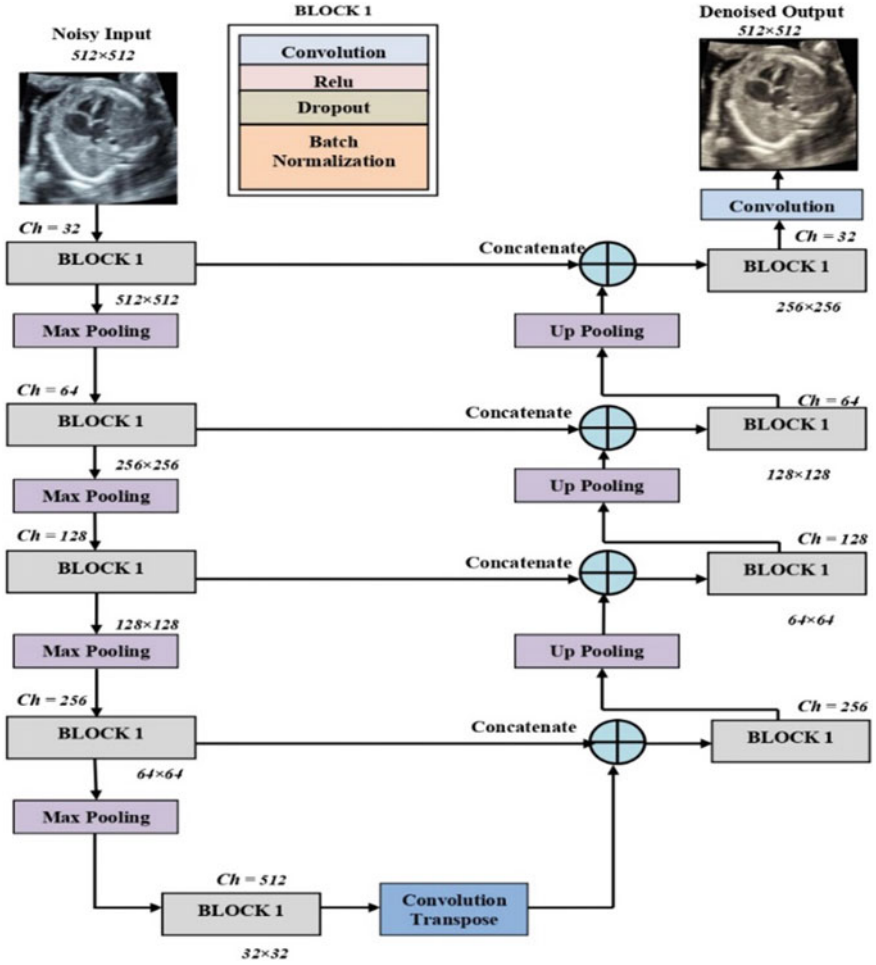


Fig. 3 U-shaped denoising autoencoder network

4 Result and Discussion

Here, the different deep learning networks are used such as U-Net-based denoising network (UDN), U-Net-based dilated convolution denoising autoencoder network, and U-Net-based denoising autoencoder network. For the above deep learning networks, different phenomena are investigated and discussed. The different phenomena are the following: (1) effect of dropout, (2) effects on accuracy and loss, and (3) effects on different noise levels for the different datasets.

Structural Similarity Index Method

The similarity measure can be used to evaluate the quality of a denoised image if the original image is of great quality. The SSIM index is determined using

$$\text{SSIM} = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{x\overline{x}} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \tag{1}$$

μ_x, μ_y are the mean of x and $\overline{x}$, respectively, σ_x and σ_y are the variances of x and $\overline{x}$, respectively, $\sigma_{x\overline{x}}$ the correlation between x and $\overline{x}$, c_1 and c_2 are fixed values.

Peak Signal-to-Noise Ratio

The PSNR approximates how well a reconstruction compares to the denoised image in terms of human perception. Decibels are used to measure the ratio of these two images.

$$\text{PSNR} = 10\log_{10}\frac{(\text{PV})^2}{\text{MSE}} \tag{2}$$

4.1 Dataset

Dataset is collected from the ultrasound clinical laboratory. For the study based on clinical data, the 1250 clinical fetal cardiac images are used. A total of 625 noisy training images, 625 clear comparison images, and 5 different σ levels of test images make up the second dataset's image set. The first 125 of the 625 noisy ultrasound images have a standard deviation of $\sigma = 1$, the second 125 have a standard deviation of $\sigma = 2$, the third 125 have a standard deviation of $\sigma = 3$, the fourth 125 have a standard deviation of $\sigma = 5$, and the fifth 125 have a standard deviation of $\sigma = 10$. The dataset's pixels measure 512×512. Here, the performance of different criteria is analyzed for the different denoising networks.

4.2 Adding Speckle Noise

Speckle noise is a common problem in ultrasound imaging and can affect image quality, making it difficult to interpret and diagnose. Fortunately, there are several approaches to reducing or suppressing speckle noise from ultrasound images. Speckle noise in US images exhibits a multiplicative behavior that is strongly correlated with non-Gaussian statistics. It multiplies into the underlying US signal reflected by the tissue, resulting in a value that is directly proportional to the local gray level in that area. As a result, speckle noise is image data-dependent, resulting

in quality degradation in the form of contrast reduction, blurry details, and local pseudo features. Speckle noise is inherent in US images and can be mathematically modeled using statistical distributions for multiplicative scattering. As a result, traditional Gaussian-based image analysis methods (e.g., filtering and segmentation) are insufficient.

$$g(x, y) = (f(x, y) \times \varphi_m(x, y)) + \varphi_a(x, y) \tag{3}$$

In Eq. (3), the original image is expressed as $f(x, y)$, $g(x, y)$ refers to the noisy image and the multiplicative and additive noises are referred as $\varphi_m(x, y)$ and $\varphi_a(x, y)$. x and y are the variables' spatial positions. In general, additive noise is ignored. Rayleigh distribution characterizes speckle patterns created for completely random scattering.

Assuming the Rayleigh distribution of the expression $\varphi_m(x, y)$ in Eq. (4), the probability density function of $\varphi_a(x, y)$ is expressed as follows:

$$p(f|u) = \frac{f}{u\sigma^2}\exp(-\frac{f}{2u^2\sigma^2})(\varphi \geq 0) \tag{4}$$

The noise intensity is symbolized by the parameter σ, the noise-free image by the parameter u, and the noise-added image by the parameter f. In our study, five different speckle noise standard deviations $\sigma = 1, 2, 3, 5$, and 10 noise levels are used for the performance analysis of different deep learning-based denoising networks. The results of the different deep learning-based denoising networks are examined concerning the effects of the hyperparameters. Table 1 also displays the hyperparameter values of the chosen denoising networks.

Table 1 Denoising network hyperparameters

Hyperparameter	UDN	U-DCDAN	U-DAN
No. of layers	50	54	44
No. of max pooling layer	4	–	4
Convolution	–	–	–
Mini-batch size	20	20	20
Validation rate (%)	10	10	10
Dropout rate (%)	10	10	10
Activation function	Sigmoid	Sigmoid	Sigmoid
Optimizer	Adamax	Adamax	Adamax
Learning rate	0.0001	0.0001	0.0001

4.3 Effect of Dropout

The effect of dropout is investigated, and peak signal-to-noise ratio (PSNR) and structural similarity index method (SSIM) are analyzed using the following graphs as shown in Fig. 4. In batch normalization for PSNR, dropout value unto 0.6 the PSNR value lies between 28.15 and 28.16 dB after further increasing the value of dropout till 0.9 the PSNR values reduce abruptly unto 27.54 dB for all the values of σ. In batch normalization for SSIM, the dropout value unto 0.6 values forms a "U-"shaped structure after further increasing the dropout value to 0.9 the SSIM value reduces abruptly between 70 and 50 for all values of σ. In batch renormalization for PSNR, the dropout value unto 0.4 PSNR value lies between 28.18 and 28.16 dB after further increasing the value of dropout to 0.9 the PSNR value reduces gradually between 28.16 and 26.60 dB. In batch renormalization for SSIM, the dropout value unto 0.4 SSIM value falls between 98 and 90% and further increasing the dropout value till 0.9 the SSIM value reduces gradually between 90 and 50% for all values of σ.

In PSNR, batch normalization dropout value is constant from dropout = 0 (σ = 1, PSNR = 28.15 dB; σ = 2, PSNR = 28.15 dB; σ = 3, PSNR = 28.15 dB; σ = 5, PSNR = 28.15 dB; σ = 10, PSNR = 28.15 dB) to dropout = 0.6 (σ = 1, PSNR = 28.14 dB; σ = 2, PSNR = 28.14 dB; σ = 3, PSNR = 28.14 dB; σ = 5, PSNR = 28.14 dB; σ = 10, PSNR = 28.14 dB) and after dropout = 0.6, PSNR values gradually decrease till 0.9 for batch normalization (σ = 1, PSNR = 27.78 dB; σ = 2, PSNR = 27.74 dB; σ = 3, PSNR = 27.70 dB; σ = 5, PSNR = 27.66 dB; σ = 10, PSNR = 27.54 dB).

In SSIM, batch normalization for dropout = 0 (σ = 1, SSIM = 76.82%; σ = 2, SSIM = 75.12%; σ = 3, SSIM = 76.22%; σ = 5, SSIM = 76.78%; σ = 10, SSIM = 71.77%) and for increasing the dropout value SSIM value reduce gradually till dropout = 0.4 (σ = 1, SSIM = 66.88%; σ = 2, SSIM = 61.56%; σ = 3, SSIM = 59.20%; σ = 5, SSIM = 64.34%; σ = 10, SSIM = 55.58%) and increases gradually till dropout = 0.6 (σ = 1, SSIM = 69.50%; σ = 2, SSIM = 70.38%; σ = 3, SSIM

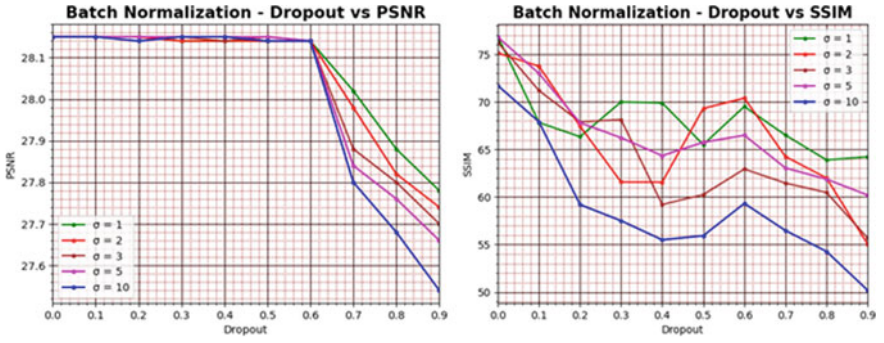


Fig. 4 Effect of dropout

= 62.93%; $\sigma = 5$, SSIM = 66.48%; $\sigma = 10$, SSIM = 59.32%) in the form of "U-"shaped structure. After dropout = 0.6, SSIM values gradually decreases till 0.9 for batch normalization ($\sigma = 1$, SSIM = 64.21%; $\sigma = 2$, SSIM = 55.05%; $\sigma = 3$, SSIM = 55.72%; $\sigma = 5$, SSIM = 60.21%; $\sigma = 10$, SSIM = 50.21%).

Finally, it is inferred that the dropout value for batch normalization lies between 0 and 0.6. The dropout value should lie between the suggested ranges for the process of effective denoising of fetal cardiac (FC) ultrasound images.

4.4 Effects of Accuracy and Loss

The effects of accuracy and loss are investigated and analyzed for different epoch values are shown in Fig. 5. It is inferred from the graph that the trade-off between epoch and accuracy, as the epoch increases accuracy also increases steadily and after reaching a particular epoch value the accuracy value becomes constant due to the decoding section of the U-Net-based denoising autoencoder network.

The effect of accuracy is investigated as follows:

- For U-DCDAN, until 400 epoch the accuracy values are steadily increasing (92.47%), further unto 800 epochs slightly increased (94.27%) and after further epoch increases the accuracy becomes constant at 94.51%.
- For U-DAN, unto 150 epochs the accuracy values are steadily increasing (97.56%), further unto 800 epochs slightly increased (98.40%), and after further epoch increases the accuracy becomes constant at 98.53%.
- For UDN, unto 400 epoch the accuracy values are steadily increasing (87.74%), further unto 900 epochs slightly increased (91.25%), and after further epoch increases the accuracy becomes constant at 91.74%.

The effect of loss is investigated as follows:

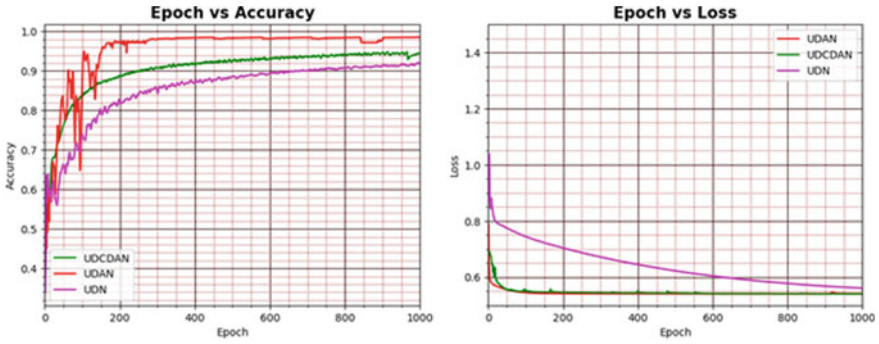


Fig. 5 Effects on accuracy and loss

- For UDN, unto 40 epochs the loss value gradually decreases from 0.78 to 0.56 and after further epoch value increases loss becomes constant at 0.54.
- For U-DAN, unto 110 epochs the loss value gradually decreases from 0.69 to 0.55 and after further epoch value increases loss becomes constant at 0.54.
- For U-DCDAN, unto 850 epochs the loss value gradually decreases from 1.04 to 0.58 and further epoch value increases loss becomes constant at 0.56.

From the above investigation and analysis, U-DAN produces higher accuracy and lower loss when compared to UDN, and U-DCDAN.

4.5 Effects on Different Noise Levels

Figure 6 shows the visual results of Tables 2, respectively. Figure 6 shows that the proposed deep learning-based denoising networks can remove speckle noise from images and improve image resolution. In the proposed denoising model, neural network models learn to reconstruct the original data from its noisy counterpart so as to clean out noise from corrupted or noisy data. A NN model's performance is based on a nonlinear activation function, which is used to backward propagate, update the weights, and generate the desired outcome, which is the real output. The model may successfully retrain itself to make adjustments for the discrepancy between the actual and desired output. The sigmoid derivative has a complicated processing capacity and always returns zero as x grows. Because of this, the gradient is zero, and when additional layers are multiplied, the gradient also becomes zero, which leads to poor model performance. We had utilize sigmoid activation function in the final output layer for classification issues similar to the autoencoder in tensorflow.

Adamax maintains a moving average of the gradient's first and second moments, which are the mean and variance of the gradients, respectively. The moving average of the second moment, which is similar to the RMSProp term, helps the optimizer to scale the learning rate for each parameter based on the variance of the gradients. This helps to improve the optimization algorithm's performance in the early stages of training. It has the ability to converge quickly and handle noisy or sparse gradients. It helps to accelerate the training of deep neural networks and improve their performance.

5 Conclusion

For de-speckling, a variety of filters and techniques are employed. By boosting the capabilities of the computer processors, various deep learning techniques are used here to accelerate. In this study, speckle noise in ultrasound images is reduced using U-Net-based denoising autoencoder network (U-DAN), with greater applications as computer technologies advance. The proposed network performance is compared in

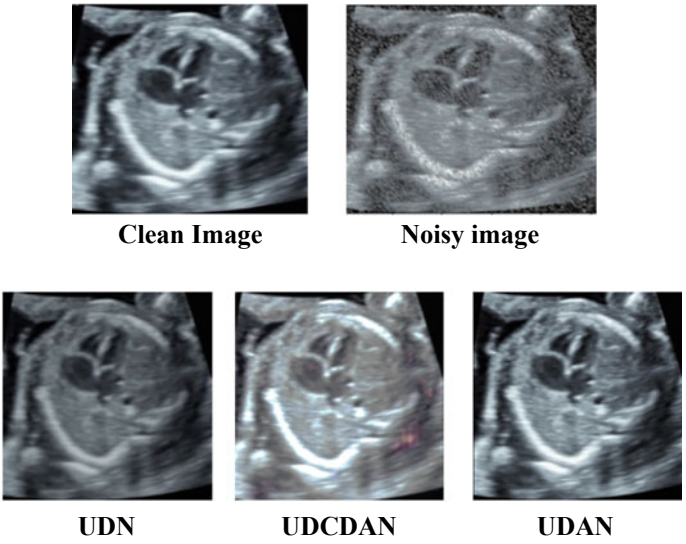


Fig. 6 Effectiveness of the proposed deep learning-based denoising method for clinical test image at $\sigma = 2$

Table 2 Performance of different denoising networks according to PSNR and SSIM metrics for the clinical dataset

Methods	PSNR					SSIM				
	$\sigma = 1$	$\sigma = 2$	$\sigma = 3$	$\sigma = 5$	$\sigma = 10$	$\sigma = 1$	$\sigma = 2$	$\sigma = 3$	$\sigma = 5$	$\sigma = 10$
Noisy image	24.68	23.98	21.86	20.25	19.64	86.65	83.54	63.18	55.86	46.58
UDN	28.46	29.12	28.75	27.56	26.05	87.42	86.52	85.14	83.52	80.24
U-DCDAN	29.14	28.58	28.03	27.06	26.58	89.68	87.26	85.49	83.64	81.76
U-DAN	29.48	29.67	29.02	28.74	28.48	90.87	88.62	86.28	84.16	82.27

terms of the PSNR and SSIM. The U-DAN achieves the best PSNR values at 2, 3, and 5 noise levels when the deep learning-based denoising methods are compared with each other. At all noise levels, the U-DAN network has the best SSIM values. This study demonstrates that the suggested deep learning networks are the best techniques for removing noise from ultrasound images. Future research will look at the impact of using a different distribution method than an equal distribution to distribute noise levels.

References

1. Gautam R, Bharti R (2018) Liver ultrasound image enhancement using bilateral filter. Int J Eng Techn Res
2. Binaee K, Hasanzadeh RPR (2014) An ultrasound image enhancement method using local gradient based fuzzy similarity. Biomed Signal Process Control 13:89–101
3. Zong X, Laine AF, Geiser EA (1998) Speckle reduction and contrast enhancement of echocardiograms via multiscale nonlinear processing. IEEE Trans Med Imaging 17:532–540
4. Michailovich OV, Tannenbaum A (2006) De-speckling of medical ultrasound images. IEEE Trans Ultrason Ferroelectr Freq Control 53(1):64–78
5. Koo JI, Park SB (1991) Speckle reduction with edge preservation in medical ultrasonic images using a homogeneous region growing mean filter (HRGMF). Ultrasonic Imaging 13(3):211–237
6. Lee J-S (1980) Digital image enhancement and noise filtering by use of local statistics. IEEE Trans Pattern Anal Mach Intell PAMI-2 2:165–168
7. Perona P, Malik J (1990) Scale-space and edge detection using anisotropic diffusion. IEEE Trans Pattern Anal Mach Intell 12(7):629–639
8. Zong X, Laine AF, Geiser EA (1998) Speckle reduction and contrast enhancement of echocardiograms via multiscale nonlinear processing. IEEE Trans Med Imaging 17(4):532–540
9. Foment J (2014) Parameter-free fast pixel-wise non-local means denoising. Image Processing Line 4:300–326
10. Kaur GS, Kaur P (2016) Image enhancement of ultrasound images using multifarious denoising filters and GA. In: 2016 international conference on advances in computing, communications and informatics, Jaipur, pp 2375–2384
11. Benzarti F, Amiri H (2012) Speckle noise reduction in medical ultrasound images. IJCSI Int J Comput Sci 9(2, 3)
12. Prudhvi Raj VN, Venkateswarlu T (2012) Denoising of medical ultrasound images using spatial filtering and multiscale transforms. Int J Comput Sci Inf Technol (IJCSIT) 4(6)
13. Andria G, Attivissimo F, Cavone G, Giaquinto N, Lanzolla AML (2012) Linear filtering of 2-D wavelet coefficients for denoising ultrasound medical images. Measurement 45:1792–1800
14. De Fontes FPX, Barroso GA, Hellier P (2010) Real-time ultrasound image denoising. J Real-Time Image Process 6(1):15–22
15. Kishore PVV, Mallika KL, Prasad MVD, Narayana KL (2015) Denoising ultrasound medical images with selective fusion in wavelet domain, second international symposium on computer vision and the internet. Procedia Comput Sci 58:129–139
16. Ranjani J, MS C (2015) Bayesian denoising of ultrasound images using heavy-tailed Levy distribution. IET Image Process 9(4):338–345
17. Farouj Y, Freyermuth JM, Navarro L, Clausel M, Delachartre P. Hyperbolic wavelet-Fisz denoising for a model arising in ultrasound imaging. IEEE Trans Comput Imaging 99:1
18. Fredj H, Malek J (2016) Real-time ultrasound image denoising using NVIDIA CUDA. In: 2016 2nd international conference on advanced technologies for signal and image processing (ATSIP), Monastir, pp 136–140
19. Wen H, Qi W (2015) Enhancement and denoising method of medical ultrasound image based on wavelet analysis and fuzzy theory. In: Seventh international conference on measuring technology and mechatronics automation, Nanchang, pp 448–452
20. Hota NR, Sanibigrah S (2018) Retrieval of random valued impulse noise with advanced weighted mean filter by using modified denoising algorithm for image processing application. Asian J Sci Technol 09(07):8444–8460
21. Adam D, Beilin-Nissan S, Friedman Z, Behar V (2006) The combined effect of spatial compounding and nonlinear filtering on the speckle reduction in ultrasound images. Ultrasonics 44(2):166–181
22. Xie J, Jiang Y, Tsui HT, Heng PA (2006) Boundary enhancement and speckle reduction for ultrasound images via salient structure extraction. IEEE Trans Biomed Eng 53(11):2300–2309

23. Rui L, Zhuoxin S, Cishen Z (2008) Adaptive filter for speckle reduction with feature preservation in medical ultrasound images. In: 2008 10th IEEE international conference on control, automation, robotics and vision, pp 1787–1792
24. Gajpal DK, Biswas AK, Potdar RM (2012) Reconstruction of arteries in heart image affected by specular reflection. Int J Emerg Technol Adv Eng 2:259–263
25. Yu J, Tan J, Wang Y (2010) Ultrasound speckle reduction by a SUSAN-controlled anisotropic diffusion method. Pattern Recogn 43(9):3083–3092
26. Hamaguchi R, Fujita A, Nemoto K, Imaizumi T, Hikosaka S (2018) Effective use of dilated convolutions for segmenting small object instances in remote sensing imagery, March, 1442–1450

Galo Lexical Tone Recognition Using Machine Learning Approach



Bomken Kamdak, Gom Taye, and Utpal Bhattacharjee

Abstract In this paper, we present the development of a Galo speech corpus and the design of a Galo lexical tone recognizer. The Galo language is spoken by the Galo people residing in Arunachal Pradesh, a frontier state of North East India. Galo belongs to the Tani branch of the Tibeto–Burman language family. Galo exhibits two discernible lexical tones: High/Plain, characterized by a moderately high-level pitch contour and Low/Tense, characterized by a descending pitch contour. The speech database used in this study contains Galo phrases spoken by 27 individuals, comprising 13 male and 14 female speakers aged between 20 and 50 years. The recording of each speaker encompasses a phonetically diverse script consisting of twenty Galo tonal words and all their possible tonal variations. The work aims to automatically recognize these lexical tones by employing seven features extracted from the fundamental frequency (F0) contour. Two classification models based on machine learning, specifically support vector machine (SVM) and random forest (RF), were employed to identify lexical tones. The RF-based recognizer achieves a recognition accuracy of 90.42%, while the SVM-based recognizer surpasses that with an accuracy of 92.31%.

Keywords Galo language · Tone recognition system · F0 · Support vector machine · Random forest

B. Kamdak · G. Taye · U. Bhattacharjee (✉)
Department of Computer Science and Engineering, Rajiv Gandhi University, Itanagar, Arunachal Pradesh 791112, India
e-mail: utpal.bhattacharjee@rgu.ac.in

B. Kamdak
e-mail: bomken.kamdak@rgu.ac.in

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2023

S. Shakya et al. (eds.), *Fourth International Conference on Image Processing and Capsule Networks*, Lecture Notes in Networks and Systems 798,
https://doi.org/10.1007/978-981-99-7093-3_23

1 Introduction

Automatic speech recognition research (ASR) has made substantial progress since its inception in the mid-twentieth century, rendering it a viable option for facilitating human–machine communication. Lots of research has already been conducted on speech recognition to make it a modality for human–machine communication, driven by the fact that speech is the most successful form of human communication [1]. Speech is a complex communication medium consisting of various modalities, wherein the majority of information is attributed to linguistic content. In addition, speech signals contain a variety of informational aspects, such as the identity of the speaker, their physical and psychological state, and geographical and cultural background, etc. The primary objective of an automatic speech recognition (ASR) system is to employ computational methods to transform speech signals into a sequence of linguistic entities, commonly represented as a series of words or phonemes. ASR has been an exciting study area for more than 60 years, but there is still much room for improvement, especially in challenging environments where human outperforms their machine counterpart in speech recognition tasks [2].

The accurate identification of the lexical tone associated with a syllable is crucial in tonal languages to ensure a precise interpretation of word meaning. While resourceful tonal languages such as Mandarin have made significant advancements in automatic tone recognition, the presence of tones remains a challenge for achieving precise speech recognition in the majority of tonal languages, which lack comprehensive literature and documentation regarding tone patterns and their associations with syllables. This difficulty in tone recognition hinders the advancement of automatic speech recognition systems for these languages. The identification and characterization of tones linked to a spoken expression using standard speech features can significantly enhance the speech recognition capabilities of tonal languages.

This paper provides a study of the Galo language, which is a tonal language spoken by the Galo tribe residing in the Arunachal Pradesh of North East India. The Galo tribe holds considerable importance within the context of Arunachal Pradesh, primarily concentrated in the central region of the state. The Galo language is classified as a member of the Tani branch within the Tibeto–Burman language family. There is very little information available for this low-resource, poorly studied language. Two tones are found in the Galo language: High/Plain, which has a medium high-level tone, and Low/Tense, which has a dropping tone [3]. The central focus of this research paper revolves around the use of a concise set of features obtained from the F0-contour in order to represent the Galo lexical tone. Following this, machine learning methods, namely support vector machine (SVM) and random forest (RF), are used to recognize the tones.

The subsequent sections of this paper are organized in the following manner. Section 2 provides an overview of the existing literature pertaining to the recognition of tone. The methodology and model are explicated in Sect. 3. The presentation of the Results and Discussion can be found in Sect. 4. Section 5 provides a comprehensive overview of the Conclusion and Future Works.

2 Related Works

A tonal language refers to a linguistic framework where the distinction of pitch or tone plays a crucial role in establishing the semantic significance of words. In contrast to non-tonal languages, which primarily employ variations in pitch to convey emotional or grammatical nuances, tonal languages integrate pitch distinctions as an inherent element of their lexical and phonetic systems. Considerable investigation has been undertaken regarding the recognition of tonal speech, with particular emphasis on languages that possess a rich abundance of linguistic resources, such as Mandarin and Cantonese. The study carried out by Chen et al. [4] centered on the identification of Mandarin tones. The researchers used Hidden Markov Modeling in conjunction with a delta modulation technique to extract features from the pitch sequence. The tone recognition of isolated Mandarin monosyllables was conducted by Chang et al. [5] using the multi-layer perceptron (MLP) model. Feature vectors were constructed by extracting ten features from the fundamental frequency and energy contours of monosyllables. The results of their study revealed a recognition accuracy rate of 93.8% for untrained speakers. The recognition of tone is of utmost importance in the identification of Cantonese, a language widely recognized for its extensive tonal variations. In their study, Lee et al. [6] employed suprasegmental feature parameters that were extracted specifically from the voiced segment of monosyllabic utterances. A three-layer feedforward neural network was employed for the purpose of feature vector classification. The study utilized a phonologically comprehensive lexicon consisting of 234 unique syllables. This led to recognition accuracies of 89.0% in single speaker situations and 87.6% in multi-speaker scenarios. The authors Lee et al. [7] introduced a robust methodology for accurately identifying explicit tones in continuously spoken Cantonese utterances. The methodology employed entailed the extraction of tone feature vectors at a brief temporal interval, with the purpose of capturing the patterns of fundamental frequency (F0) and energy fluctuations at the syllable level. The authors proposed the implementation of a moving-window normalization technique in order to address the impact of tone-irrelevant fluctuations in F0 and energy features. The study involved the utilization of Hidden Markov Models to conduct context-dependent acoustic modeling of distinct tones. The findings of the experiment revealed that the accuracy of tone recognition in the speaker-independent scenario was 66.4%. In their study, Peng and Wang [8] utilized support vector machines (SVMs) as a means to identify tones in the context of continuously spoken Cantonese speech. An adaptive log-scale 5-level F0 normalization technique was proposed in order to address the influence of F0 variations that are not related to tone. A proposed extension of this normalization method incorporates the consideration of intonation. The findings from the experiment revealed that the accuracy of tone recognition in a speaker-independent context was 71.50%. In their study, Wang and Levow [9] put forth a novel approach aimed at improving tone recognition. Their method involved integrating vowel landmark detection into the modeling of the tone nucleus. The frames exhibiting the greatest sonority during the production of vowels were designated as vowel landmarks. The researchers utilized a support vector machine that was

trained on Mel-frequency cepstral coefficient (MFCC) features in order to identify and classify these specific landmarks. The results of the tone recognition experiments demonstrated that the landmark-based approach, which specifically targets the tone nucleus regions, outperformed the best baseline system by a margin of more than 6%. In a preliminary investigation, it was discovered that relying solely on vowel landmark evidence for tone nucleus regions resulted in a slight reduction of accuracy, amounting to less than 2%, when compared to utilizing both landmark frames and force-aligned vowel boundary information. Based on the aforementioned findings, the researchers arrived at the conclusion that the utilization of vowel landmarks to extract features from the most sonorous section of the tone nucleus is adequate for achieving precise recognition of the tonal quality of a syllable. In their study, Hsieh et al. [10] devised a technique to represent Taiwanese tones. This involved employing a piece-wise curve fitting algorithm on the F0-contour. F0-contour of each syllable was divided into four segments, and each segment was separately fitted to a curve. Based on their research, it was found that using feature extraction from curve-fitted F0 values yields higher accuracy in tone recognition compared to features extracted from raw F0-contour. The researchers employed a SVM-based classifier for the purpose of identifying and distinguishing the various tones in the Taiwanese language. In their research, Gogoi et al. [11] presented a system that utilizes support vector machines (SVM) to differentiate between three distinct lexical tones, namely High, Mid, and Low, in the Ao language. The distribution of tones in trisyllabic words was examined by the researchers, using a sample of 36 individuals. The F0-contour was derived using zero-frequency filtering (ZFF), followed by the computation of four acoustic features based on the F0-contour. The Ao language is comprised of three discrete dialects, namely Changki, Mongsen, and Chungli. To facilitate the identification of tones in different dialects, distinct tone recognition systems were utilized, employing SVM as the underlying methodology. The accuracy rates for identifying Ao tone in the Changki, Mongsen and Chungli dialects were reported as 53.20%, 62.26%, and 52.69%, respectively. In an independent investigation, Gogoi et al. [12] endeavored to develop an automated system for the recognition of four phonological tones in the Mizo language. The researchers employed acoustic-prosodic parameters as the foundation for their feature extraction methodology. In order to generate input for two classifier models, namely support vector machine (SVM) and deep neural network (DNN), six features were computed from the fundamental frequency (F0) contours. The aforementioned models were employed for the purpose of conducting tone recognition tasks. The database utilized for assessing the system comprises 31,950 instances of syllables, encompassing the four distinct tones that are present in the Mizo language. The aforementioned instances were gathered from a subset of 19 individuals who were instructed to utter trisyllabic phrases for the purpose of analysis and subsequent incorporation into the database. The researchers observed that the classifier based on deep neural network (DNN) achieved a recognition accuracy of 74.11% in the identification of the four phonological tones in Mizo. When comparing the two classifiers, it was observed that the SVM-based classifier achieved a recognition rate of 73.39%, indicating a slight improvement of 0.72% for DNN-based recognizer.

3 Methodology and Model

The proposed framework for the Galo tone recognition system is given in Fig. 1.

3.1 Galo Speech Database Creation

The Galo language exhibits the presence of both monosyllabic and disyllabic words with tones. Table 1 illustrates a selection of Galo tonal words. High/plain tone is depicted without any accompanying tone letter, whereas Low/Tense tone is denoted by a (‘) tone letter positioned at the beginning of the word.

The speech corpus used in the study comprises data obtained from 27 individuals, with 13 being male and 14 being female speakers. The speakers belong to the Galo

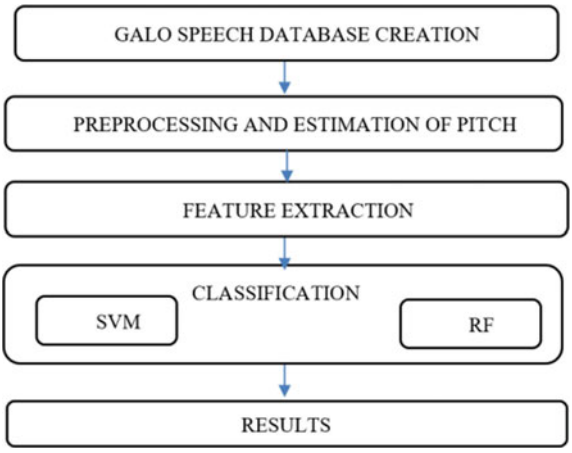


Fig. 1 The proposed framework for Galo tone recognition system

Table 1 Example of Galo tonal words

Sl no	Galo word	English meaning
1	*aii*	Teeth
2	*‘aii*	Heavy
3	*axi*	Elder sister
4	*‘axi*	Number two
5	*onam*	Pluck
6	*‘onam*	Falling something from a height
7	*vtww*	Herd of cattle
8	*‘vtww*	Bamboo grove
9	*umnam*	Humid weather
10	*‘umnam*	Boil something in the water

tribe, which is indigenous to Arunachal Pradesh. The individuals are of the age group 20 to 50 years. The audio recording was conducted within a controlled acoustical setting, employing a sampling frequency of 16 kHz and a mono format with a 16-bit resolution. The participants are provided with a phonetically rich written Galo script, consisting of twenty Galo tonal words with all their tonal instances. Each speaker is recorded eight times, divided into two equal recording sessions. These sessions are separated by a minimum of one week. Consequently, a comprehensive database consisting of 17,280 different tonal words was generated. The subjective evaluation of the database involved the participation of five individuals from the Galo tribe who possess limited prior knowledge in the field, as well as three listeners who have received phonetic training. Speech samples assessed correctly by at least 60% of the naïve audience and a minimum of two specialists have been deemed suitable for subsequent analysis. The final database comprises a total of 14,623 tonal words. The experiments described in this paper were conducted using this database.

3.2 Preprocessing and Estimation of Pitch

A first-order digital filter with a transfer function given in Eq. (1) is used to pre-emphasize the digitized speech.

$$H(Z) = 1 - 0.95z^{-1} \tag{1}$$

Subsequently, an energy-based voiced activity detection (VAD) algorithm is employed to identify the voiced segment of the speech signal and discard the initial and final silence period. The voiced speech signal is divided into frames of 30 microseconds. These frames are overlapped by one-third of their duration. F0 is derived from the individual frames of the speech signal. To reduce the influence of speaker dependencies on fundamental frequency (F0), *Z*-score normalization has been applied. The *Z*-score normalization can be accomplished using the following equation:

$$x^* = \frac{x - \mu}{\sigma} \tag{2}$$

where μ and σ are the mean and standard deviation of the F0 values, respectively. A smoothing algorithm is applied to smooth the F0-contour [13].

3.3 Feature Extraction

The Galo tones are represented by seven features extracted from the smoothed pitch contour of Galo words. The features extracted from pitch contour are mean F0,

median F0, standard derivative of F0, F0 value measured at 0, 25, 75 and 100% of the way through the contour. The F0 value at 0% is denoted as F0 onset and F0 at 100% is denoted as F0 offset. These seven features extracted from the pitch contour constitute the feature vectors for the classifiers to classify the two categories of Galo tones.

3.4 Classification

In order to classify the features derived from the F0-contour of the Galo syllables, two machine learning techniques, namely support vector machine (SVM) and random forest (RF), were employed.

3.4.1 Support Vector Machine (SVM)

Vapnik was the first to propose the support vector machine (SVM), a statistical machine learning method [14]. Since its debut, it has attracted a lot of interest and been widely used across numerous fields. The ideal hyperplane, a final surface that optimizes the margin between the classes in the given dataset, is used by the SVM algorithm to divide the classes. Support vectors were initially used to refer to the data points within these classes that were closest to the hyperplane. The SVM statistical learning algorithm's primary goal goes beyond just differentiating between two classes. Additionally, it seeks to identify the best hyperplane for effectively dividing the two classes in the training dataset. Three key components—the maximization of margins, the use of dual theory, and the kernel trick—are what make the SVM algorithm effective. The basic objective of conventional support vector classification is to maximize the distance between two hyperplanes. This goal can be achieved by solving a convex quadratic programming problem (QPP). The dual theory makes it possible to apply kernel tricks and incorporate kernel functions to address nonlinear situations. The training data for the binary classification can are given by $\{x_i, y_i\}$, $i = 1,2, \ldots, r$ and $y_i = \{-1, 1\}$, where r is the number of training samples and the training vector is made up of the two classes.

α_1 and α_2, which are represented by $y_i = 1$ and $y_i = -1$, respectively. According to the equation, if the classes are linearly separable, it is possible to define at least one hyperplane that can properly separate the classes (training error 0).

$$w.x + b = 0 \tag{3}$$

where w is the vector normal to the hyperplane and b is an offset. To find such a hyperplane, w and b are estimated in a way that $y_i(w.\, x_i + \mathrm{b}) \geq 1$ for $y_i = 1$ that is class

α_1 and $y_i(w.\, x_i + b) \leq -1$ for $y_i = -1$ for class α_2. These can be combined into one set of inequality as given below:

$$yi(w.xi + b) - 1 \geq 0, \forall xi \tag{4}$$

There are several hyperplane systems that can be used to separate two classes. However, in an n-dimensional space, it is only possible to identify a single hyperplane as optimal. The support vectors refer to the training data points that are in close proximity to the optimal hyperplane. These support vectors can be identified at the boundaries defined by the equations $w.\ x_{\mathrm{i}} + b = \pm 1$. The determination of the optimal hyperplane separation is contingent upon the center of the margin. Determining the optimal hyperplane that effectively separates two classes involves maximizing the distance between the nearest samples belonging to the respective class. From the mathematical standpoint, the primary aim is to differentiate between the two classes by ascertaining the utmost distance between their support vectors. The mathematical formulation of the issue in conventional SVM is represented as a convex quadratic programming problem (QPP) [15].

$$\min_{w,b} \frac{1}{2}\|w\|^2 \tag{5}$$

such that:

$$((w.x_i) + b) \geq 1 \tag{6}$$

The constraints can be changed for nonlinearly separable classes by introducing slake variables ζ_i. The QPP is reformulated as follows [15]:

$$\min_{w,b,\xi} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{r} \xi_i \tag{7}$$

Such that

$$y_i((w.x_i) + b) \geq 1 - \xi_i, i = 1, \ldots, r; \xi_i \geq 0, i = 1, \ldots, r \tag{8}$$

where $\zeta = (\zeta_1, \zeta_2, \ldots, \zeta_r)^T$ and $C \geq 0$ is a penalty parameter. This is the primal problem, which is solved by using Lagrangian dual problem given by [15]:

$$\min_{\alpha} \frac{1}{2}\sum_{i=1}^{r}\sum_{j=1}^{r} \alpha_i\alpha_j y_i y_j K\left(x_i, x_j\right) - \sum_{i=1}^{r} \alpha_i \tag{9}$$

Such that

$$\sum_{i=1}^{r} y_i\alpha_i = 0, 0 \leq \alpha_i \leq C, i = 1, \ldots, r \tag{10}$$

where $K(x, x')$, the kernel function is a convex QPP. The structural risk minimization (SRM) principle is utilized to minimize the confidential interval $\|w\|^2$ and empirical risk term $\sum^r \zeta_i$ simultaneously.

3.4.2 Random Forest (RF)

Bagging techniques in ensemble learning were used to construct classification and regression trees, also known as CART. There is a potential for the trees to demonstrate correlation and statistical dependence as a result of their construction using identical data. Breiman [16] introduced a technique to improve the diversity and decrease the correlation in bagged trees. This technique involves incorporating a random selection of attributes and observation samples during each split of the tree. Therefore, this approach is frequently known as random forest (RF) methodology. In the random forest (RF) algorithm, it is necessary to define three parameters: the total number of trees (B) that determines the magnitude of the forest, the number of predictors (m) that signifies the randomly chosen subset of p variables (features) for each split, and the least number of observations that each node can have ($n_{\min}$), commonly referred to as the leaf size. The RF building algorithm consists of three primary steps, as outlined in reference [17].

1. Use the training data to Create B overlapping or replaceable sample datasets, each of size N.
2. Build a random forest tree T_b for each sample dataset by repeatedly applying the next steps to each terminal node. The process will continue till the required minimum node size $n_{\min}$ is obtained.
   (a) Randomly select m predictors from the set of p variables.
   (b) Select one of the m selected predictors as the best predictor for the split point.
   (c) Establish a set of decision rules to partition this particular point (node) into two distinct child nodes.
3. Finally, the objective is to determine the collection of trees $\{T_b\}^B$, wherein B is the number of trees in the random forest.

By averaging the results of the different trees, we may derive a prediction for a specific value x of the response variable:

$$\hat{f}_{RF} = \frac{1}{B}\sum_{b=1}^{B} T_b(x) \tag{11}$$

The ensemble learning algorithm iteratively constructs an assembly of regression or classification trees using the input data to capture the association between the features and the output effectively. The process of decorrelating the trees within

a random forest model serves to decrease the level of variability and enhance the overall reliability of the results.

4 Results and Discussion

The current investigation involves developing a tone recognition system designed for Galo tonal words. A dataset of 14,623 tonal words spoken by 27 Galo speakers (13 males and 14 females) has been considered for developing the system. The dataset consists of twenty Galo tonal words with their tonal instances. Seventy percent of the dataset has been considered for training the system and the remaining 30% for testing the system. The speech signal is first pre-emphasized and framed into the segment of 30 microseconds with one-third overlapping using a Hamming window given by:

$$w(n) = 0.54 - 0.46\cos(2\pi nN) \tag{12}$$

The F0 values are computed by applying the autocorrelation method to each speech frame. The F0 values obtained from the computation process are subjected to normalization using z-score normalization techniques. Additionally, a smoothing operation is performed on the F0-contour. Figure 2 depicts the smoothed pitch contour for the Galo tonal words */aii/* and */'aii/*. The seven features extracted from the normalized and smoothed fundamental frequency (F0) contour are mean F0, median F0, standard derivative of F0, F0 value measured at 0, 25, 75, and 100% of the pitch contour. Consequently, a 7-dimensional feature vector has been obtained for each frame. The mean, variance, and standard deviation statistical features were derived from the segment-level features to create the utterance-level feature vector. Consequently, a 21-dimensional feature vector was obtained for each utterance. An SVM model is trained using the feature vector. The RBF kernel is used for the SVM model. In the next experiment, we apply the random forest algorithm to train a model to recognize tones associated with the tonal words. The number of trees and their depth are tuned dynamically. To cross-validate the model, the KFold grid search algorithm is used. The models are tested for their recognition accuracy using the test dataset. The confusion matrices for the SVM classifier and RF classifier are presented in Fig. 3a, b, respectively. The recognition accuracy of the conducted experiments are briefly outlined in Table. 2.

The experiments show that the F0-based features have tone discriminative capability in the context of the Galo tonal word. The SVM and RF models show promising performance in recognizing the lexical tones associated with the tonal words. The SVM model with RBF kernel gives slightly higher recognition accuracy compared to the RF model.

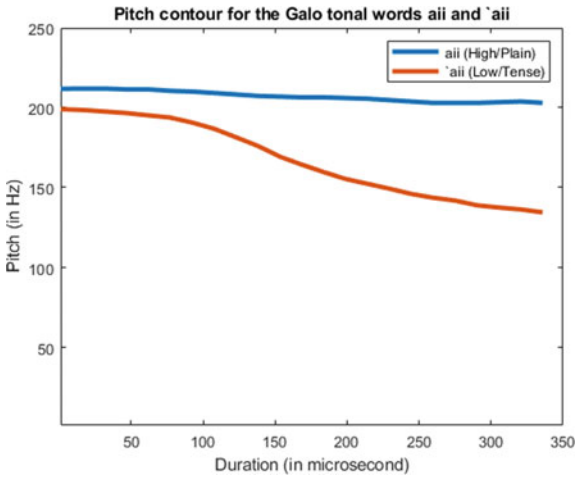


Fig. 2 Smoothed pitch contour for the Galo tonal words aii and ‘aii

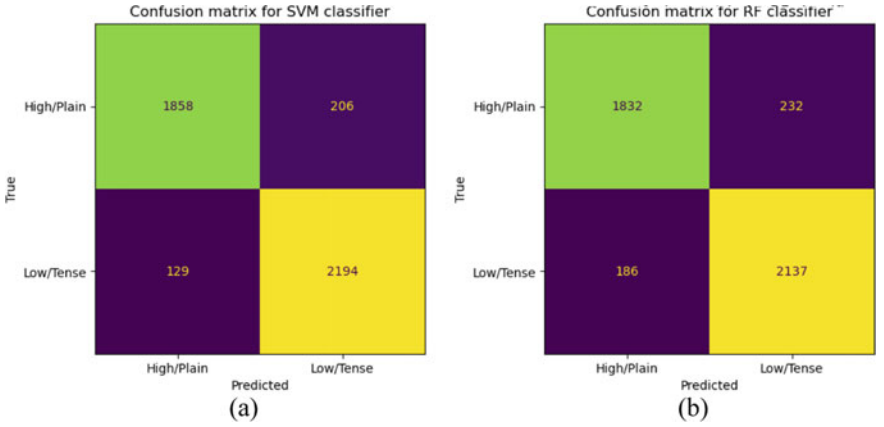


Fig. 3 Confusion matrix for **a** SVM classifier and **b** Random forest classifier

Table 2 Evaluation metrics for the RF and SVM classifiers

Evaluation metrics	RF (in %)	SVM (in %)
Precision	90.47	92.36
Recall	90.49	92.46
F1 score	90.37	92.23
Accuracy	90.42	92.31

5 Conclusion and Future Works

The tone is important in distinguishing between tonal words. We created a simplified tone recognition system in this paper using only the feature vectors derived from the F0-contour. SVM and RF are used to model the tone. One of the most important findings of this work is that the feature vector derived from the F0-contour has tone-discriminating capability in the context of the Galo language. We tested two machine learning models for tone recognition, SVM and RF. The SVM and RF models have recognition accuracy of 92.31% and 90.42%, respectively. The aforementioned findings highlight the necessity of developing an algorithm that effectively integrates independently generated scores for base syllables and tones. Using these scores, a comprehensive decision can be made regarding tonal syllables. The significance of these findings lies in their relevance to the recognition of tonal low-resource languages. In this context, the tones are identified based on standard spectral features, effectively isolating the tone recognition process from other activities. These findings open up possibilities for utilizing cutting-edge pre-trained models developed for non-tonal systems in tonal environments through the fusion of independent scores.

References

1. Moore R (2007) Presence: a human-inspired architecture for speech-based human-machine interaction. IEEE Trans Comput 56(9):1176–1188
2. Li YW, Cheng X, Ding C, Galvin JJ, Chen B, Fu QJ (2023) Benefits of long-term music training for segregation of competing speech by tonal language speakers. J Acoust Soc Am 153(no. 3_ supplement):A330–A330
3. Post MW (2007) A grammar of Galo. Ph.D. Dissertation. La Trobe University
4. Chen XX, Cai CN, Guo P, Sun Y (1987) A hidden Markov model applied to Chinese four-tone recognition. In: ICASSP'87 IEEE international conference on acoustics, speech, and signal processing, vol 12, pp 797–800 IEEE
5. Chang, PC, Sun SW, Chen SH (1990) Mandarin tone recognition by multi-layer perceptron. In: international conference on acoustics, speech, and signal processing, IEEE pp 517–520
6. Lee T, Ching PC, Chan LW, Cheng YH, Mak B (1995) Tone recognition of isolated Cantonese syllables. IEEE Trans Speech Audio process 3(3):204–209
7. Lee T, Kochanski G, Shih C, Li Y (2002) Modeling tones in continuous Cantonese speech. In: Proceedings ICSLP, pp 2401–2404
8. Peng G, Wang WS (2004) Parallel tone score association method for tone language speech recognition. In: Proceedings international conference on spoken language processing (ICSLP)
9. Wang S, Levow GA (2008) Mandarin Chinese tone nucleus detection with landmarks. In: ninth annual conference of the International Speech Communication Association
10. Hsieh YL, Chuang CT, Hsieh FF, Chang YC, Hsu WL (2014) Taiwanese tone recognition using fractionalized curve-fitting of prosodic features. In: Proceedings 7th international conference on speech prosody 2014, pp 772–775
11. Gogoi P, Tzudir M, Sarmah P, Prasanna SRM (2020) Automatic tone recognition of Ao language. In Proceedings 10th international conference on speech prosody 2020, pp 1005–1008
12. Gogoi P, Dey A, Lalhminghlui W, Sarmah P, Prasanna SM (2020) Lexical tone recognition in Mizo using acoustic-prosodic features. In: Proceedings of the twelfth language resources and evaluation conference, pp 6458–6461

13. Zhao X, Shaughnessy DO, Minh-Quang N (2007) A processing method for pitch smoothing based on autocorrelation and cepstral F0 detection approaches. In: 2007 international symposium on signals, systems and electronics, pp 59–62 IEEE
14. Vapnik V (1999) The nature of statistical learning theory. Springer science and business media
15. Gigović L, Pourghasemi HR, Drobnjak S, Bai S (2019) Testing a new ensemble model based on SVM and random forest in forest fire susceptibility assessment and its mapping in Serbia's Tara National Park. Forests 10(5):408
16. Breiman L (2001) Random forests. Mach Learn 45:5–32
17. Abuella M, Chowdhury B (2017) Hourly probabilistic forecasting of solar power. In: 2017 North American power symposium (NAPS), IEEE pp 1–5

Tea Leaf Disease Classification Using an Encoder-Decoder Convolutional Neural Network with Skip Connections



Swati Shinde and Sagar Lahade

Abstract In this study, a Convolutional Neural Network (CNN) model for tea leaf disease classification is introduced, which utilizes an encoder–decoder structure with skip connections. The proposed model is designed to reduce computational complexity and generate high-resolution images from lower-resolution representations obtained using the encoder. The skip connections in the model enable the transfer of information between the encoder and decoder, thereby allowing for the reconstruction of the original dimension image from the encoded representation. The performance of the proposed CNN model is compared against several popular computer vision models, including GoogleNet, ResNet50, and VGG16. An ensemble model is also developed to further improve classification performance. In addition, two hybrid CNN models are created, one by combining the computer vision models without using them as feature extractors, and the other by using their feature extraction capabilities with various machine learning algorithms for image classification. Experimental results indicate that the proposed CNN model with skip connections outperforms the individual models with accuracy, precision, recall, and F1-score 87.36, 88.12, 87.36, 87.30, respectively. Based on our experimental results, our proposed CNN model with skip connections did not outperform the ensemble and hybrid models in terms of accuracy. However, the difference in accuracy was marginal, and our proposed model offered lower computational complexity, making it a more efficient option. Therefore, while our proposed model may not be the best performer in terms of accuracy, it can still provide a viable alternative for tea leaf disease classification, especially when computational efficiency is a critical factor.

Keywords CNN · Vgg16 · Computer-vision

S. Shinde · S. Lahade (✉)
Pimpri Chinchwad College of Engineering, Pune, India
e-mail: sagar.lahade21@pccoepune.org

S. Shinde
e-mail: swati.shinde@pccoepune.org

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2023

S. Shakya et al. (eds.), *Fourth International Conference on Image Processing and Capsule Networks*, Lecture Notes in Networks and Systems 798,
https://doi.org/10.1007/978-981-99-7093-3_24

1 Introduction

Tea, a pivotal cash crop, plays a significant role in augmenting rural development, ameliorating poverty, and ensuring food security in both developed and developing countries. It is the predominant source of livelihood for numerous smallholder farmers and bears immense cultural significance in several nations. Moreover, tea consumption is associated with a plethora of health benefits, including anti-inflammatory and antioxidant properties, and weight loss effects. With its origins dating back to 2700 BC, tea made from the leaves of the Camellia sinensis plant is one of the oldest and most widely consumed beverages worldwide. The information provided is sourced from the Food and Agriculture Organization of the United Nations.

Despite the annual global tea production exceeding $17 billion, the value of the global tea trade is estimated to be approximately $9.5 billion [1]. In 2022, China emerged as the top revenue-generating country in the tea industry, amassing a staggering sum of 99,810 million USD. Following closely behind, India recorded an income of 15,670 million USD, ranking second in the world. India, which accounts for 30% of the world's tea production, is the largest consumer of tea globally. Moreover, India is one of the top five tea exporters globally, contributing 10% of the total global tea exports. The consumption of tea in India is anticipated to increase by 2.1% per annum. The tea industry in India provides direct employment opportunities to over one million individuals (10 lakhs) and indirectly supports the livelihoods of another 10 million people (100 lakhs) through its forward and backward linkages.

The production of tea is a significant contributor to India's agricultural output and GDP. India had been the foremost producer of tea for nearly a century, but China has recently overtaken India due to the availability of more land [2]. The tea industry faces a daunting challenge from leaf diseases that significantly impact yields, as tea's high export value makes it a crucial economic driver for tea-growing countries. The pathogen's direct impact on the plant's delicate leaves is a threat to the industry. Leaf infections pose a major concern as tea leaves used for tea production originate from young plant shoots. Given the economic significance of tea, any disruptions to yield are of paramount importance. Tea crops have suffered losses ranging from 6 to 14% of their production due to the ravages of insect pests. Effective prevention and control of tea leaf diseases necessitate timely diagnosis and identification.

Artificial Intelligence (AI) plays a pivotal role in the detection of tea leaf diseases by virtue of its ability to accurately analyze vast volumes of data from disparate sources and discern patterns that may elude human observation. Employing machine learning and other AI techniques, computers can learn to identify the nuanced symptoms of various tea leaf diseases, empowering farmers to take appropriate measures to forestall or mitigate their propagation [3]. By automating the detection and diagnosis process, AI can save time and resources while enhancing the precision of disease identification. AI facilitates quick identification and treatment of diseased plants, curtailing the spread of diseases to neighboring plants and minimizing crop losses. Additionally, AI-based disease detection systems can provide real-time updates and

alerts, empowering farmers to take timely action to safeguard their crops. In sum, the use of AI in tea leaf disease detection is imperative for ensuring crop vitality, minimizing crop losses, and boosting yields. Given the escalating demand for tea worldwide, AI-based disease detection systems represent a sustainable and efficient solution for tea farmers globally.

This research aims to develop a robust predictive model for tea plant disease classification based on leaf images. The proposed methodology involves the implementation of an encoder-decoder-based CNN architecture. The encoder network captures and encodes essential features from tea leaf images, while the decoder network reconstructs the input image using the extracted features. Through careful preprocessing of a diverse dataset and iterative refinement techniques, the study seeks to achieve high accuracy in disease classification. The outcomes of this research will contribute to the field by providing an effective approach for detecting tea plant diseases, thereby improving crop health management and enabling early disease diagnosis in tea plantations.

2 Literature Review

The literature on the various approaches and methodologies used for detecting diseases in tea leaves is summarized in Table 1 that comes next.

3 Dataset

The tea sickness dataset [17] version 2, published on May 20, 2022, consists of tea leaves images displaying seven common diseases of tea, along with a healthy class. The dataset contains approximately more than 100 images per class (excluding healthy class) and was collected from a Johnstone Boiyon farm, Koiwa location, Bomet county. The dataset was contributed by Gibson Kimutai and Anna Förster and has a DOI of https://doi.org/10.17632/j32xdt2ff5.2. The objective of this dataset is to be used for transfer learning in machine learning models to predict tea plant diseases. The following Fig. 1 displays a visual representation of the dataset utilized in the study. The dataset contains 885 images of jpg.

Table 2 illustrates the distribution of images of tea sickness dataset which contains images from 8 classes including healthy category.

Table 1 Documentation synopsis of tea leaves disease detection methods

References	Description	Advantage	Disadvantage	Result
[4]	The present study exhibits the capability of Internet of Things (IoT) and deep learning techniques to augment illness identification and management in intricate ecological systems	The utilized network structure in this context is Swim Transformer The model's overall ability to generalize was enhanced through the incorporation of real-time images captured by cameras, which offered a diverse and comprehensive range of data	The reliance on mobile devices for connectivity through 4G network is apparent	The tests conducted in difficult natural surroundings demonstrated that swim transformer's identification rate approached 94%, resnet 90, densenet 91
[5]	An image extraction system was developed to evaluate whether a given input image of a tea leaf is afflicted with disease or is in a healthy state	The proposed model integrates Deep Hashing with Integrated Autoencoders, which is an effective and adaptable method, for retrieving photographs of Tea Leaves Except for segmentation, no preprocessing was utilized	Inadequate dataset employed for training	Mean average precision 98.27%
[6]	The diagnosis of grey blight disease in tea leaves was accomplished through the use of a deep convolutional neural network featuring inverted residuals and linear bottleneck layers	A total of 1320 images were collected from India, and bottleneck layers were utilized in the model	The dataset is exceedingly small	The proposed approach achieved an accuracy of 98.99%

(continued)

Table 1 (continued)

References	Description	Advantage	Disadvantage	Result
[7]	To distinguish between disease symptoms and healthy tea leaf regions, K-means clustering was employed for identification, followed by the extraction of features using different color models. The resulting features were classified using SVM	The processing times of the proposed method are faster when compared to those of a traditional CNN	In the suggested strategy, background reduction is essential	GLCM effectiveness with PCA is 98.97% accurate
[8]	The approach involved converting RGB images to HSI format, isolating green pixels, performing segmentation, and utilizing LeNet for classification	The proposed method demonstrated higher precision when compared to the conventional backpropagation network	The identification of red leaf spot appears to be somewhat less accurate, with an 85% precision rate	Accuracy 90.23%
[9]	By integrating different CNN modules such as ResNet and Inception-v3, the MergeModel is capable of extracting a diverse range of distinguishing features, which enhances its ability to identify diseases	This technique allows training to occur with less data than is customary while still improving recognition performance	The model is becoming increasingly complex	On the small-sized dataset, MergeModel outperforms Inception-v3 by 13% in disease identification and ResNet34 by 5%
[10]	A CNN model comprising of 7 layers was utilized	Some images were captured using a field camera	The customary number of filters used in a convolutional layer is 64 or 128	94.45% accuracy
[11]	C-DCGAN enhanced training data	Support Vector Machine (SVM) analysis was employed to detect and discriminate disease spots in images of tea leaves	VGG16 suffers from a significant overfitting problem, and its accuracy drastically reduces in the absence of augmented data	90% accuracy

(continued)

Table 1 (continued)

References	Description	Advantage	Disadvantage	Result
[12]	Prior to being utilized for training the network, raw images underwent pre-processing techniques such as image segmentation and data augmentation	In comparison to SVM and BP, CNN has been found to be more efficient	CNNs have shown to be effective with large datasets	CNN has a 93.754% success rate in recognising images
[13]	The models, which were initially trained using ImageNet data, were retrained to detect diseases in tea leaves	Transfer Learning with full fine-tuning accuracy improved	Evaluation of transfer learning and tuning on small-footprint architectures ResNet, VGGNet, and Xception	Accuracy is improved with respect to training from scratch Accuracy obtained ResNet 94.05 VGGNet 91.26 Xception 91.27
[14]	Images captured in the wild may be inaccurate due to various factors such as changing lighting conditions, shadows, irregular shapes, and overlapping leaves	The Retinex technique was incorporated to enhance the quality of the original images CNN-based methods have facilitated the more rapid and efficient identification of faint, obscured, and diminutive diseases in tea leaves. In addition, VGG16 networks have been utilized to perform severity grading by inputting the diseased leaves	There may be variations in the light conditions in different regions of China, which may not be reflected in the images present in the dataset	Both the average precision of detection and the accuracy of severity rating have improved by more than 6% and 9%, respectively, when compared to conventional machine learning techniques

(continued)

Table 1 (continued)

References	Description	Advantage	Disadvantage	Result
[15]	In addition, features are gathered and exploited using the dense scale-invariant feature transform. Make a model. incorporating the multi-layer perceptron and the support vector machine to categorize diseases on the basis of descriptions provided by a bag of visual words	Local characteristics taken from pictures that remain unchanged by translation, scaling, or rotation	While certain diseases only appear at specific periods of the year, others can affect tea plants at any time. The timing of disease diagnosis could therefore vary throughout the year, which would have an impact on how accurately diseases are identified	LeafNet accuracy 90% SVM 60% MLP 70%
[16]	In this research, they reveal a Deep Learning modified Mask R-CNN for fully independent segmentation as well as tracking of tomato plant leaf disease	Optimizing the RPN network's anchor fractions and indeed the architecture of feature extraction to uplift detection performance and metric computation effectiveness	-	Accuracy 98%

4 Proposed Method

The encoder–decoder architecture, complemented by skip connections and a classification head, demonstrates superior performance in classification tasks, owing to its unique capabilities. This approach offers a synergistic integration of hierarchical feature learning, spatial context preservation, and context-aware predictions.

During the encoding phase, the network effectively captures hierarchical features from the input image, encompassing both low-level patterns such as edges and intricate high-level structures. This hierarchical feature learning empowers the model to discern local and global visual patterns, thus facilitating better discrimination between different classes. Additionally, the encoder–decoder architecture retains the spatial context throughout encoding and decoding, a departure from traditional CNNs that lose spatial information through pooling layers. The upsampling process in the decoder preserves the spatial relationships between pixels, thereby ensuring that the model can recognize object parts and their spatial arrangement, which are crucial factors for accurate classification.

Notably, the incorporation of skip connections plays a pivotal role in enhancing the encoder-decoder's performance. These connections facilitate direct access to

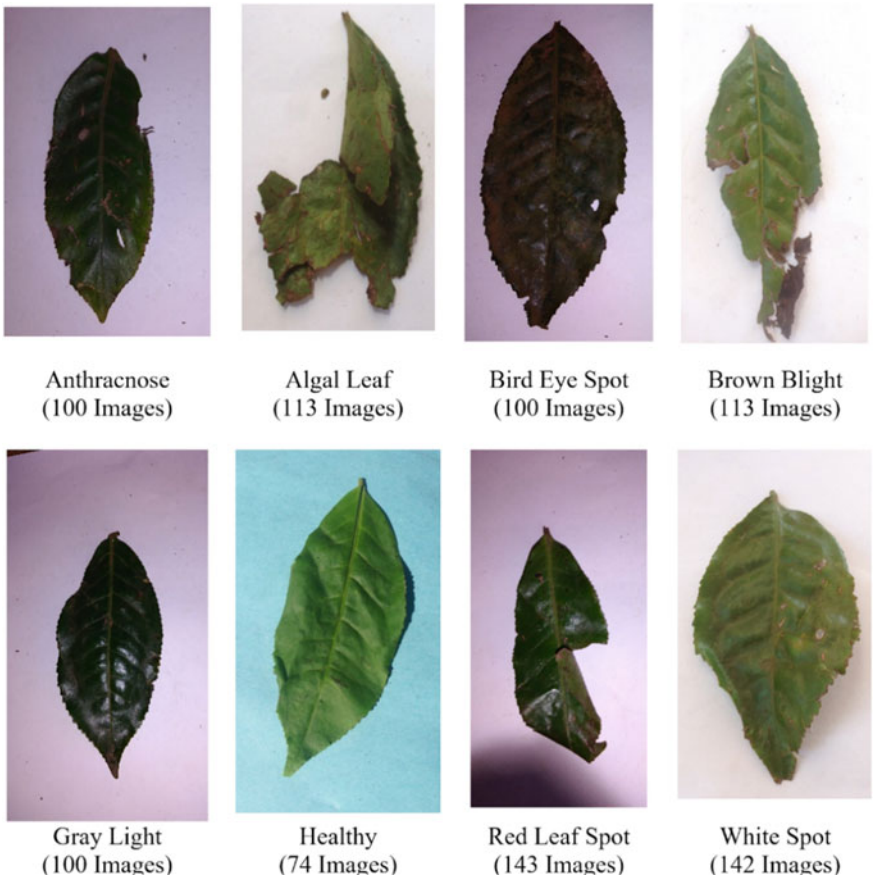


Fig. 1 A graphical depiction of the dataset utilized in the study

Table 2 A depiction of the dataset utilized in the study

Sr. no	Disease name	Total number of images
1	Anthracnose	100
2	Algal leaf	113
3	Bird eye spot	100
4	Brown blight	113
5	Gray light	100
6	Healthy	74
7	Red leaf spot	143
8	White spot	142

Table 3 Dataset splitting to create test, train and validation pipeline

	Algal leaf	Anthracnose	Bird eye spot	Brown blight	Gray light	Healthy	Red leaf spot	White spot	Total
Training	90	80	80	91	80	59	114	114	708
Validation	11	10	10	11	10	7	15	14	88
Testing	12	10	10	11	10	8	14	14	89
Total	113	100	100	113	100	74	143	142	885

multi-level features from the encoder during decoding. As a result, the decoder can effectively utilize high-level semantic information alongside fine-grained details, enabling context-aware predictions. The efficient information flow facilitated by skip connections enables the model to adapt to various levels of granularity in the input data, further augmenting its classification capabilities.

Subsequently, the classification head, which receives the decoder's contextualized features, benefits from the rich representations learned by the encoder–decoder. These context-aware features furnish a more informative and discriminative foundation for classification, outperforming traditional flattened features. Consequently, the classification head delivers more accurate predictions, especially in scenarios involving visually similar classes.

The proposed method section is structured into two distinct parts. Section 4.1 describes dataset distribution used for training and testing Sect. 4.2 outlines the architecture of the proposed method, while Sect. 4.3 details the training process of the model.

4.1 Dataset Distribution for Training and Testing

To create a robust predictive model, the dataset was split into three categories: training, validation, and testing. The table provided illustrates the breakdown of the dataset in terms of the number of images for each class in the three categories. The dataset splitting was designed to ensure that the model could generalize well and perform accurately on new, unseen data. Table 3 presented below outlines the dataset splitting process carried out to develop a reliable predictive model for a group of plant diseases.

4.2 Architecture of Proposed Method

One of the main advantages of this architecture is that it allows for the effective processing of images at multiple resolutions, enabling it to capture both high-level

and low-level image features. The architecture consists of an encoder network that reduces the input image dimensions and extracts its features, followed by a decoder network that generates an output image of the same dimensions as the input image. Skip connections are added between the encoder and decoder networks to facilitate the transfer of information from the encoder to the decoder, which helps preserve low-level image features that would have otherwise been lost during the encoding process.

The architecture appears to be a U-Net style encoder-decoder with skip connections. The input layer is an unspecified 3D tensor with variable shape. The encoder consists of six convolutional layers with increasing number of filters (32, 64, 128, 256, 512, 1024), followed by a max pooling layer. The decoder consists of four convolutional layers with decreasing number of filters (512, 256, 128, 64), with two up sampling layer. The skip connections are implemented through concatenation of feature maps from corresponding encoder and decoder layers. The output layer is a dense layer with 8 units and a softmax activation function, which suggests that the network is trained for a classification task with eight classes.

The total number of parameters in the model is 15,072,136, all of which are trainable. This high number of parameters is likely due to the large number of filters in the convolutional layers, as well as the use of skip connections which add additional connections and computations to the network. Following Table 4 shows architecture of proposed Model.

The proposed architecture is specifically designed to capture and preserve essential features across various levels of image abstraction. By integrating the encoder, decoder, and skip connections, the model effectively utilizes both local and global image context. This comprehensive approach enhances the model's capability to accurately classify tea plant diseases based on leaf images. It ensures that relevant features are adequately represented and leveraged throughout the encoding and decoding stages, ultimately contributing to improved classification performance. Table 4 shows the architecture of proposed CNN.

4.3 Training Process of Proposed Model

The model was trained using the Adam optimizer with a learning rate of 0.0001 and a categorical cross-entropy loss function for a total of 100 epochs.

5 Result

In this study, the three models, Vgg16, ResNet50, and GoogleNet, were trained and evaluated on the tea sickness dataset using the same data splitting approach. To create hybrid models, individual models such as VGG16, ResNet50, and GoogLeNet were trained and tested. The performance of these models on the testing set was

Table 4 Proposed CNN architecture

Sr. No	Layer type	Filter size	Number of filters/neurons	Input shape	Output shape
1	Inputlayer	N/A	N/A	[]	[]
2	Conv2D	(3, 3)	32	(None, None, 3)	(None, None, 32)
3	Conv2D	(3, 3)	64	(None, None, 32)	(None, None, 64)
4	Conv2D	(3, 3)	128	(None, None, 64)	(None, None, 128)
5	Maxpooling2d	(2, 2)	N/A	(None, None, 128)	(None, None, 128)
6	Conv2D	(3, 3)	256	(None, None, 128)	(None, None, 256)
7	Conv2D	(3, 3)	512	(None, None, 256)	(None, None, 512)
8	Maxpooling2d	(2, 2)	N/A	(None, None, 512)	(None, None, 512)
9	Conv2D	(3, 3)	1024	(None, None, 512)	(None, None, 1024)
10	Upsampling2d	(2, 2)	N/A	(None, None, 1024)	(None, None, 1024)
11	Concatenate	N/A	N/A	[(None, None, None, 512)]	(None, None, 1536)
12	Conv2D	(3, 3)	512	(None, None, 1536)	(None, None, 512)
13	Conv2D	(3, 3)	256	(None, None, 512)	(None, None, 256)
14	Upsampling2d	(2, 2)	N/A	(None, None, 256)	(None, None, 256)
15	Concatenate	N/A	N/A	[(None, None, None, 128)]	(None, None, 384)
16	Conv2D	(3, 3)	128	(None, None, 384)	(None, None, 128)
17	Conv2D	(3, 3)	64	(None, None, 128)	(None, None, 64)
18	Global average pooling2d	N/A	N/A	(None, None, 64)	(64)
19	Dense	N/A	128	(64,)	(128)
20	Dense	N/A	8	(128,)	(8)

evaluated, and the resulting confusion matrix is depicted in the figure. Analysis of the confusion matrix reveals that each model achieved 100% accuracy on specific classes. This observation suggests that combining these models into a hybrid model holds promise for improving overall performance and achieving higher accuracy in tea plant disease classification. Following figures shows the confusion matrixes for vgg16, GoogleNet and resnet 50 respectively in Figs. 2, 3 and 4.

The obtained results for VGG16, ResNet50 and GoogleNet model are summarized in Table 5. It is important to note that all models were trained from scratch.

Based on the evaluation results, the GoogleNet model demonstrated the highest performance for the classification of tea sickness among the three models tested, as evidenced by its highest recall and F1 score.

The study combined three models, Vgg16, ResNet50, and GoogleNet, to create an ensemble model. The results obtained from the ensemble model are presented in Table 6.

The ensemble model was created by leveraging pretrained individual models and applying the inverse confusion matrix method to handle the class imbalance problem. The study found that the ensemble model outperformed the individual models, achieving a testing accuracy of 93.10%, which is higher than the best-performing individual model, GoogleNet. The precision, recall, and F1 score for the ensemble model were also impressive, with precision at 93.17%, recall at 93.10%, and F1 score at 92.72%. These results suggest that the ensemble model was able to strike a good balance between accurately identifying true positives, avoiding false positives, and avoiding false negatives, resulting in superior accuracy and balanced performance across all metrics. Thus, it can be concluded that the ensemble model offers an effective approach for accurately identifying tea leaf diseases, outperforming

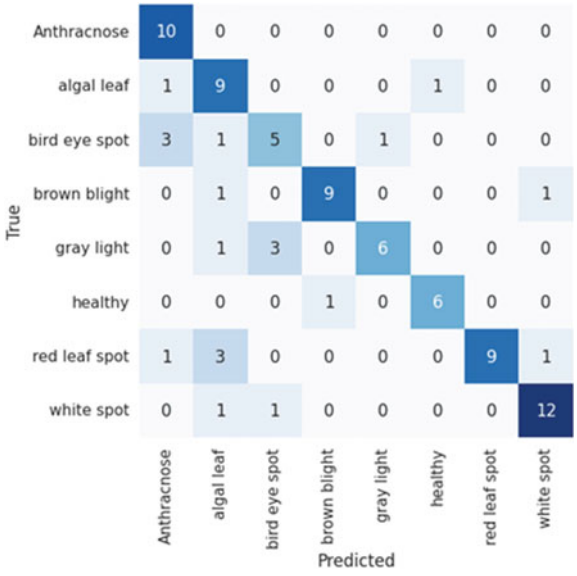


Fig. 2 Confusion matrix of VGG16

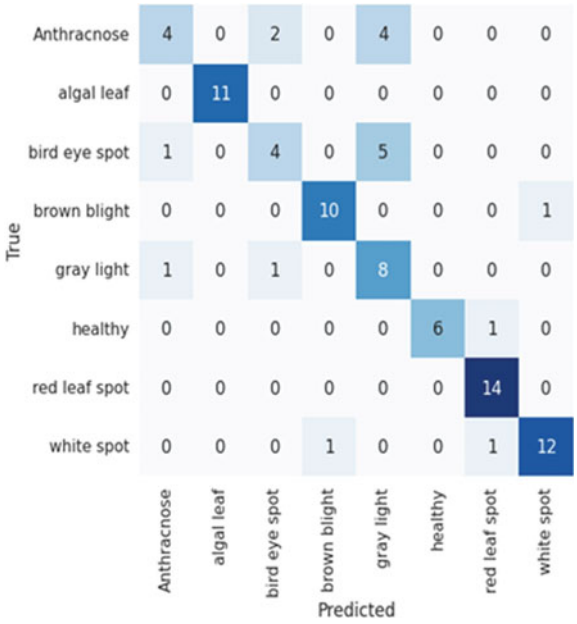


Fig. 3 Confusion matrix of ResNet50

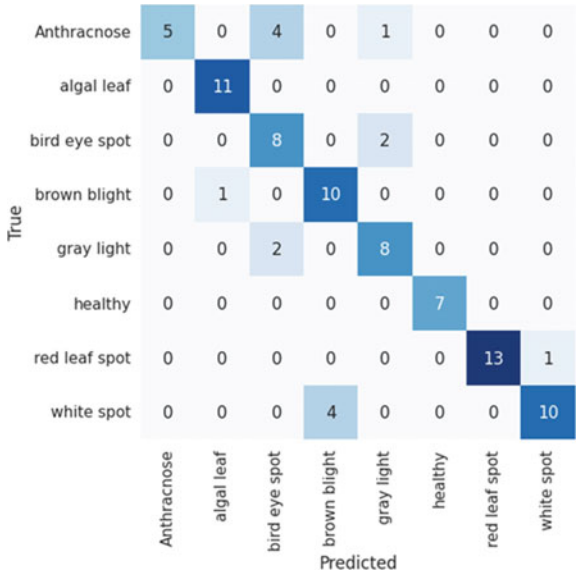


Fig. 4 Confusion matrix of GoogleNet

Table 5 Performance of the individual models trained from scratch on the tealeaf dataset

	Models used	Vgg16	ResNet50	GoogleNet
	Method		Trained from scratch	
Training	Training samples		708	
	Training accuracy	65.59	85.38	84.42
	Validation samples		88	
	Validation accuracy	71.27	79.32	75.52
Testing	Samples		89	
	Accuracy	75.86	79.31	82.76
	Precision	79.17	80.76	85.81
	Recall	75.86	79.31	82.76
	F1 score	75.91	78.86	82.71

Table 6 Performance of the ensemble model trained on the tealeaf dataset

Models used	Vgg16	ResNet50	GoogleNet
Method		Ensemble	
Optimal weight finder		Inverse confusion matrix method	
Unseen testing samples		89	
Testing accuracy		93.10	
Testing precision		93.17	
Testing recall		93.10	
Testing F1 score		92.72	

individual models in terms of accuracy and achieving a good balance between other performance metrics.

Table 7 summarizes the performance of, a model developed by fusing three models (Vgg16, ResNet50, and GoogleNet) using a feature concatenation approach. The model was trained and evaluated on separate training, validation, and testing datasets.

The results obtained for the hybrid model, created using the feature fusion approach to combine three models, Vgg16, ResNet50, and GoogleNet, were summarized in table provided. The testing accuracy achieved by the model was 86.2%, which was higher than the highest performing individual model (GoogleNet). In addition, the model achieved a precision of 86.9%, recall of 86.65%, and an F1 score of 86.6%.

Overall, the results suggest that while the individual models achieved high training accuracy, their performance on the validation and testing datasets was relatively lower. However, the hybrid model, created using feature fusion, was able to achieve a relatively high testing accuracy and a good balance between precision and recall.

The study employed a hybrid model created by using feature fusion approach to combine vgg16, ResNet50, and GoogleNet as a feature extractor to extract features from input images. The extracted features were reduced to 65 using PCA and were

Table 7 Performance of the feature fusion approach trained on the tealeaf dataset

Models used	Vgg16	ResNet50	GoogleNet
Method		Feature concatenation	
Training samples		708	
Training accuracy		78.69	
Validation samples		88	
Validation accuracy		86.26	
Testing samples		86	
Testing accuracy		86.2	
Testing precision		86.9	
Testing recall		86.65	
Testing F1 score		86.6	

then utilized as input to various machine learning algorithms such as Random Forest, SVM, Decision Tree, KNN, Logistic Regression, Naive Bayes, and LDA. The table below presents the results obtained by these algorithms.

Table 8 presents the performance results of several machine learning (ML) algorithms, when using features extracted by the hybrid model.

Figure 5 presented illustrates the performance of feature extractor approach.

The figure presented depicts the performance of Tealeafnet as a feature extractor. The training dataset consisted of feature samples extracted from 620 images, and the ML algorithms achieved a training accuracy ranging from 90.57 to 100%. The testing dataset consisted of feature samples extracted from 265 images, and the ML algorithms achieved a testing accuracy ranging from 79.18 to 90.70%. The testing precision ranged from 79.67 to 90.91%, while the testing recall ranged from 79.49 to 90.76%. The testing F1 score ranged from 79.42 to 90.81%.

Based on the results, the Decision Tree algorithm achieved the lowest testing precision, recall, and F1 score among the algorithms tested. However, it had a training accuracy of 100%, indicating potential overfitting of the training data. Therefore, Decision Tree may be considered the worst-performing algorithm for this specific dataset and feature extraction technique.

On the other hand, the SVM algorithm achieved the highest testing accuracy of 90.91% among the algorithms tested. The testing precision, recall, and F1 score of SVM were also among the highest. Thus, for this specific dataset and feature extraction technique, SVM may be considered the best algorithm.

Following Table 9 shows the performance of proposed model.

The proposed encoder–decoder CNN model achieved an overall testing accuracy of 87.36%, which indicates that the model is performing well on the task of tea sickness classification. Additionally, the model achieved a testing precision of 88.12%, indicating that it is correctly identifying true positives and avoiding false positives. The model also achieved a testing recall of 87.36%, indicating that it is correctly

Table 8 Performance of the ML algorithms on features obtained by using hybrid model

Feature extractor	Created hybrid model						
Feature extracted	1024	1024	1024	1024	1024	1024	1024
PCA (reduced feature count)	65	65	65	65	65	65	65
ML Algorithm	Random forest	SVM	Decision tree	KNN	Logistic regression	Naive Bayes	LDA
Training samples	620	620	620	620	620	620	620
Training accuracy	100	96.96	100	90.89	95.68	90.57	92.49
Testing samples	268	265	265	265	265	265	265
Testing accuracy	88.84	90.7	79.18	88.47	90.33	89.21	89.21
Testing precision	89.11	90.91	79.67	88.55	90.49	89.81	89.57
Testing recall	89	90.76	79.49	88.44	90.37	89.96	89.2
Testing F1 score	89.01	90.81	79.42	88.32	90.4	89.23	89.3

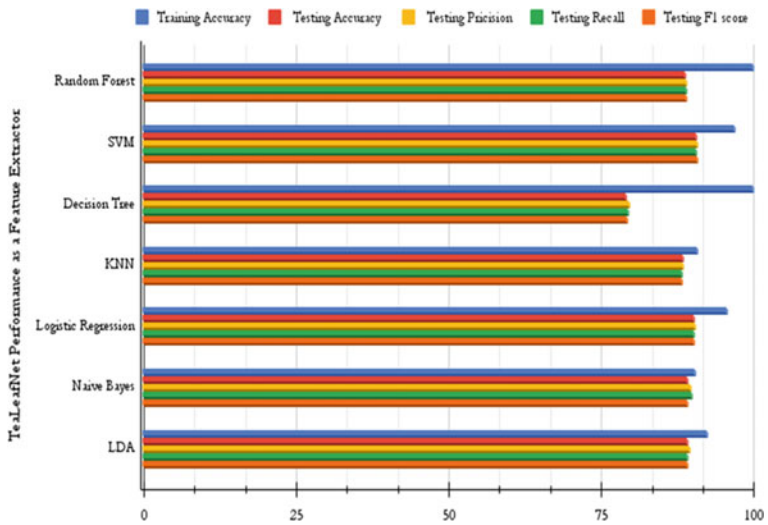


Fig. 5 Hybrid model performance as a feature extractor

Table 9 Performance of proposed model

Model	Proposed model
Training samples	708
Training accuracy	82.86
Validation samples	88
Validation accuracy	83.91
Testing samples	89
Testing accuracy	87.36
Testing precision	88.12
Testing recall	87.36
Testing F1 score	87.30

identifying true positives and avoiding false negatives. The testing F1 score of the model was 87.30%, which is a balanced measure of precision and recall.

Overall, the results suggest that the proposed model has performed well on both the training and testing datasets, with a training accuracy of 82.86% and a validation accuracy of 83.91%. The performance of the model on the testing dataset indicates that it is not overfitting the training data and is generalizing well to new data. The obtained high accuracy can be attributed to encoder-decoder architecture allowed the model to effectively learn and represent complex patterns and distinguishing features present in the tea leaf images. By capturing these intricate details, the model could discern subtle differences between healthy leaves and those affected by different diseases proposed encoder-decoder CNN model, it achieved a relatively high testing accuracy of 87.36% on the Tea Sickness dataset, which is comparable to the best-performing individual model, GoogleNet. Additionally, the proposed model has the advantage of being computationally efficient due to its use of encoder-decoder architecture, which reduces the number of parameters and computational complexity.

Comparing this with the feature extraction approach used in the hybrid model, the proposed model also has the advantage of being an end-to-end learning approach, whereas the hybrid model required the use of multiple machine learning algorithms after feature extraction. Additionally, the proposed model may have a better ability to generalize to new data, as it was trained on the entire dataset, rather than just a subset of features. Feature extraction approach is better than proposed method but it's more complex. The proposed model is better than feature fusion approach. In terms of the ensemble model, while it achieved the best performance among all models, it may not be as computationally efficient as the proposed model, as it required training multiple individual models and then combining them. Furthermore, the ensemble model may not be as interpretable as the proposed model, as it may be difficult to discern the contribution of each individual model to the overall performance.

6 Conclusion

In conclusion, the encoder–decoder architecture with skip connections, combined with a classification head, presents a powerful and effective approach for classification tasks in computer vision. The integration of hierarchical feature learning, spatial context preservation, and context-aware predictions enables the model to extract informative and discriminative features from the input images. The encoder-decoder's ability to retain spatial context and leverage skip connections further enhances its performance, enabling it to handle various levels of granularity in the data. The classification head benefits from the rich representations learned by the encoder-decoder, resulting in more accurate predictions, especially for visually similar classes. Overall, this architecture proves to be a promising and versatile solution for achieving improved performance in classification tasks, making it a valuable addition to the field of computer vision research.

References

1. International tea market: market situation, prospects, and emerging issues, food and agriculture organization of the United Nations (FAO), Published: May 2022
2. Chen J et al (2020) Automatic recognition of tea diseases based on deep learning. Adv Manage Global Change (IntechOpen) 2020 chapter 8. https://doi.org/10.5772/intechopen.91953
3. Bao W, Fan T, Hu G et al (2022) Detection and identification of tea leaf diseases based on AX-RetinaNet. Sci Rep 12:2183. https://doi.org/10.1038/s41598-022-06181-z
4. Xie S et al (2023) Online identification method of tea diseases in complex natural environments. IEEE Open J Comput Soc 4. https://doi.org/10.1109/OJCS.2023.3247505
5. Jayapal SK et al (2023) Enhanced disease identification model for tea plant using deep learning. Intell Autom Soft Comput. https://doi.org/10.32604/iasc.2023.026564
6. Pandian JA et al (2023) Grey blight disease detection on tea leaves using improved deep convolutional neural network. Comput Intell Neurosci 2023. https://doi.org/10.1155/2023/7876302
7. Javidan SM et al (2023) Diagnosis of grape leaf diseases using automatic K-means clustering and machine learning. Smart Agric Technol 3:100081. https://doi.org/10.1016/j.atech.2022.100081
8. Gayathri S et al (2020) Image analysis and detection of tea leaf disease using deep learning. In: 2020 international conference on electronics and sustainable communication systems (ICESC). https://doi.org/10.1109/ICESC48915.2020.9155850
9. Hu et al (2022) Using a multi-convolutional neural network to automatically identify small-sample tea leaf diseases. Sustain Comput Inf Syst 35:100696. https://doi.org/10.1016/j.suscom.2022.100696
10. Latha RS et al (2021) Automatic detection of tea leaf diseases using deep convolution neural network. In: 2021 international conference on computer communication and informatics (ICCCI). https://doi.org/10.1109/ICCCI50826.2021.9402225
11. Hu G et al (2019) A low shot learning method for tea leaf's disease identification. Comput Electron Agri 163:104852. https://doi.org/10.1016/j.compag.2019.104852
12. Xiaoxiao SUN (2018) Image recognition of tea leaf diseases based on convolutional neural network. In: 2018 international conference on security, pattern analysis, and cybernetics (SPAC). https://doi.org/10.1109/SPAC46244.2018.8965555

13. Ramadan A (2020) Transfer learning and fine-tuning for deep learning-based tea diseases detection on small datasets. In: Ade ramadan, transfer learning and fine-tuning for deep learning-based tea diseases detection on small datasets. https://doi.org/10.1109/ICRAMET51080.2020.9298575,10.1109/ICRAMET51080.2020.9298575
14. Hu G et al (2021) Detection and severity analysis of tea leaf blight based on deep learning. Comput Electr Eng 90:107023. https://doi.org/10.1016/j.compeleceng.2021.107023
15. Chen J et al (2019) Visual tea leaf disease recognition using a convolutional neural network model. Symmetry 11(3):343. https://doi.org/10.3390/sym11030343
16. Kaur P et al (2022) An approach for characterization of infected area in tomato leaf disease based on deep learning and object detection technique. Eng Appl Artif Intell 115:105210. https://doi.org/10.1016/j.engappai.2022.105210
17. Kimutai G, Förster A (2022) Tea sickness dataset. Mendeley Data V2. https://doi.org/10.17632/j32xdt2ff5.2

Deep Learning-Based Classification of Lung Cancer Lesions in CT Scans: Comparative Analysis of CNN, VGG-16, and MobileNet Models

P. M. Hareesh and Sivaiah Bellamkonda

Abstract Lung cancer exhibits one of the lowest survival rates among various cancer types over a five-year period, necessitating the utmost importance of early detection. Lung cancer can be detected in its early stage widely through the Computed Tomography (CT) scans. In recent years, deep learning models, particularly Convolutional Neural Networks (CNNs), have been integrated into end-to-end methodologies for the accurate perception of lung nodules. This research project presents a novel system constructed using CNN, VGG-16, and MobileNet models to automatically classify between nonmalignant and malignant tumor lesions within the lungs. The proposed CNN model, trained with hyper parameters including batch size of 8, learning rate of 0.01 with 50 epochs, achieved a peak accuracy of 95.15%. Likewise, VGG-16 achieved a high accuracy of 95.88% with same hyper-parameters except the learning rate is changed to 0.1. Furthermore, MobileNet demonstrated exceptional performance with an accuracy of 98.39% for learning rate of 0.001 with same values for other hyper-parameters batch size and epochs. The classification of lung cancer lesions into benign and malignant categories was effectively accomplished using the LIDC (Lung Image Database Consortium), IDRI (Image Database Resource Initiative) data sets. This study underscores the efficiency of deep learning-based approaches in automated lung cancer detection and emphasizes the significance of early diagnosis in improving patient outcomes.

Keywords Lung cancer · Deep learning · VGG-16 · MobileNet · CNN

P. M. Hareesh · S. Bellamkonda (✉)
Computer Science and Engineering Department, Indian Institute of Information Technology, Kottayam, Kerala, India
e-mail: sivaiah@iiitkottayam.ac.in

P. M. Hareesh
e-mail: haripmohan82.emo2009@iiitkottayam.ac.in

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2023

S. Shakya et al. (eds.), *Fourth International Conference on Image Processing and Capsule Networks*, Lecture Notes in Networks and Systems 798,
https://doi.org/10.1007/978-981-99-7093-3_25

1 Introduction

Cancer is a leading global cause of mortality, presenting a significant public health concern. In 2020, cancer accounted for 9.93 million deaths worldwide [2], and projections indicate that it can be increased to 13 million by 2030 and 16.3 million by 2040. Among the major cancer types, lung cancer, along with prostate, colorectal, breast, stomach, and cervical cancer, exhibits higher prevalence [1].

Early detection of lung cancer is critical, given its high fatality rates [4]. Artificial intelligence has shown promise in medical image analysis and cancer diagnosis, mitigating discrepancies in human interpretation [3]. Late-stage diagnosis of lung cancer yields a meager 10–16% five-year survival rate, whereas early detection significantly improves it to 70% [5]. Lung nodules serve as an early indicator of lung cancer and can range from 3 to 30 mm in size [6].

Computed Tomography (CT) has revolutionized lung nodule imaging since its advent in the 1970s [7]. However, manual approaches to nodule detection from CT scans face limitations such as human visual system constraints, variations in radiologists' expertise, and workload-related challenges [6]. Additionally, nonspherical nodules can introduce further errors, potentially impacting diagnostic accuracy and efficiency.

To overcome these challenges and improve early lung cancer detection, advanced automated techniques are needed. Hence, this study is mainly focused on investigating the efficient deep learning models in detecting lung nodules from CT scans, addressing the limitations of manual approaches and empowering clinicians with efficient and accurate diagnostic tools.

1.1 Automatic Lung Nodule Detection and Classifications

Research on lung nodule identification using Computer-Aided Diagnosis (CAD) has shown promising results. CAD enhances the identification of lung nodules while diminishing false positives. CAD systems can handle tasks such as lung segmentation, nodule formation, feature extraction, and classification.

1.2 Research Gaps

This study aims to investigate deep learning architectures for lung nodule identification in CT images. The specific objectives include exploring CNNs for feature extraction and classification, evaluating CNN, VGG, and MobileNet architectures for lung nodule classification, and analyzing and comparing results based on different learning rates, batch sizes, and epochs. The motivation is to develop an accurate and effective system for identifying and categorizing lung cancer using CNN, VGG,

and MobileNet. CNNs have shown success in medical image analysis, and VGG and MobileNet architectures have performed well in picture categorization tasks. CNNs, VGG, and MobileNet networks have been widely used in the recent research to detect and classify lung cancer from medical images, improving patient outcomes and reducing healthcare costs. This system empowers doctors to make informed decisions regarding treatment options for lung cancer patients.

2 Literature Survey

CAD systems play a vital role in the early detection of lung nodules and the comprehensive evaluation of lung tissue, resulting in improved efficiency for radiologists when interpreting CT scans. The integration of these techniques in healthcare has significantly enhanced overall performance, particularly in lung cancer screening. A study predicts that by 2025, approximately 23% of CAD systems will be dedicated to lung cancer detection, contributing to the expected 11.6% annual growth rate of the global CAD market [8]. This highlights the increasing importance and utilization of CAD systems in lung cancer diagnosis.

2.1 Lung Cancer Classification Using Conventional Techniques

Nithila et al. [9] presented a CAD system for lung nodule characterization using PSO-based back propagation neural network. It involved histogram equalization, Gaussian blur for image enhancement, and a novel active contour model for lung parenchyma separation. Rule-based approaches were used for nodule identification, and the PSO-based BPNN classifier achieved high success rates for different nodule types.

Several other studies aimed on lung nodule detection and classification using different techniques such as shape analysis, image processing, evaluation of CAD systems, artificial neural networks, and specialized filters like the quantized convergence index filter and the difference image approach. These studies reported promising results and highlighted the benefits of utilizing advanced methods in lung nodule analysis.

The LUNA16 Challenge provided a platform for evaluating nodule identification technologies using CT scans with various nodule sizes. Deep learning techniques showed significant performance improvements compared to traditional image processing methods, indicating the dominance of deep learning in this domain.

2.2 Lung Cancer Classification Using Deep Learning Techniques

Dou et al. [10] developed a method using 3D convolutional neural networks (CNN) to minimize false positives (FPs) in CT scans. Their 3D CNN architecture-encoded spatial information effectively, leading to a more discriminative representation. The focus was on reducing FPs, achieving a sensitivity of 92.2% with 8 FPs per scan.

Monkam et al. [11] proposed three CNN architectures with varying convolutional layers to discriminate micro-nodules from non-nodules in CT images. The most successful model had two convolutional layers and a patch size of 32 × 32 pixels, achieving a sensitivity of 83.82%.

Winkels et al. [12] found that using 3D roto-translation group convolutions rather than the traditional translational convolutions reduced the complexity of CNN models. They applied 3D group convolutions (3D G-CNNs) to effectively reduce false positives in pulmonary nodule identification.

Bushara et al. [13] briefed the deep learning models that used CT images for lung cancer detection, using data augmentation for preprocessing. Their CNN approach achieved 95% accuracy.

Nasser and Abu-Naser [14] developed an ANN-based lung cancer detection model that utilized various human data, including symptoms and medical history, achieving 96.67% accuracy.

Qin et al. [15] introduced a deep learning architecture that integrated fine-grained features from PET and CT images for noninvasive lung cancer diagnosis. Their approach, incorporating multidimensional attention mechanism, achieved an AUROC of 0.92.

Asuntha [16] extracted features using wavelet transform, Histogram of Oriented Gradients (HoG), Local Binary Pattern (LBP), SIFT, and Zernike Moment combined with Fuzzy PSO (FPSO) for selecting features and DNN was then employed to classify.

Gopi Kasinathan [17] introduced a modified computer-aided diagnosis system for lung tumor stage classification, utilizing a deep neural model and cloud data collection to categorize pulmonary illness phases.

Although various techniques using machine learning and image processing have been proposed for nodule detection and classification, algorithms that perform well on one type of nodule may not generalize to others. Deep learning systems require large annotated datasets, which are challenging to obtain in medical imaging due to annotation efforts, privacy concerns, and other factors. Transfer learning and combining other features with CNN-extracted features can be employed to improve classification accuracy.

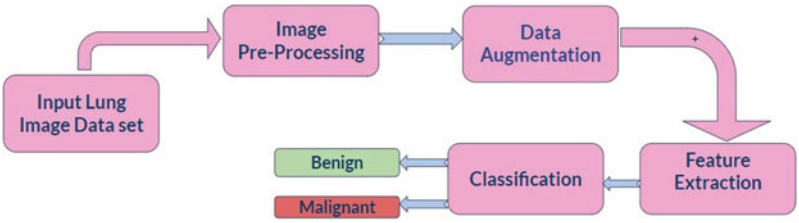


Fig. 1 Architecture of the proposed model

3 Proposed Methodology

3.1 Lung Cancer Classification Using CNN

In a recent study by Thakur et al. [18], it was found that CNNs have the potential to diagnose lung cancer more accurately than manual evaluation by radiologists. Clinical trials have also shown that CNN outperforms professional radiologists in detecting lung nodules. The CNN approach consists of two stages: function extraction and image classification, where the ANN takes over.

CNN utilizes end-to-end learning as its primary learning strategy, but transfer learning can also be employed by leveraging pretrained models. To overcome overfitting issues and improve accuracy, a large dataset is required. Figure 1 illustrates the classification model used to determine whether lung images contain benign or malignant lung nodules.

3.2 Lung Cancer Classification Using VGG-16

VGG-Architecture The VGG-16 network, a deep convolutional neural network (CNN), is the main focus of this research for visual object recognition. It consists 16 layers in which 13 layers are for convolutional and the remaining three are fully connected layers. The convolutional layers extracts features while the fully connected layers classify. Each layer applies multiple filters to the input image, capturing different properties at various resolutions. The network forms a hierarchical representation of the input by passing the results from each layer to the subsequent layer. This hierarchical representation proves effective for object recognition and categorization tasks (Fig. 3).

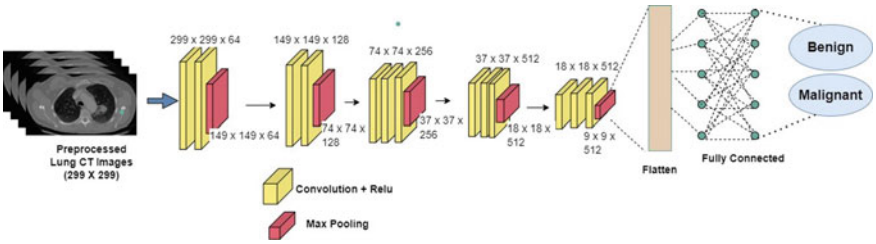


Fig. 2 VGG-16 architecture for lung cancer classification

3.3 Lung Cancer Classification Using MobileNet

MobileNet, a convolutional neural network architecture, is widely recognized for its effectiveness in image classification and object detection on resource-constrained devices. Its combination of accuracy and efficiency makes it suitable for applications such as lung cancer detection and classification. The performance of MobileNet can be evaluated using various metrics to assess its capabilities in this domain. In a feasibility study on lung cancer screening, MobileNet achieved an accuracy of 93.6% in detecting lung nodules from CT scans, indicating its precision and accurate results. Additionally, MobileNet can be utilized to categorize various types of lung cancer such as NonSmall Cell Lung Cancer (NSCLC) and Small Cell Lung Cancer (SCLC), which require distinct treatment approaches. Similar metrics such as precision, recall, and $F1$ score can be employed to evaluate MobileNet's performance in lung cancer classification.

4 Results and Discussion

4.1 Experimental Setup

The experimental setup involved the utilization of Python 3 and the Keras framework for implementing the models. The execution of the models took place on Google Colab, which provided the hardware specifications of an Intel Xeon CPU @2.20 GHz, 13 GB RAM, Tesla K80 accelerator, and 12 GB GDDR5 VRAM. Prior to usage, the Image-Data Generator class in Keras was employed to resize and normalize all input images.

Lung Cancer Database To create and train a convolutional neural network capable of classifying lung images as benign or malignant, the datasets were collected and prepared. The images obtained from the LIDC and IDRI datasets were downloaded and organized into directories: train, test, and valid, each containing separate directories for benign and malignant images. Additionally, a valid directory was created.

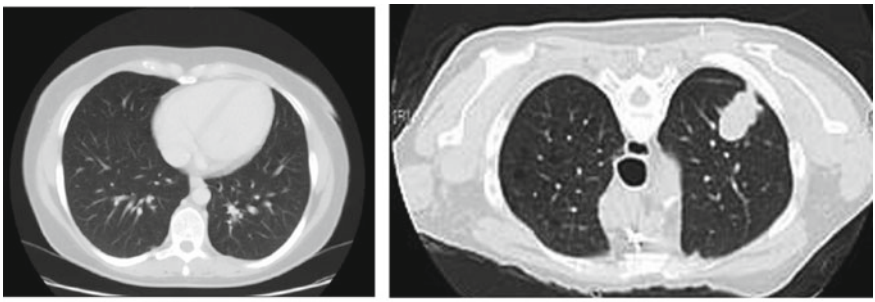


Fig. 3 Example of **a** non-malignant lung image and **b** malignant lung image

For this particular experiment, a total of 2066 images were used. Among them, 80% (1132 images) were allocated for training, while 20% (413 images) were reserved for testing. Out of the training data, 330 images were specifically utilized for validation purposes.

In this research, the radiologists went through two phases of evaluation. In the blinded-reading phase, each radiologist independently examined CT scans and categorized the lesions into three groups: "nodule less than or equal to 3 mm," "nodule less than 3 mm," or "non-nodule greater than or equal to 3 mm." In the unblinded-reading phase, the radiologists reviewed their own marks and anonymized scores provided by the other three radiologists before making their final decisions.

Nodules with a diameter greater than or equal to 3 mm were classified as malignant, while those with a diameter less than 3 mm were considered benign. The research utilized CT scans from the LIDC-IDRI database, and example scans of the lungs are shown in Fig. 3. The DICOM photographs produced by the CT scans had a resolution of 512 by 512 pixels, and the number of slices varied from 64 to 764. On average, the width of the CT scans included in the collection was 240 mm.

Data Preprocessing To overcome the challenge of acquiring a large amount of data for optimal performance of deep learning models, a data augmentation strategy was employed. This strategy involved various techniques such as scaling, flipping, and rotation to fabricate a synthetic dataset. By augmenting the existing data, the dataset size was increased without the need for additional manual data collection, which can be laborious and time-consuming.

The proposed CNN architecture for lung cancer classification, and the architectural design is presented in Fig. 4.

Hyperparameters Table 1 displays the hyper-parameters that were utilized in the training of the CNN for the identification of lung cancer.

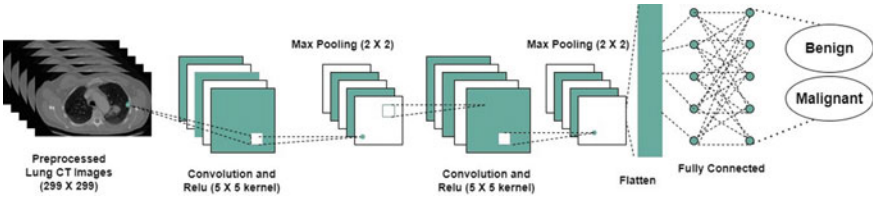


Fig. 4 CNN process flow model

Table 1 Hyperparameters of CNN network

Sl. No.	Hyper-parameters	Value
1	Epoch	50
2	Optimizer	SDG
3	Learning rate	0.0001
4	Batch size	16

4.2 Lung Cancer Classification Using CNN

CNN is trained on the dataset and test on various performance measures. The accuracy and loss curves for both the training and validation sets were plotted, as depicted in Fig. 5.

The accuracy curve shows an improvement in accuracy with increase in the number of epochs. Similarly, the loss curve demonstrates a decrease in loss with an increasing number of epochs, indicating improved model performance.

Figure 6 depicts the evaluation results of the proposed CNN architectures. The ROC curve, shown in Fig. 6a, shows an AUC of 0.912. A higher AUC value suggests better performance of the model, with the range of AUC values being 0.5 to 1.

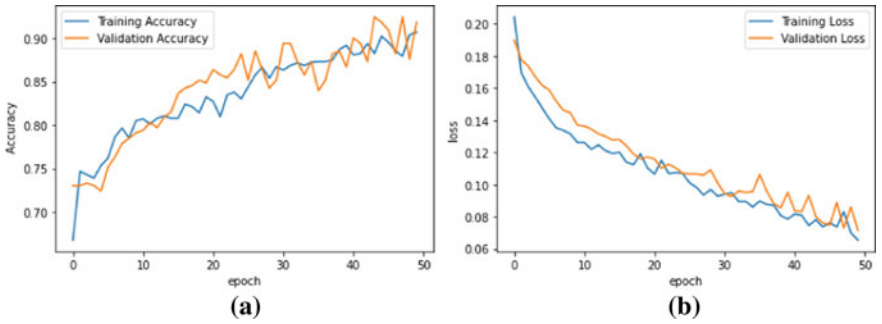


Fig. 5 CNN training **a** acc. versus validation acc. curves; **b** loss versus validation loss curves

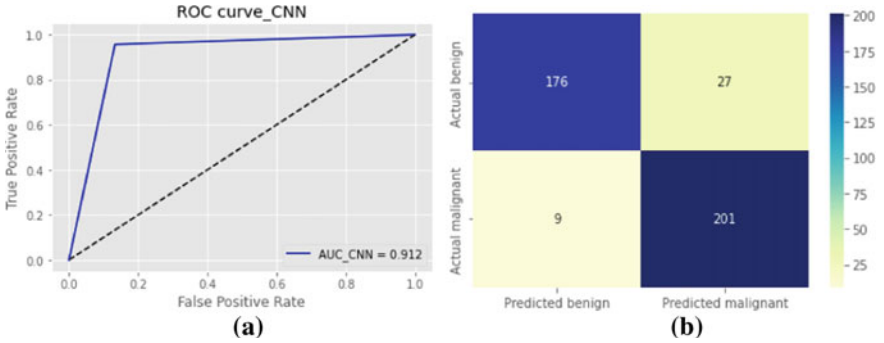


Fig. 6 CNN **a** ROC curve; **b** confusion matrix

The confusion matrix, depicted in Fig. 6b reveals that out of 210 malignant data points, 201 are predicted as true positives (TP) and 9 as false negatives (FN). Additionally, 176 out of 203 benign data points are classified as true negatives (TN), while 27 are false negatives (FN).

The proposed CNN architecture is also evaluated using various metrics such as accuracy, precision, sensitivity, recall, and $F1$-score. The evaluation results are presented in Table 2. The detection accuracies of a proposed CNN model developed for the detection of lung nodules using CT images from the LIDC-IDRI dataset are as follows: Accuracy: 91.28%, Precision: 88.15%, Specificity: 86.60%, Recall: 95.70%, and $F1$-score: 91.78%.

The VGG-16 architecture was employed for lung cancer classification, as depicted in Fig. 2. The accuracy of the modified VGG-16 model and the calculation of the loss function are depicted in Fig. 7.

This study presents the training and validation cycles for two performance metrics using curves. The model weights are recorded when the highest validation accuracy is achieved during training. The maximum accuracy attained during training was 91.45% with a corresponding loss function of 0.0594. The validation accuracy reached 93.94% with a loss function of 0.0507. The latest VGG-16 weights were stored for testing the network. A total of 413 test images, comprising 213 benign and 210 malignant cases, are used.

Table 2 Performance evaluation of proposed CNN architecture

Sl. No.	Quantifying parameters	Value
1	Accuracy	91.28%
2	Precision	88.15%
3	Specificity	86.60%
4	Recall	95.70%
5	$F1$-score	91.78%
6	Area under the curve	0.912

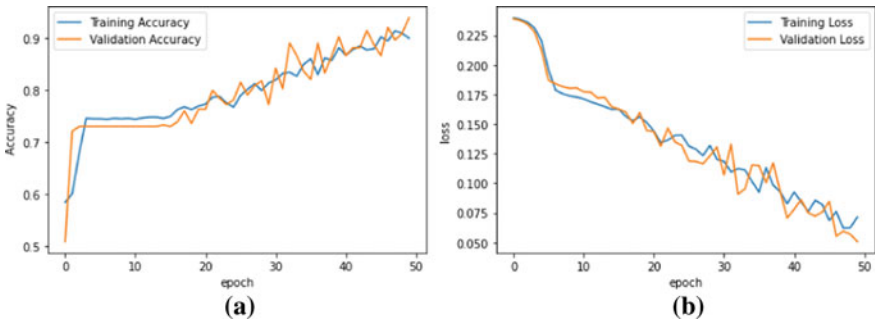


Fig. 7 VGG-16 training **a** acc. versus validation acc. curves; **b** loss versus validation loss curves

The performance of the trained CNN under test conditions is also evaluated using confusion matrix. The confusion matrix of the VGG-16 Architecture is depicted in Fig. 8 as a visual representation. It indicates that out of 210 malignant data points, 194 are predicted as true positives (TP) and 16 as false negatives (FN). Additionally, out of 203 benign data points, 183 are identified correctly as true negatives (TN), while 20 are classified as false positives (FP).

The research includes the ROC curve of VGG-16, shown in a visual representation. The computed area under the curve (AUC) for VGG-16 is 0.913. In this study, the AUC values ranges from 0.5 to 1, where higher values reflect the effective performance of the device.

Furthermore, the performance evaluation of the proposed VGG-16 architecture is produced in Table 3. The evaluation is based on quantifying parameters such as accuracy, precision, specificity, recall, and $F1$-score.

In this research, the VGG-16 model was constructed to detect lung nodules using CT images from the LIDC-IDRI dataset. The model was trained using these images and implemented in Python with the Keras package. The achieved accuracy of the model was 91.28%. The evaluation scores for accuracy, precision, specificity, recall, and $F1$-score were 91.28%, 90.65%, 85.91%, 92.38%, and 91.50%, respectively.

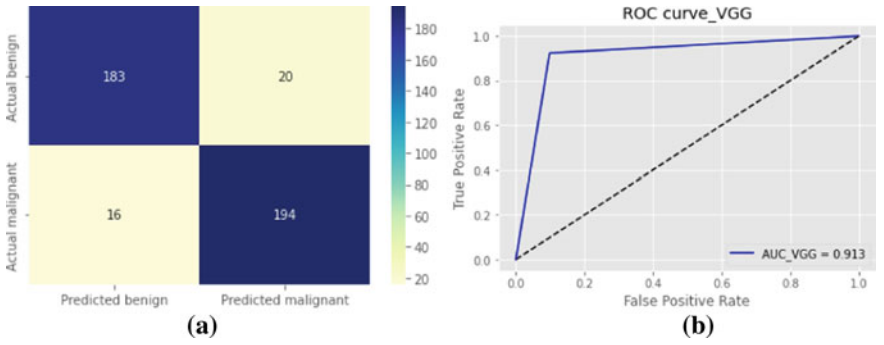


Fig. 8 VGG-16 confusion matrix

Table 3 Performance evaluation of proposed VGG-16 architecture

Sl. No.	Quantifying parameters	Value
1	Accuracy	91.28%
2	Precision	90.65%
3	Specificity	85.91%
4	Recall	92.38%
5	$F1$-score	91.50%
6	Area under the curve	0.913

4.3 Lung Cancer Classification Using MobileNet

Sample images from the training dataset are analyzed following the successful installation of a lung nodule classification model using the MobileNet architecture. The accuracy values and loss curve during the training phase and validation phase are depicted in Fig. 9. The findings reveal an improvement in accuracy with increasing epochs, while the loss decreases as the number of epochs progresses.

The performance of MobileNet is also evaluated using various matrices such as precision, recall, and $F1$ score. By leveraging these performance matrices,

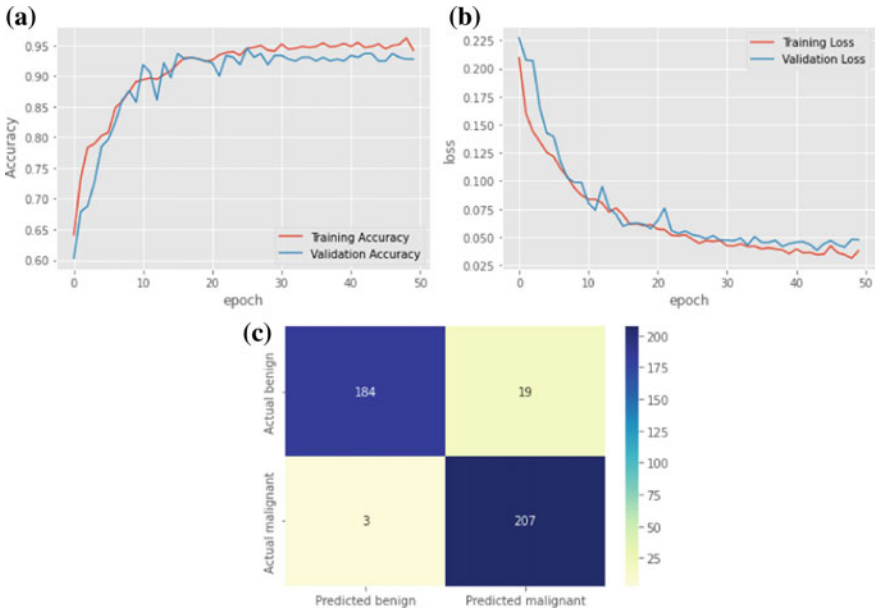


Fig. 9 MobileNet training **a** acc. versus validation acc. curves; **b** loss versus validation loss curves; **c** confusion matrix

Table 4 Performance evaluation of proposed MobileNet architecture

Sl. No.	Quantifying parameters	Value (%)
1	Accuracy	94.67
2	Precision	91.59
3	Specificity	9.64
4	Recall	98.57
5	$F1$-score	94.95

researchers and clinicians can evaluate the accuracy and efficiency of MobileNet in identifying lung cancer, and ultimately improve patient outcomes. The results of the evaluation of various proposed MobileNet architectures is depicted through Table 4.

4.4 Comparative Analysis on Various Hyper-parameters

When training convolutional neural networks (CNNs) like VGG-16 and MobileNet, it is critical to take into account the interaction between learning rate, batch size, and epochs. The adjustment of the weights of a network during each iteration of training is determined by the learning rate. An elevated learning rate has the potential to expedite convergence; however, it can also result in overshooting and instability. The term "batch size" pertains to the quantity of training instances that undergo processing as a group prior to the updating of weights. Increasing the batch size can potentially lead to enhanced weight updates stability, albeit at the cost of higher memory requirements. In the context of neural network training, epochs denote the frequency with which the complete dataset is iterated through the network. Augmenting the number of epochs has the potential to enhance performance, however, it also has the possibility of inducing overfitting. The identification of an optimal combination of said parameters is of utmost importance in attaining a high level of accuracy while circumventing potential complications such as overfitting or unstable training.

Learning Rate The outcomes of the CNN, VGG-16, and MobileNet with different learning rate are displayed in Table 5 for 0.1, 0.01, 0.001, 0.0001, and 0.00001 learning rates. The largest accuracy was obtained for CNN with a learning rate of 0.01, while the least accuracy was obtained with a learning rate of 0.1. The VGG model achieved the highest level of accuracy when learning rate = 0.1, whereas the lowest level of accuracy was observed when learning rate = 0.0001. With a learning rate of 0.01, and 0.001 MobileNet achieved the highest accuracy, while a rate of 0.1 yielded the lowest.

Batch Size The results from executing CNN, VGG-16, and MobileNet with varying batch sizes are presented in Table 6, with the batch sizes 8, 16, 32, and 64 respectively. With a batch size of 8, the proposed CNN architecture achieved better accuracy while the minimum accuracy is obtained where the batch size is 16 and 64. When utilizing

Table 5 Accuracy comparisons of CNN, VGG-16, and MobileNet with different learning rates

Learning rate	CNN (%)	VGG-16 (%)	MobileNet (%)
0.1	50.85	**95.64**	50.85
0.01	**94.43**	95.16	**95.16**
0.001	93.95	95.16	**95.16**
0.0001	91.28	91.28	94.67
0.00001	77.00	95.16	88.31

Bold values signifies best value

a batch size of 8, the VGG model was able to achieve the highest level of accuracy, while when utilizing a batch size of 32, and 64 the model was only able to achieve the lowest level of accuracy. MobileNet attained the maximum level of Accuracy with a batch size of 8, however, a batch size of 16 and 64, resulted in the lowest level of Accuracy.

Epochs Table 7 displays the results of CNN, VGG-16, and MobileNet with 10, 20, 30, and 50 Epochs. CNN achieved the highest Accuracy with an epoch of 50, while the lowest Accuracy was achieved with an Epoch of 10. The VGG model obtained the highest level of accuracy with an epoch of 50, and the lowest level of accuracy with an epoch of 10. With Epochs of 50, MobileNet obtained the highest Accuracy, while Epochs 10 produced the lowest.

The summary of the results are as follows. The CNN fine tuned with 0.01 learning rate, 8 batch size for 50 epochs attained a maximum accuracy of 95.15%. The most accurate prediction was made by VGG-16 with a learning rate at 0.1, batch size at 8, for 50 epochs has recorded an accuracy of 95.88%. The maximum level of accuracy,

Table 6 Accuracy comparisons of CNN, VGG-16, and MobileNet with different batch size

Batch size	CNN (%)	VGG-16 (%)	MobileNet (%)
8	**95.15**	**95.88**	**98.39**
16	94.43	95.16	94.92
32	94.92	93.95	95.40
64	94.43	93.95	94.92

Bold values signifies best value

Table 7 Accuracy comparisons of CNN, VGG-16, and MobileNet with different epochs

Epochs	CNN (%)	VGG-16 (%)	MobileNet (%)
10	81.84	87.17	94.67
20	91.28	94.67	95.40
30	94.67	95.40	95.16
50	**95.15**	**95.88**	**98.39**

Bold values signifies best value

Table 8 Accuracy comparison of proposed method versus state of the art

References	Acc. value (%)
Alakwaa et al. [19]	86.6
Lakshmanaprabu et al. [20]	94.56
Nanglia et al. [21]	98.08
Proposed model	**98.39**

Bold values signifies best value

98.39%, was attained by MobileNet with a learning rate = 0.001, batch size = 8 for epochs.

The highest Accuracy, Precision, Recall, and $F1$-score are 98.39%, 95%, 99%, and 97%, respectively are obtained for MobileNet with learning Rate 0.001, Batch Size 8, and Epoch 50 for lung cancer detection and classification.

Table 8 represents the comparison analysis of the performance of the proposed model with the state of the art on LIDC-IDRI datasets.

5 Conclusion

The aim of the proposed model is to explore various deep learning models for the classification of lung cancer, considering its high mortality rate and significant impact on public health. The proposed model is developed on LIDC-IDRI dataset to classify the lung cancer by employing Convolutional Neural Networks (CNN), VGG-16, and MobileNet models. It was observed from our study that the MobileNet model with a learning rate of 0.01, batch size of 8, and 50 epochs achieved a better accuracy of 98.39%. VGG-16 attained an accuracy of 95.88% with a learning rate of 0.1, 8-sample batches, and 50 epochs whereas CNN attained 95.15 for the same settings. MobileNet yielded the highest accuracy of 98.39% using a learning rate of 0.001, batch size of 8, and 50 epochs. Overall, this study demonstrated the effectiveness of deep learning models, particularly CNN, VGG-16, and MobileNet, in classifying lung cancer using CT images from the LIDC-IDRI dataset.

References

1. Pfeffer M, Ling S (2022) Evolving optimised convolutional neural networks for lung cancer classification. Signals 3:284–295
2. Suresh S, Mohan S (2020) ROI-based feature learning for efficient true positive pre-diction using convolutional neural network for lung cancer diagnosis. Neural Comput Appl 32:15989–16009
3. Samhitha B, Mana S, Jose J, Vignesh R, Deepa D (2020) Prediction of lung cancer using convolutional neural network (CNN). Int J 9

4. Pandit B, Alsadoon A, Prasad P, Al Aloussi S, Rashid T, Alsadoon O, Jerew O (2023) Deep learning neural network for lung cancer classification: enhanced optimization function. Multimed Tools Appl 82:6605–6624
5. Saleh A, Chin C, Penshie V, Al-Absi H (2021) Lung cancer medical images classification using hybrid CNN-SVM. Int J Adv Intell Inform 7:151–162
6. Sajja T, Devarapalli R, Kalluri H (2019) Lung cancer detection based on CT scan images by using deep transfer learning. Traitement Du Signal 36:339–344
7. Tekade R, Rajeswari K (2018) Lung cancer detection and classification using deep learning. In: Fourth international conference on computing communication control and automation (ICCUBEA). IEEE, pp 1–5
8. Wang Z, Gao X (2018) Improved lung nodule diagnosis accuracy using lung CT images with uncertain class. Comput Methods Programs Biomed 162:197–209
9. Nithila E, Kumar S (2017) Automatic detection of solitary pulmonary nodules using swarm intelligence optimized neural networks on CT images. Eng Sci Technol Int J 20(3):1192–1202
10. Dou A, Heng P (2017) Multilevel contextual 3-D CNNs for false positive reduction in pulmonary nodule detection. IEEE Trans Biomed Eng 64(7):1558–1567
11. Monkam P, Qian W (2018) CNN models discriminating between pulmonary micro-nodules and non-nodules from CT images. BioMedical Eng OnLine 17(1)
12. Winkels M, Cohen T (2019) Pulmonary nodule detection in CT scans with equivariant CNNs. Med Image Anal 55:15–26
13. Bushara AR (2022) Deep learning-based lung cancer classification of CT images using augmented convolutional neural networks. Electron Lett Comput Vis Image Anal 21(1):130–142
14. Nasser IM, Abu-Naser SS (2022) Lung cancer detection using artificial neural network. Int J Eng Inf Syst 3(3):17–23
15. Qin R, Wang Z, Jiang L, Qiao K, Hai J, Chen J, Xu J, Shi D, Yan B (2020) Fine-grained lung cancer classification from PET and CT images based on multidimensional attention mechanism. Complexity 2020:1–12
16. Asuntha A, Srinivasan A (2020) Deep learning for lung cancer detection and classification. Multimed Tools Appl 79:7731–7762
17. Kasinathan G, Jayakumar S (2022) Cloud-based lung tumor detection and stage classification using deep learning techniques. BioMed Res Int 79:17 p.
18. Thakur SK, Choudhary J (2020) Lung cancer identification: a review on detection and classification. Cancer Metastasis Rev 39:989–998
19. Alakwaa W, Badr A (2017) Lung cancer detection and classification with 3D convolutional neural network (3D-CNN). Int J Adv Comput Sci Appl 8(8)
20. Lakshmanaprabu SK, Ramirez G (2019) Optimal deep learning model for classification of lung cancer on CT images. Futur Gener Comput Syst 92:374–382
21. Nanglia P, Rathee D (2021) A hybrid algorithm for lung cancer classification using SVM and neural networks. ICT Express 3:335–341

A Novel Fingerprint-Based Age Group Classification Approach Using DWT and DFT Analysis



Ashwini Sanjay Gaikwad and Vijaya B. Musande

Abstract In recent years, fingerprint analysis has expanded to include the estimation of age information, providing an additional layer of personal identification because of its uniqueness and permanence. In this study, 2D-Discrete Wavelet Transform (DWT) and 2D-Discrete Fourier transform (DFT) are used for fingerprint-based age group classification and their performance is compared against fingerprint dataset which is distributed among various age groups from the age of 10 to 90 including both genders. The face-based age group classification techniques have limited practice for crime scene investigation due to their dependency on the availability of vast global features, that is, complete face image. The research gap identifies that the accuracy between 2D-DWT and 2D-DFT, varied, and this is proved even for the unbalanced age group classes collected in this study's real-time database and its corresponding results. The three variants of 2D-DWT Level-4, level-3, level-2 and 2D-DFT are considered for this research work and the most accurate was the DWT Level-4 which classified the age group using the fingerprint images with an astonishing accuracy 99.96 which is verified and supported by detailed analysis of all classification metrics.

Keywords Fingerprint · 2D-discrete wavelet transform · 2D-discrete Fourier transform · Support vector machine classification metrics

1 Introduction

Fingerprint recognition has been a topic of extensive research in recent years, with various techniques proposed for accurate and efficient classification. One such technique is the use of Discrete Wavelet Transform (DWT) for fingerprint-based classification. DWT is a mathematical tool used for signal processing, and its application in

A. Sanjay Gaikwad (✉) · V. B. Musande
Department of Computer Science and Engineering, Jawaharlal Nehru Engineering College, Aurangabad, Maharashtra, India
e-mail: ashwinigaikwad5056@gmail.com

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2023

S. Shakya et al. (eds.), *Fourth International Conference on Image Processing and Capsule Networks*, Lecture Notes in Networks and Systems 798,
https://doi.org/10.1007/978-981-99-7093-3_26

fingerprint recognition has shown promising results. Currently in crime scene investigation, there is inadequate practice of face-based gender classification due to their dependency on massive global features, that is, appearance-based approach that uses whole face image containing thousands of pixels which is reduced to handful number of pixels by dimensionally reduction schemes.

Fingerprint analysis plays a role in convicting the person responsible for a rude crime. Fingerprint biometrics has been widely recognized as a reliable and secure means of identity verification due to its unique and unchanging nature throughout an individual's life. In recent years, fingerprint analysis has expanded to include the estimation of gender and age information, providing an additional layer of personal identification. In the research domain of age group and gender recognition, mainly used biometric is fingerprints due to its uniqueness and do not change throughout the life of an individual. Gender identification is been studied in insufficient research therefore, more examination is required which will increase the accuracy level with a lesser amount of computer complexity. Fingerprint biometrics are widely used for reliable and secure identity verification due to their unique and unchanging nature throughout an individual's life.

In recent years, the scope of fingerprint analysis has expanded to include the estimation of gender and age information, providing an additional layer of personal identification. However, there is a lack of studies on the estimation of gender information from fingerprint images. This research gap presents an opportunity to develop more accurate gender classification methods that require less computer complexity. By addressing this gap in the literature, the proposed research aims to enhance the accuracy and efficiency of fingerprint-based gender identification. This research paper aims to explore the effectiveness of DWT for fingerprint-based classification.

The paper will present a detailed analysis of the DWT-based fingerprint recognition system, including the preprocessing steps, feature extraction, and classification process. The proposed method will be evaluated using a benchmark fingerprint database, and the results will be compared with other state-of-the-art fingerprint recognition techniques. The significance of this research lies in the potential applications of fingerprint recognition in various fields, such as law enforcement, security, and biometric authentication. The proposed DWT-based approach can lead to the development of more accurate and reliable fingerprint recognition systems, which can enhance security and prevent fraud.

2 Related Work

In earlier study, an experiment on human age estimation through fingerprint has been implemented by extracting fingerprint feature using discrete wavelet transform and principal component analysis methods and further combining the feature vectors. Then SVM classifier is applied to estimate the age of the person [1]. An attempt using backpropagation algorithm from artificial neural network is proposed in this research article which matches twelve parameters of fingerprint with a unique number allotted

to every authorized person. Then the best match is given by the proposed algorithm after the matching is done [2]. S. S. Gornale, Mallikarjun Hangarge et al. have proposed in their work, that the gender information can be extracted from fingerprints with the help of Haralick texture features for gender classification. Fingerprints from different age intervals of rural and urban population were collected for performing this experiment. As per the results achieved for linear discriminant analysis (LDA) and quadratic discriminant analysis (QDA) classifiers 92% and 94% classification rate is gained respectively [3].

It is studied from the literature that Daubechies Orthogonal Wavelets (db9) is mainly referred for preparing the feature vector because texture analysis is done smoothly with the help of wavelets, and for computational model Standard deviation and entropy is calculated upon the fingerprint images. DWT decomposes an image into subbands that are localized in frequency and orientation. The literature reveals that many researchers have explored gender classification using various approaches and achieved promising results with their datasets. However, there is still room for the development of a robust algorithm using different parameters [4, 5].

In law enforcement, latent fingerprint identification is an important area of interest. This study analyses the most bulging fingerprint feature representations reported in the literature, recognize drifts in fingerprint feature representation, and observe that representations designed for verification are commonly used in latent fingerprint identification. This experiment aims to assess the performance of the most widespread fingerprint feature representations over a common latent fingerprint database and protocol. Hence, a protocol that evaluates some of the most suitable feature representation is introduced [6].

In a recent research study, time domain approach is referred to identify the gender of a specific fingerprint through methodical pixel counting. This precise pixel counting method got 90.2% accuracy for females' classification and 96.4% for males [7].

Fingerprints are widely recognized as a unique biometric that remains unchanged throughout an individual's life. As a result, fingerprint analysis has been used for identification, verification, and official documentation purposes. Moreover, fingerprints play a crucial role in solving criminal cases, where the analysis of fingerprints often leads to identifying the perpetrator. In addition to identification and verification, fingerprint biometrics can also be used for gender and age estimation. Previous research has used DWT and SVD methods to estimate the age of a person using fingerprints. Furthermore, the unique nature of fingerprints has made them a reliable biometric for gender identification. In conclusion, fingerprint analysis is a crucial aspect of forensic science that plays a vital role in the identification and verification of individuals, as well as in solving criminal cases. Additionally, the unique nature of fingerprints makes them suitable for age and gender estimation. By utilizing different techniques and methods, the accuracy and efficiency of fingerprint analysis can be enhanced, providing more reliable results for various applications [8, 9].

Wavelet transform is a widely used tool in image processing and computer vision due to its comprehensive theoretical framework, flexibility in choosing bases, and low computational complexity. The wavelet transform enables the analysis of an

image in both the frequency and time domains, which makes it particularly useful for feature extraction and pattern recognition tasks [10].

One of the most advanced biometric technologies, fingerprints are accepted as valid forms of proof in courts of law all around the world [11]. Because of this need for trustworthy human identification in computerized access control, interest in biometrics has developed. Based on their visual pattern, all fingerprints fall into one of three categories: arches, loops, or whorls. 60% of people have loops, 35% have whorls, and 5% have arches, making loops a relatively prevalent pattern. Even the fingerprints of identical twins differ for this reason. Both a spatial domain and frequency domain technique can be used to study the gender classification of fingerprints. The spatial domain technique utilizes both the physical properties and pixel values of a picture. The physical features of a fingerprint image that relate to ridges and troughs are called physical parameters [12].

Instead of focusing on the physical parameters of an image, the frequency domain technique works on the measurement parameters of an image. This approach employs the energy of the image as a feature of the image and operates on the frequencies of that image. Discrete Wavelet Transform, Discrete Cosine Transform, and other techniques are used here to extract features [13].

In this study, using a frequency domain technique, which ignores the ridge-based physical properties of a picture in favor of assessing parameters such as frequency and pattern recognition, the features of the fingerprint image are extracted using the 2D Discrete Wavelet Transform. Wavelet serves as the basis function for DWT. A wavelet is an effective limited-duration waveform with a zero average value. An image is divided into sub-bands by two-dimensional DWT that are localized in frequency and orientation (horizontal, vertical, diagonal). Decomposition of images into several frequency bands aids in the isolation of various frequency components. In order to investigate the details of an image, it is helpful to discern minute changes in the image using various frequency components [14].

Four decomposed sub-band images, referred to as low–low (LL), low–high (LH), high–low (HL), and high–high (HH), are produced by the 2-D wavelet decomposition of an image. Each of these sub-band's aids in the analysis of various visual details. Since low frequencies hold the majority of an image's energy, decomposition is often repeated just on the LL sub band. The analysis of these subtleties is made easier by multi-resolution since many signals or images contain features at several sizes. A high-resolution should be used to investigate little items. Examining large items at low resolution is advised. Low–low (LL), Low–high (LH), High–low (HL), and High–High (HH) are the four decomposed sub-band pictures produced by the 2-D wavelet decomposition of an image. Each of these sub-bands is useful for examining various image details. Since the majority of an image's energy is found at low frequencies, decomposition is often repeated solely on the LL sub band. Numerous signals or images have features at different scales, and multi-resolution aids in the examination of such details. High-resolution examination of small items is recommended. Low-resolution examination of large-scale items is recommended.

This work uses the phase information gained by Discrete Fourier Transform (DFT) of images. The phase information conserves the essential image features. Its correlation function, called phase correlation or Phase-Only Correlation (POC), provides us both the good likeness ration for biometric recognition. The researchers have successfully applied phase information to fingerprint, face, iris, and palmprint recognition [15].

3 Proposed Work

3.1 Identification of Research Gap

One potential research gap for age detection using fingerprint is the lack of robust and accurate feature extraction methods that can effectively capture the subtle variations in fingerprint patterns related to age. While some studies have proposed various features for age estimation, there is still a lack of consensus on the most effective and efficient feature extraction techniques. Another research gap is the lack of comprehensive and diverse datasets for age detection using fingerprint, particularly in terms of age ranges and ethnic diversity. Most existing datasets are limited in terms of sample size and diversity, which may limit the generalization of age estimation models to real-world scenarios.

3.2 Motivation

As there is lack of comprehensive and diverse datasets for age detection using fingerprint, particularly in terms of age ranges. It was difficult to detect the fingerprint samples belonging to the different age ranges and working class with better or damaged skin over time, which can potentially impact the quality and clarity of fingerprints. Some fingerprint samples were collected from this part of society as well to get the better accuracy and to explore the differences in fingerprints among age groups.

3.3 Discrete Wavelet Transform

DWT is a wavelet transform for which the wavelets are sampled at discrete intervals. DWT provides a simultaneous spatial and frequency domain information of the image. In DWT operation, an image can be analyzed by the combination of analysis filter bank and decimation operation. The DWT decomposes a digital signal

into different sub bands so that the lower frequency sub-bands have finer frequency resolution and coarser time resolution compared to the higher frequency sub bands.

One Level of the Transform

The DWT of a signal x is calculated by passing it through a series of filters. First the samples are passed through a low-pass filter with impulse response g resulting in a convolution of the two:

$$y[n] = (x * g)[n] = \sum_{k=-\infty}^{\infty} x[k]g[n-k] \tag{1}$$

Here, k represents the samples of original signal function x and n represents the filtered coefficients of input signal. The signal is also decomposed simultaneously using a high-pass filter h. The outputs give the detail coefficients (from the high-pass filter) and approximation coefficients (from the low-pass). However, since half the frequencies of the signal have now been removed, half the samples can be discarded according to Nyquist's rule. The filter output of the low-pass filter g in the diagram above is then subsampled by 2 and further processed by passing it again through a new low-pass filter g and a high-pass filter h with half the cut-off frequency of the previous one, that is:

$$\begin{aligned} y_{\text{low}}[n] &= \sum_{k=-\infty}^{\infty} x[k]g[2n-k] \\ y_{\text{high}}[n] &= \sum_{k=-\infty}^{\infty} x[k]h[2n-k] \end{aligned} \tag{2}$$

This decomposition has halved the time resolution since only half of each filter output characterizes the signal. However, each output has half the frequency band of the input, so the frequency resolution has been doubled.

With the subsampling operator $\downarrow$, the above summation can be written more concisely.

$$\begin{aligned} y_{\text{low}} &= (x * g) \downarrow 2 \\ y_{\text{high}} &= (x * h) \downarrow 2 \end{aligned} \tag{3}$$

This decomposition is repeated to further increase the frequency resolution and the approximation coefficients decomposed with high- and low-pass filters and then down-sampled. The filter bank implementation of wavelets can be interpreted as computing the wavelet coefficients of a discrete set of child wavelets for a given mother wavelet $\Psi(t)$. In the case of the discrete wavelet transform, the mother wavelet is shifted and scaled by powers of two.

$$\psi_{j,k}(t) = \frac{1}{\sqrt{2^j}}\psi\left(\frac{t - k2^j}{2^j}\right) \tag{4}$$

where j is the scale parameter and k is the shift parameter, both which are integers.

$$T_{m,n} = \int_{-\infty}^{\infty} x(t)\psi_{m,n}(t)\mathrm{d}t \tag{5}$$

3.4 Discrete Fourier Transform

The (2D) Fourier transform is a very conventional instrument in image processing. It is the extension of the well-known Fourier transform for signals which decomposes a signal into a sum of sinusoids. So, the Fourier transform gives information about the frequency content of the image. The DFT of an image f of size $M \times N$ is an image of same size defined as:

$$F(u, v) = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} f(m, n)e^{-j2\pi\left(\frac{um}{M} + \frac{vn}{N}\right)} \tag{6}$$

where $f(m, n)$ is an input 2D image of size $(M * N)$ with m and n as the spatial domain variables and $F(u, v)$ is the frequency domain transformed image after applying the 2D-Discrete Fourier Transform which includes u and v as the frequency domain variables.

3.5 Proposed Algorithm

1. Take input fingerprint image.
2. Preprocessing the image enhancing the quality of image for better accuracy of classification.
   - 2.1 Normalize the image to the resolution of 256 × 256 for better performance.
   - 2.2 Apply weighted Gaussian filter to remove noise and smoothen the image of size (7 × 7).
   - 2.3 Perform histogram equalization to enhance the contrast.
   - 2.4 Apply segmentation of contrast enhanced image using global thresholding method to isolate the area of interest for further computation of image.
3. Apply skeletonization on the above preprocessed image for sharpening the features of the fingerprint using morphological image processing operations.

3.1. Erode the preprocessed image for getting the sharpened fingerprint ridges.
3.2. Dilate the eroded image for restructuring the image ridges
3.3. Apply median smoothing filter for smoothing all the ridges of the fingerprint.
3.4. Apply global thresholding to the morphologically processed of fingerprint for better fingerprint

4. Apply 2D-DWT (up to Level-4)/2D-DFT for extracting the fingerprint features in its corresponding structures.
5. Train the SVM classifier with some of the samples from the collected dataset for training purpose.
6. Apply SVM classifier on remaining fingerprint images from the dataset for testing purpose

4 Results

To make sure that the suggested system is appropriate for the age group classification from fingerprints, this study provides a detailed analysis of the classification metrics. The performance of the suggested system along with comparison for these performances of all these various transforms is demonstrated in statistics. As per the study there is no standard database which contains fingerprints of various age groups among male and female.

Therefore, as shown in Table 1, a gender-wise and age group-wise real time database has been created for conducting experiment on age classification based on fingerprints. In the database, fingerprints data is collected among nine different age groups from the various urban areas located in Aurangabad district of Maharashtra state in India. The database is spread among nine age groups ranging from (1 to 10) to (81 to 90). The acquisition of the fingerprint was made by SecuGen Hamster plus optical fingerprint sensor manufactured by nitgen biometric solution [with interface USB 2.0] with the 8-bit image size as 260 × 300 pixels.

Table 1 Age group categorization with respect to gender

Age group	Male	Female
1–10	40	10
11–20	350	260
21–30	320	120
31–41	380	140
41–50	150	70
51–60	70	50
61–70	210	250
71–80	240	270
81–90	25	–

Table 1 describes that; the database is created with a total of 2877 images including 1761 images of the male category and 1116 images of the female category from different age groups. Every person's right thumb's 10 samples have been collected for this research purpose. Here, 10 samples per person are for better accuracy purposes. Since the database is unbalanced among the various age groups and genders, the more robust results are observed.

The classification metrics included in this study for analysing the results are as follows:

$$\text{Accuracy} = (\text{True positives} + \text{True Negatives})/(\text{True positives} + \text{True negatives} + \text{False positives} + \text{False negatives}) \quad (7)$$

$$\text{Precision} = \text{True positives}/(\text{True positives} + \text{False positives}) \quad (8)$$

$$\text{Recall} = \text{True Positives}/(\text{True Positives} + \text{False Negatives}) \quad (9)$$

$$F1\ \text{Score} = 2 * (\text{Precision} * \text{Recall})/(\text{Precision} + \text{Recall}) \quad (10)$$

The classification report of 2D-DFT is shown in Table 2. It is observed very evidently that 2D-DFT transform has responded only for the female age group of 11–20 and male age group of 11–20 with minor precision, recall and $f1$-score for the available database. For remaining age groups across both the genders 2D-DFT has given 0 response with respect to precision, recall and $f1$-score which assures that this transform is not capable of providing better accuracy of age group classification.

In coordination to the results obtained through Table 2, the graphical representation of 2D-DFT transform's classification metrics analysis is shown in Fig. 1. As stated above, the metrics have only responded to both male and female age group from 11 to 20, holding remaining age groups at 0.

Since it is illustrated in Table 2 that the precision, recall and $f1$-score along with the accuracy for the 2D-DFT are not satisfactory, these findings are corresponded with the extended classification metrics such as macro average and weighted average of the precision, recall and $f1$-score across various age groups. As shown in Table 3, these extended metrics proven to be very weak throughout the entire database of fingerprint samples remarking macro average of precision as 0.04, and that of recall as 0.10 while for $f1$-score as merely 0.07, whereas weighted average of precision as 0.24, and that of recall as 0.38 while for $f1$-score as 0.28. These results debar 2D-DFT for an efficient age group classification based on fingerprint and these results are demonstrated in Fig. 2 as well.

The confusion matrix for 2D-DWT level-2 is depicted in Table 4. It is perceptible in this confusion matrix that 2D-DWT level-2 has not classified all the fingerprint samples correctly. The values shown on the diagonal line in the matrix indicate the true positive ratio of sample fingerprints across different age groups as mentioned in Table 1. According to the confusion matrix shown in Table 4, the 10 fingerprint

Table 2 Classification report of 2D-DFT

	Precision	Recall	*f*1-score	Support
Female age group 1–10	0	0	0	10
Female age group 11–20	0.43	0.45	0.44	249
Female age group 21–30	0.00	0.00	0.00	120
Female age group 31–40	0.00	0.00	0.00	140
Female age group 41–50	0.00	0.00	0.00	70
Female age group 51–60	0.00	0.00	0.00	50
Female age group 61–70	0.00	0.00	0.00	250
Female age group 71–80	0.00	0.00	0.00	270
Male age group 1–10	0.00	0.00	0.00	40
Male age group 11–20	0.36	0.80	0.50	310
Male age group 21–30	0.00	0.00	0.00	320
Male age group 31–40	0.00	0.00	0.00	380
Male age group 41–50	0.00	0.00	0.00	150
Male age group 51–60	0.00	0.00	0.00	70
Male age group 61–70	0.00	0.00	0.00	210
Male age group 71–80	0.00	0.00	0.00	240
Male age group 81–90	0.00	0.00	0.00	21

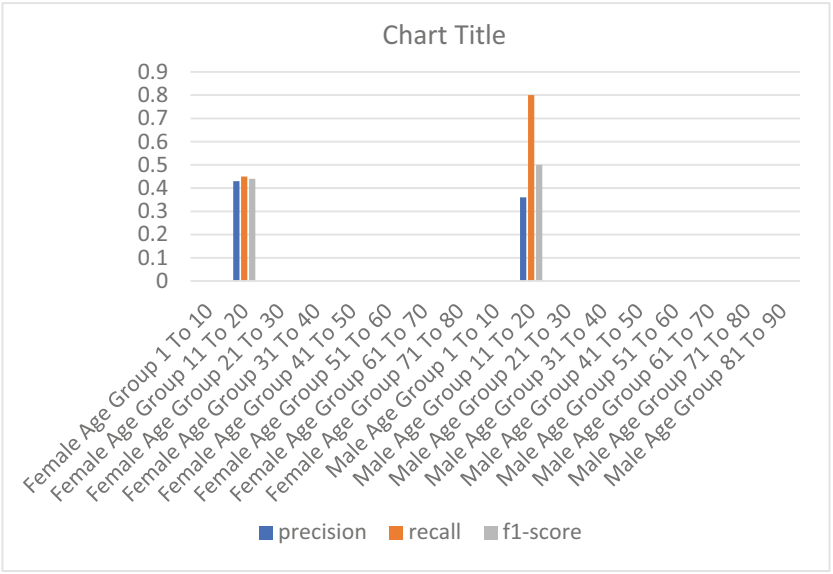


Fig. 1 Classification metrics analysis of 2D-DFT level with emphasis on precision, recall and *f*1-score metrics

Table 3 Extended classification metrics analysis of 2D-DFT

	Precision	Recall	$f1$-score	Support
Accuracy			0.38	2877
Macro avg	0.04	0.10	0.07	2877
Weighted avg	0.24	0.38	0.28	2877

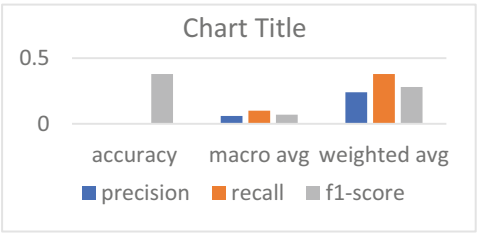


Fig. 2 Extended classification metrics analysis of 2D-DFT using macro avg and weighted avg of precision, recall and $f1$-score

samples from the female age group 1 to 10, are not all correctly classified as indicated by the first true positive value as 0 on diagonal line. Also, 2D-DWT level-2 transform has correctly classified 212 fingerprints out of the 260 samples from female age group 11 to 20. Similarly remaining age groups classification results are demonstrated in the above confusion matrix of Table 4.

Table 4 Confusion matrix of 2D-DWT level-2

0	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	212	0	0	0	0	7	5	0	27	1	5	0	0	0	2	0
0	9	34	0	0	0	3	17	0	22	10	23	0	0	0	1	0
0	18	0	9	0	0	5	20	0	50	10	26	1	0	0	1	0
0	3	0	0	2	0	2	17	0	20	0	25	0	0	0	1	0
0	0	0	0	0	14	8	9	0	0	0	19	0	0	0	0	0
0	2	1	0	0	0	116	49	0	13	4	25	0	0	0	6	0
0	0	0	0	0	0	2	225	0	5	3	16	0	0	0	1	0
0	22	1	0	0	0	0	0	9	7	1	0	0	0	0	0	0
0	13	0	0	0	0	2	1	0	309	5	20	0	0	0	0	0
0	10	0	0	0	0	4	28	0	14	225	37	0	0	0	2	0
0	11	0	0	0	0	0	24	0	15	6	317	0	0	0	0	0
0	6	0	0	0	0	4	7	0	25	19	75	11	0	0	3	0
0	1	0	0	0	0	10	0	0	11	4	38	0	2	0	3	0
0	8	0	0	0	0	7	61	0	21	9	23	0	0	63	14	0
0	2	0	0	0	0	7	62	0	10	8	20	0	0	0	123	0
0	0	0	0	0	0	1	19	0	0	0	0	0	0	0	1	0

A detailed classification report for 2D-DWT level-2 is represented in Table 5, where the average precision ratio, recall and $f1$-score for each age group is given along with the respective support sample's count in the last column. All the three-classification metrics range from 0 (indicating failure) to 1 (indicating 100% success ratio). Therefore, from the precision, recall and $f1$-score values in Table 5, it is verified that the performance of 2D-DWT level-2 is not satisfactory than the remaining 2D-DWT variants but better than 2D-DFT transform.

Figure 3 represents the classification report for 2D-DWT level-2, where the average precision, recall and $f1$-score for each age group is specified on horizontal axis along with the respective age group's support count on vertical axis. Therefore, from the precision, recall and $f1$-score values in Fig. 3, it is verified that the performance of 2D-DWT level-2 is average than the remaining 2D-DWT variants.

Table 6 shows an extended classification metrics analysis for 2D-DWT level-2 which includes the computations of macro average and weighted average of precision, recall and $f1$-score between the existing age groups. Since the collected real time database is unbalanced among various age groups, the computed accuracy needs to be reviewed and re-verified with the help of macro average and weighted average supportive classification metrics. Therefore, from Table 6, it is quantified and shown that 2D-DWT level-2 transform gives merely average accuracy for fingerprint-based age group classification.

Table 5 Classification report of 2D-DWT level-2

	Precision	Recall	$f1$-score	Support
Female age group 1–10	0	0	0	10
Female age group 11–20	0.65	0.82	0.72	259
Female age group 21–30	0.94	0.29	0.44	119
Female age group 31–40	1	0.06	0.12	140
Female age group 41–50	1	0.03	0.06	70
Female age group 51–60	1	0.28	0.44	50
Female age group 61–70	0.65	0.54	0.59	216
Female age group 71–80	0.41	0.89	0.57	252
Male age group 1–10	1	0.23	0.37	40
Male age group 11–20	0.56	0.88	0.69	350
Male age group 21–30	0.74	0.7	0.72	320
Male age group 31–40	0.47	0.85	0.61	373
Male age group 41–50	0.92	0.07	0.14	150
Male age group 51–60	1	0.03	0.06	69
Male age group 61–70	1	0.31	0.47	206
Male age group 71–80	0.78	0.53	0.63	232
Male age group 81–90	0	0	0	21

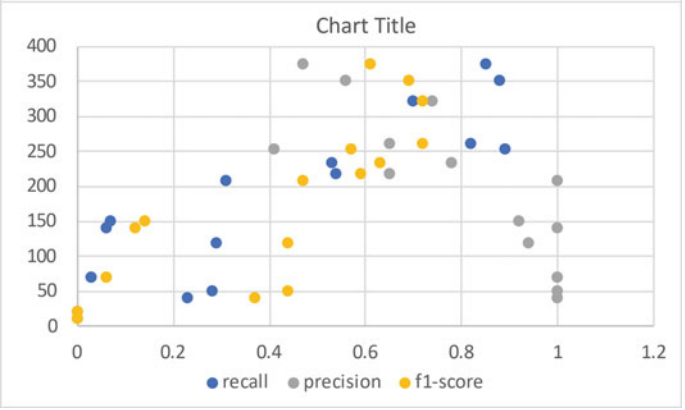


Fig. 3 Classification report of 2D-DWT level-2

Table 6 Extended classification metrics analysis 2D-DWT2

Accuracy			0.58	2877
Macro avg	0.71	0.38	0.39	2877
Weighted avg	0.7	0.58	0.53	2877

An extended classification metrics analysis for 2D-DWT level-2 which includes the computations of macro average and weighted average of precision, recall and f1-score between the existing age groups is shown in Fig. 4. It has made very pure that the performance of precision is better over that of recall and f1-score metrics.

The confusion matrix for 2D-DWT level-3 is depicted in Table 7. It is noticeable in this confusion matrix that 2D-DWT level-3 has not classified all the fingerprint samples acceptably. The values shown on the center diagonal line in the matrix indicate the true positive ratio of sample fingerprints across different age groups as mentioned in Table 1. According to the confusion matrix shown in Table 7, the 10 fingerprint samples from the female age group 1 to 10, are not all correctly classified

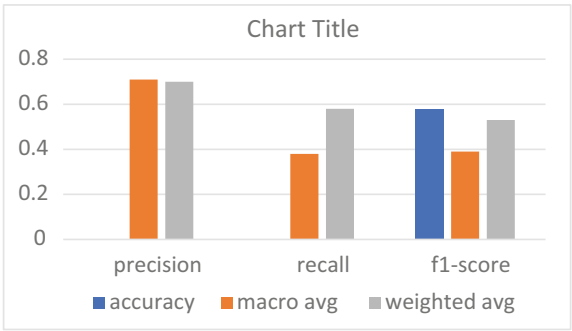


Fig. 4 Extended classification metrics analysis DWT2 using macro avg and weighted avg of precision, recall and f1-score

as the first true positive value is 0 on the center diagonal line. Also, 2D-DWT level-3 transform has correctly classified 170 fingerprints out of the 260 samples from female age group 11 to 20. Similarly remaining age groups classification results are demonstrated in the above confusion matrix of Table 7.

A complete classification report for 2D-DWT level-3 is represented in Table 8, where the average precision ratio, recall and $f1$-score for each age group is given. All the three-classification metrics range from 0 (indicating failure) to 1 (indicating 100% success ratio). Therefore, from the precision, recall and $f1$-score values in Table 8, it is verified that the performance of 2D-DWT level-3 is good with respect to the precision ratio but poor with respect to the recall and $f1$-score making this transform weaker than the remaining 2D-DWT variants but to some extent better than the 2D-DFT transform. From Table 8, it is stated that for the female age group 31–40, 41–51 and 51–60 as well as for male age group 1–10 and 61–70 the precision ratio performed very well with the 100% classification performance. It can be noticed from Table 8 that 2D-DWT level-3 transform failed to classify female age groups 1–10, male age groups 41–50, 51–60 and 81–90 giving their corresponding precision, recall and $f1$-score values as 0.

Then the inclusive classification metrics analysis for 2D-DWT level-3 is represented in Fig. 5, where the average precision, recall and $f1$-score for each age group is specified on the vertical axis and the respective age groups are represented on the horizontal axis. Here, all the three metrics have noted astonishing output of 1 only for female age groups 31–40, 41–50, 51–60, and male age groups 1–10 and 61–70,

Table 7 Confusion matrix 2D-DWT3

0	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	170	1	0	0	0	2	5	0	56	5	20	0	0	0	0	0
0	10	22	0	0	0	3	19	0	24	11	29	0	0	0	1	0
0	15	2	3	0	0	2	18	0	51	16	32	0	0	0	1	0
0	2	0	0	1	0	2	17	0	20	0	27	0	0	0	1	0
0	0	2	0	0	7	6	11	0	0	3	21	0	0	0	0	0
0	8	1	0	0	0	91	59	0	11	4	32	0	0	0	10	0
0	0	0	0	0	0	3	210	0	1	5	26	0	0	0	7	0
0	16	0	0	0	0	0	0	6	15	1	2	0	0	0	0	0
0	29	0	0	0	0	0	3	0	246	15	56	0	0	0	1	0
0	11	1	0	0	0	2	40	0	30	140	90	0	0	0	6	0
0	12	1	0	0	0	1	29	0	29	28	271	0	0	0	2	0
0	5	0	0	0	0	3	7	0	21	23	87	0	0	0	4	0
0	1	0	0	0	0	4	7	0	16	5	33	0	0	0	3	0
0	5	0	0	0	0	5	69	0	21	9	36	0	0	42	19	0
0	2	0	0	0	0	9	80	0	8	10	33	0	0	0	90	0
0	0	0	0	0	0	0	19	0	0	0	1	0	0	0	1	0

Table 8 Classification report of 2D-DWT level-3

	Precision	Recall	$f1$-score	Support
Female age group 1–10	0	0	0	10
Female age group 11–20	0.57	0.66	0.61	259
Female age group 21–30	0.73	0.18	0.3	119
Female age group 31–40	1	0.02	0.04	140
Female age group 41–50	1	0.01	0.03	70
Female age group 51–60	1	0.14	0.25	50
Female age group 61–70	0.68	0.42	0.52	216
Female age group 71–80	0.35	0.83	0.5	252
Male age group 1–10	1	0.15	0.26	40
Male age group 11–20	0.45	0.7	0.55	350
Male age group 21–30	0.51	0.44	0.47	320
Male age group 31–40	0.34	0.73	0.46	373
Male age group 41–50	0	0	0	150
Male age group 51–60	0	0	0	69
Male age group 61–70	1	0.2	0.34	206
Male age group 71–80	0.62	0.39	0.48	232
Male age group 81–90	0	0	0	21

gathering the highest possible classification score among all the other age groups. It is very clear from Fig. 5 that 2D-DWT level-3 transform performance is worst among all other 2D-DWT variants considered for this experiment.

Table 9 shows an extended classification metrics analysis for 2D-DWT level-3 which includes the computations of macro average and weighted average of precision, recall and $f1$-score between the existing age groups. Therefore, from Table 9, it is computed and shown that for the utilized unbalanced database among various age groups, 2D-DWT level-3 transform gives even below average accuracy for fingerprint-based age group classification. And it is also shown in Fig. 6 for better understanding purpose in graphical manner.

It is shown in Fig. 6 for better understanding purpose in graphical manner that 2D-DWT level-3 transform gives even below average accuracy for fingerprint-based age group classification.

The confusion matrix for 2D-DWT level-4 is depicted in Table 11. It is clearly visible in this confusion matrix that 2D-DWT level-4 has classified all the fingerprint samples more accurately than 2D-DWT level-3 and 2D-DWT level-2. The values shown on the diagonal line in the matrix indicate the true positive ratio of sample fingerprints across different age groups as mentioned in Table 1. According to the confusion matrix shown in Table 11, the 10 fingerprint samples from the female age group 1 to 10, are all correctly classified as indicated by the first true positive value on the center diagonal line. Also, 2D-DWT level-4 has correctly classified

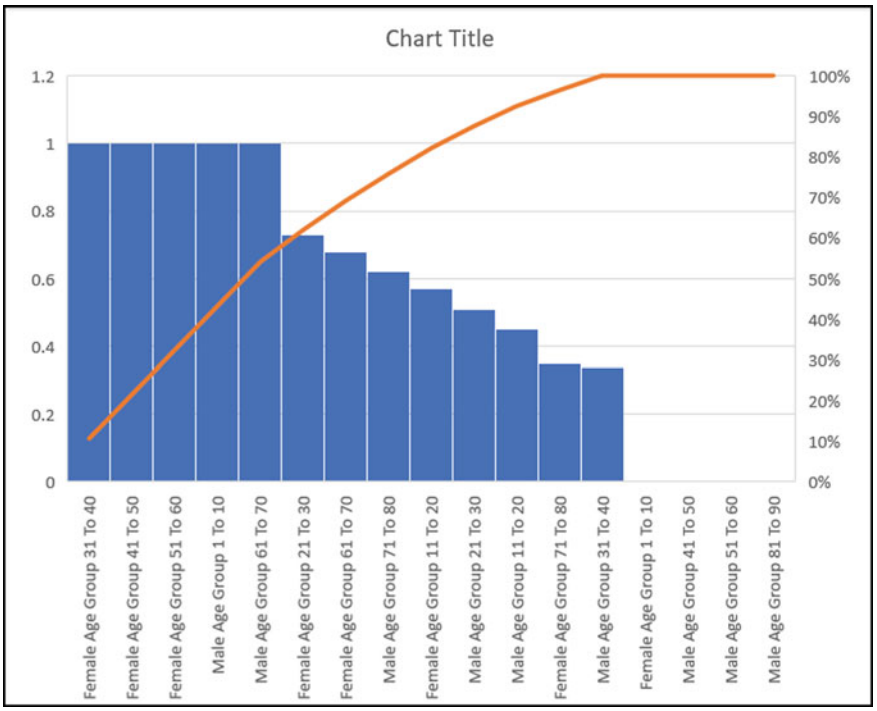


Fig. 5 Classification report of 2D-DWT level-3

Table 9 Extended classification metrics analysis 2D-DWT level-3

	Precision	Recall	f1-score	Support
Accuracy			0.45	2877
Macro avg	0.54	0.29	0.28	2877
Weighted avg	0.55	0.45	0.4	2877

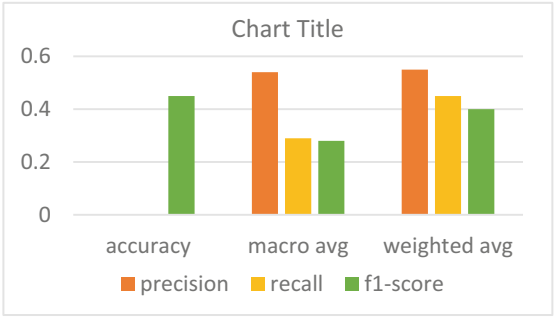


Fig. 6 Extended classification metrics analysis DWT3 using macro avg and weighted avg of precision, recall and f1-score

259 fingerprints out of the 260 samples from female age group 11 to 20. Similarly remaining age groups classification results are demonstrated in the above confusion matrix of Table 11.

Table 10 illustrates the empirical results of 2D-DWT level-4 transform, in the form of age group-wise distribution of fingerprint samples and then showing the correctly classified samples in the last column of Table 10 (Table 11).

A detailed classification report for 2D-DWT level-4 is represented in Table 12, where the average precision ratio, recall and $f1$-score for each age group is given along with the respective support sample's count in the last column. All the three-classification metrics range from 0 (indicating failure) to 1 (indicating 100% success ratio). Therefore, from the precision, recall and $f1$-score values in Table 12, it is verified that the performance of 2D-DWT level-4 is best among the remaining 2D-DWT variants and 2D-DFT transform.

Figure 7 represents the classification report for 2D-DWT level-4, where the average precision, recall and $f1$-score for each age group is specified along with the respective age group's support count on horizontal axis and the concerned age groups are represented on the vertical axis. Therefore, from the precision, recall and $f1$-score values in Fig. 7 it is verified that the performance of 2D-DWT level-4 is higher than the remaining 2D-DWT variants and 2D-DFT transform.

Then, the comprehensive classification metrics analysis for 2D-DWT level-4 is represented in Fig. 8, where the average precision, recall and $f1$-score for each age group is specified on the vertical axis and the respective age groups are represented on the horizontal axis. Here, all the three metrics have noted astonishing output of 1, inferring the highest possible classification score. It is very clear from Fig. 8 that 2D-DWT level-4 transform performance is best among all other considered for this experiment.

Table 10 Empirical results on confusion matrix 2D-DWT level-4

Age group	Actual samples	Correctly classified samples
Female age group 1–10	10	10
Female age group 11–20	259	259
Female age group 21–30	119	119
Female age group 31–40	140	140
Female age group 41–50	70	70
Female age group 51–60	50	50
Female age group 61–70	216	215
Female age group 71–80	252	252
Male age group 1–10	40	40
Male age group 11–20	350	350
Male age group 21–30	320	320
Male age group 31–40	373	373

Table 11 Confusion matrix 2D-DWT level-4

10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	259	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	119	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	140	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	70	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	50	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	215	1	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	252	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	40	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	350	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	320	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	373	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	150	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	69	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	206	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	232	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	21

Table 12 Classification report 2D-DWT level-4

	Precision	Recall	f1-score	Support
Female age group 1–10	1	1	1	10
Female age group 11–20	1	1	1	259
Female age group 21–30	1	1	1	119
Female age group 31–40	1	1	1	140
Female age group 41–50	1	1	1	70
Female age group 51–60	1	1	1	50
Female age group 61–70	1	1	1	216
Female age group 71–80	1	1	1	252
Male age group 1–10	1	1	1	40
Male age group 11–20	1	1	1	350
Male age group 21–30	1	1	1	320
Male age group 31–40	1	1	1	373
Male age group 41–50	1	1	1	150
Male age group 51–60	1	1	1	69
Male age group 61–70	1	1	1	206
Male age group 71–80	1	1	1	232
Male age group 81–90	1	1	1	21

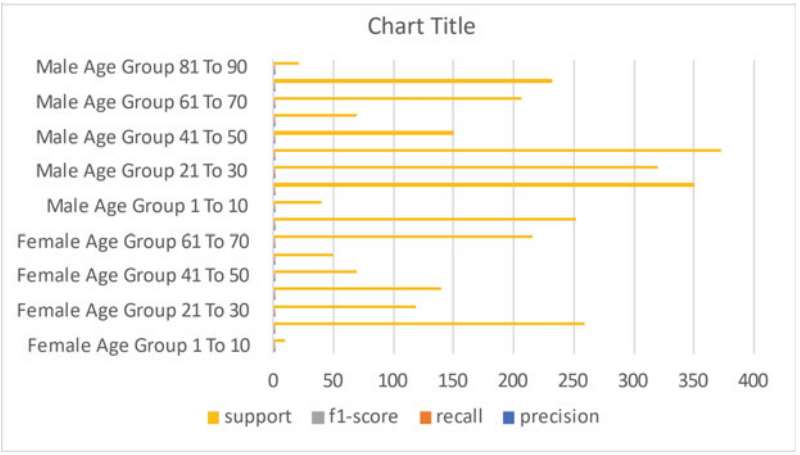


Fig. 7 Classification report of 2D-DWT level-4 with emphasis on support metric

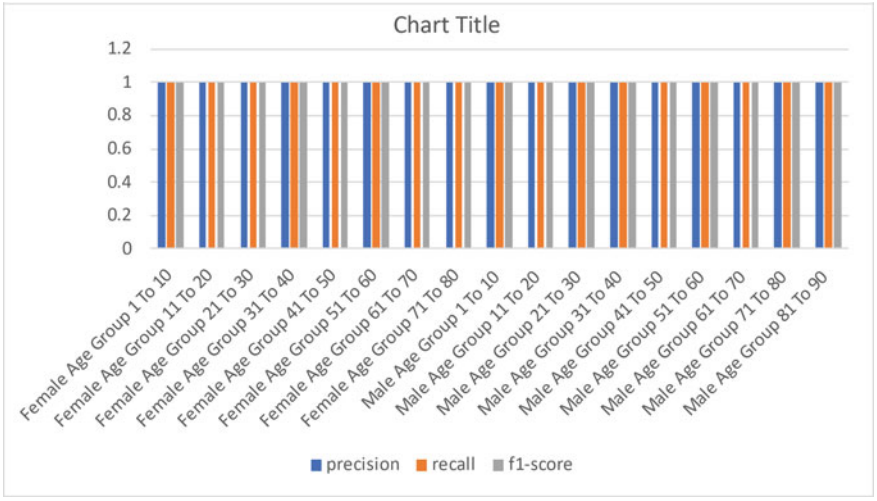


Fig. 8 Classification metrics analysis of 2D-DWT level-4 with emphasis on precision, recall and $f1$-score metrics

Table 13 shows an extended classification metrics analysis for 2D-DWT level-4 which includes the computations of macro average and weighted average of precision, recall and $f1$-score between the existing age groups. Since our database is unbalanced among various age groups, the computed accuracy needs to be reviewed and re-verified with the help of macro average and weighted average supportive classification metrics. Therefore, from Table 13, it is quantified and shown that 2D-DWT

Table 13 Extended classification metrics analysis of 2D-DWT level-4

	Precision	Recall	$f1$-score	Support
Accuracy			1	2877
Macro avg	1	1	1	2877
Weighted avg	1	1	1	2877

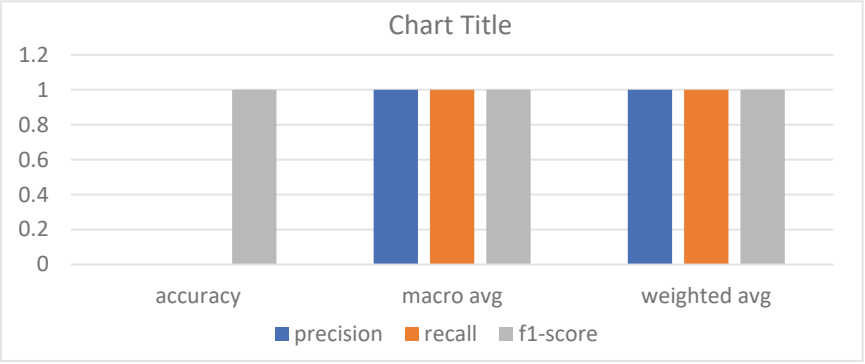


Fig. 9 Extended classification metrics analysis DWT4 using macro avg and weighted avg of precision, recall and $f1$-score

level-4 transform gives better accuracy and performance for fingerprint-based age group classification.

Figure 9 depicts graphical representation of the extended classification metrics analysis for 2D-DWT level-4 which includes the computations of macro average and weighted average of precision, recall and $f1$-score between the existing age groups.

As shown in Table 14, it has been determined that when 2D-DWT level-4 was applied then the accuracy gained was 99.96% and classification finished in 71.004 s. When DWT3 was applied then the accuracy gained was 45.15% and classification finished in 34.985 s. When DWT2 was applied then the accuracy gained was 58.08% and classification finished in 107.015 s. When DFT was applied then the accuracy gained was 38% and classification finished in 212.908 s and the most accurate of these four transforms was the DWT Level-4 which classified all the age groups using the fingerprint images with an astonishing accuracy 99.96. The accuracy is further verified and supported by performing detailed analysis of all classification metric including macro average and weighted average of precision, recall and $f1$-score metrics.

Table 14 Comparison of 2D-DWT2, 2D-DWT3, 2D-DWT4 and 2D-DFT based classification performance

Transform	Accuracy	Precision	Recall	$f1$-score
2D-DWT level-4	0.9996	0.9996	0.9996	0.9996
2D-DWT level-3	0.4515	0.4515	0.4515	0.4515
2D-DWT level-2	0.5808	0.5808	0.5808	0.5808
2D-DFT	0.3827	0.3827	0.3827	0.3827

5 Conclusion

The age group classification using fingerprint is an important research area with numerous potential applications in various fields. In this study, it has been concluded that 2D-DWT level-4 has correctly classified all the age groups with 99.96% accuracy in 71.004 s as compare to that of 2D-DWT level-3 with 45.15% and 2D-DWT level-2 with 58.08%. When 2D-DFT was applied, the accuracy gained was 38%. Therefore, the most accurate of these four transforms has been the DWT Level-4. The accuracy is further reviewed by performing detailed analysis of all classification metrics including macro average and weighted average of precision, recall and $f1$-score. Overall, the motivation for research in the area of age group recognition using fingerprint is driven by the potential benefits it can provide in various fields such as biometric identification, law enforcement, healthcare, personalization, and security. Further research in this area can lead to the development of advanced age group estimation models that can significantly improve the accuracy and reliability of biometric identification systems, enhance personalization, and improve the quality of healthcare services.

5.1 Future Scope

The future scope of age group detection using fingerprint is promising as it has numerous potential applications in various fields such as law enforcement, healthcare, and biometrics. One possible future direction is the development of more advanced and accurate age group and gender recognition models using a fingerprint dataset of people from rural areas with very rough skin surface due to their daily physical works as the quality of such people's fingerprint samples will be surely poor than the database collected for this research work from urban areas and the development of a robust age group classification model for such fingerprint samples will strengthen confidence of utilizing it for the latent fingerprint samples too for criminal investigation purpose in future. Therefore, there is a need for more comprehensive and diverse datasets for age group detection using fingerprint, particularly in terms of age and gender ranges and ethnic diversity. The development of large-scale and

diverse datasets can help improve the generalization and reliability of age and gender estimation models to real-world scenarios.

References

1. Basavaraj Patil GV, Rafi M (2015) Human age estimation through fingerprint. Int J Innov Res Comput Commun Eng 3(4). ISSN: (Online): 2320-9801
2. Abdullah HA (2012) Fingerprint identification system using neural networks. Nahrain Univ Coll Eng J (NUCEJ) 15(2):234–244
3. Gornale SS, Hangarge M, Pardeshi R, Kruthi R (2015) Haralick feature descriptors for gender classification using fingerprints: a machine learning approach. Int J Adv Res Comput Sci Softw Eng 5(9)
4. Gornale S, Patil A, Veersheety C (2016) Fingerprint based gender identification using discrete wavelet transform and Gabor filters. Int J Comput Appl (0975-8887) 152(4)
5. Munir MU, Javed MY (2004) Fingerprint matching using Gabor filters. In: National conference on emerging technologies
6. Valdes-Ramirez D, Medina-Pérez MA, Monroy R, Loyola-González O, Rodríguez-Ruiz J, Morales A, Herrera F (2019) A review of fingerprint feature representations and their applications for latent fingerprint identification: trends and evaluation. IEEE Access 7(1):48484–48499
7. Narayanan A, Sajith K (2019) Gender detection and classification from fingerprints using pixel count. In: Proceedings of the international conference on systems, energy & environment (ICSEE) 2019, GCE Kannur, Kerala, July
8. Gnanasivam P, Muttan S (2012) Estimation of age through fingerprints using wavelet transform and singular value decomposition. Int J Biometr Bioinform (IJBB) 6(2)
9. Gnanasivam P, Muttan S (2012) Fingerprint gender classification using wavelet transform and singular value decomposition. Int J Comput Sci Issues 9(2)
10. Zhang B-L, Zhang H, Ge SS (2004) Face recognition by applying wavelet sub band representation and kernel associative memory. IEEE Trans Neural Netw 15(1):166–177
11. Win KN, Li K, Chen J, Viger PF, Li K (2020) Fingerprint classification and identification algorithms for criminal investigation: a survey. Futur Gener Comput Syst 110:758–771
12. Yang W, Wang S, Hu J, Zheng G, Valli C (2019) Security and accuracy of fingerprint-based biometrics: a review. Symmetry 11(2):141
13. Tarare S, Anjikar A, Turkar H (2015) Fingerprint based gender classification using DWT transform. In: International conference on computing communication control and automation (ICCUBEA), 26–27 Feb 2015
14. Akbar S, Ahmad A, Hayat M (2014) Identification of fingerprint using discrete wavelet transform in conjunction with support vector machine. IJCSI Int J Comput Sci Issues 11(d5):1. ISSN (Print): 1694-0814 | ISSN (Online): 1694-0784
15. Ito K, Aoki T (2018) Recent advances in biometric recognition. ITE Trans Media Technol Appl 6(1):64–80

Exploring the Potential Use of Infrared Imaging in Medical Diagnosis: Comprehensive Framework for Diabetes and Breast Cancer Screening



Asok Bandyopadhyay, Himanka Sekhar Mondal, Barnali Pal, Bivas Dam, and Dipak Chandra Patranabis

Abstract This research paper comprehensively explores the potential use of infrared (IR) imaging in medical diagnosis, focusing on diabetes and breast cancer screening. The study builds on earlier work on region selection for infrared imaging and addresses the need for automated image processing and analysis techniques. The data collection for diabetes was done at SRM Medical College and Hospital and Cachar Cancer Hospital and Research Centre (CCHRC), whereas breast cancer screening data were collected at CCHRC. The result demonstrates the importance of the ear cavity region in diabetes screening and the importance of rotational thermography and dynamic temperature analysis in infrared image-based screening systems for breast cancer screening. This paper also presents advances in data collection techniques, image analysis, and machine learning algorithms to improve the accuracy and efficiency of the proposed framework.

Keywords Infrared imaging · Diabetes and breast cancer screening · Medical thermography · Temperature quantification · Machine learning

A. Bandyopadhyay (✉) · H. S. Mondal · B. Pal
ICT and Servics Group, Centre For Development of Advanced Computing, Kolkata, India
e-mail: asok.bandyopadhyay@cdac.in

H. S. Mondal
e-mail: himanka.mondal@cdac.in

B. Pal
e-mail: barnali.pal@cdac.in

B. Dam · D. C. Patranabis
Department of Instrumentation and Electronics, Jadavpur University, Kolkata, India
e-mail: b_dam@rediffmail.com

D. C. Patranabis
e-mail: dipakchandrapatranabis@gmail.com

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2023

S. Shakya et al. (eds.), *Fourth International Conference on Image Processing and Capsule Networks*, Lecture Notes in Networks and Systems 798,
https://doi.org/10.1007/978-981-99-7093-3_27

1 Introduction

This paper reports a comprehensive approach combining infrared (IR) imaging for diabetes screening tools and breast cancer screening systems. The selection of the ear cavity region as an essential body region for diabetes screening, as well as advances in image processing, data acquisition, and machine learning, contributes to the accuracy and effectiveness of the proposed system. The findings of this study have important clinical implications and provide insights into the potential of infrared imaging in the diagnosis of type 2 diabetes. For breast cancer detection, the data acquisition technique used four pilot studies (PS1, PS2, PS3, and FS) to acquire infrared images for breast cancer screening. Each pilot study employed specific settings and modifications to simplify data collection, analysis, and optimization for breast cancer detection.

The research consisted of two main parts: selecting regions of interest for infrared imaging in diabetes-related equipment using the most influential body parts and developing a non-contact, non-invasive breast imaging method using infrared technology. In the first part, the study builds on previous work on region selection for infrared imaging. Various research groups have investigated how infrared imaging can assess the temperature of different body regions, such as the surface of the eye, foot, cornea, toes, and skin, for diabetes-related applications. From previous studies, Sodi et al., Johansson and Roback investigated the feasibility of thermal imaging-based ocular surface temperature in patients with diabetic foot disease and non-proliferative diabetic retinopathy (NPDR) [1, 2]. Nishide et al., Nandao et al. worked on using thermal imaging and ultrasound techniques to identify underlying inflammation of foot calluses [3]. To elucidate the typical pattern features of diabetic autonomic neuropathy in the toes, Fujiwara et al. perform treadmill walking stress testing and thermal imaging [4]. Fushimi et al. measurement of skin temperature in type 2 diabetic patients using thermal imaging [5] S. Sivanandam et al. investigation of abnormal vasovagal responses in the hand [6]. It acts on different body parts such as the face, forehead, hands, feet, and neck [7]. Sivanandam et al. investigated potential of infrared thermography (IR) in the diagnosis of type 2 diabetes [7] in their early works. The authors have previously discussed the selection of the ear cavity region as a critical body region for diabetes screening, with advances in image processing, data acquisition, and machine learning contributing to its accuracy and validity.

In this paper, the authors emphasize the importance of automated image processing and analysis methods to improve the accuracy and efficiency of infrared imaging for diabetes and breast cancer screening. The collaboration between CDAC Kolkata, SRM Medical College Hospital and Cachar Cancer Hospital, has contributed to advances in data acquisition techniques, image analysis, and pattern-matching algorithms. The first part of the study emphasizes diabetes screening using IR imaging techniques.

The second part of the study focused on detecting breast cancer using infrared imaging. Previously, various methods for detecting breast cancer using thermal imaging have been proposed. Ethard Tabakor et al. [8] introduced bispectral invariant

features and achieved high detection rates in their study [9]. They also compared color segmentation techniques and found that K-means clustering performs better than other methods [9]. Ramon et al. [10] developed a non-invasive tool for breast cancer diagnosis using a thermal imaging camera, using temperature and texture features with promising results [10]. Prakash et al. comparing K-means, fuzzy C-means, and EM algorithm, it is concluded that EM algorithm has the highest segmentation accuracy [11]. Venkataramani et al. [12] proposed a semi-automated method using thermal video imaging and achieved high sensitivity and specificity through morphological filtering and thresholding. Pramanik et al. proposed a deep learning-based level-set method for segmenting suspicious regions in chest thermograms and achieved high accuracy [13]. In brain tumor detection, Wu et al. [14] proposed a color-based K-means clustering segmentation method. Reviewed by Kandlikar et al., the current status, protocols, advantages, and limitations of infrared imaging techniques for breast cancer detection have been discussed [15]. These methods and studies demonstrate the potential of thermal imaging in breast cancer detection and provide insights into different methods and techniques of image processing and analysis [16]. The second part addresses challenges such as quantifying suitable temperature values for malignant diseases and the lack of qualified medical professionals for infrared imaging-based screening systems. To address the above issues, the research team developed a novel non-contact, non-invasive breast imaging method based on rotational thermography and dynamic temperature, which can detect any abnormalities found in the subjects. They propose a rotating thermal imaging device that can make the affected area more visible, improving the accuracy of breast cancer detection. A machine learning algorithm trains the extracted features and predicts normal versus abnormal values for the subjects.

2 Methods

2.1 Diabetes Screening Methodology

To detect diabetes, the team focused on capturing infrared images of different body parts, such as the face, neck, hands, legs, and ears. Imaging setup and data acquisition were performed at a hospital in the South Indian region. An Arduino Uno microcontroller with an integrated data acquisition system based on a temperature sensor was developed to account for changes in ambient temperature, which helps to normalize the effects of temperature changes caused by environmental factors.

Special precautions were taken during data collection to ensure accurate imaging results. Selected areas of the body are stripped naked and free of any metallic ornaments or objects. Subjects were allowed to acclimate for 15 min in an air-conditioned room at approximately 25.5 °C while maintaining controlled temperature, humidity, and airflow. This step aims to capture skin temperature changes in the range of 0.05–0.1 °C between the affected area and the surrounding or contralateral area.

Data collection was performed in two sessions: a fasting blood draw and a post-prandial (PP) blood draw with a two-hour interval between two. This allows data to be collected on approximately 12–13 subjects daily. Infrared image processing has employed various techniques, including manual, semi-automatic, and fully automatic region of interest (ROI) selection. Manual ROI selection involves cropping an ear template from an infrared image. In contrast, semi-automatic ROI selection allows the user to accept or reject the coordinate region suggested by the software. Fully automatic ROI selection uses image registration and feature extraction algorithms to select ROIs. Figure 1 shows the automatic selection of the ear cavity area by image processing technology, data acquisition, and analysis of the heat distribution in the diabetic "ear" area. Image processing algorithms such as thresholding and morphological operations are used to extract thermal information of the affected area. Figure 2 shows a flowchart of an image analysis algorithm using the image processing system of the diabetes screening system.

Image analysis techniques include thresholding to segment particles based on pixel intensity values. Virtual instrument software tracks regions of interest through shape-based object-tracking techniques. Save the temperature values extracted from the ROI for further analysis. Feature extraction techniques include using a generalized "Hough transform" algorithm to extract geometric shape features, which are then trained as geometric shape patterns. Statistical image analysis algorithms record mean temperature, standard deviation, median, mode, skewness, kurtosis, moments, root mean square (RMS), entropy, energy, and maximum temperature values for each infrared image as a function of the ROI. Rank the features according to the feature evaluation method based on Euclidean distance and information gain theory.

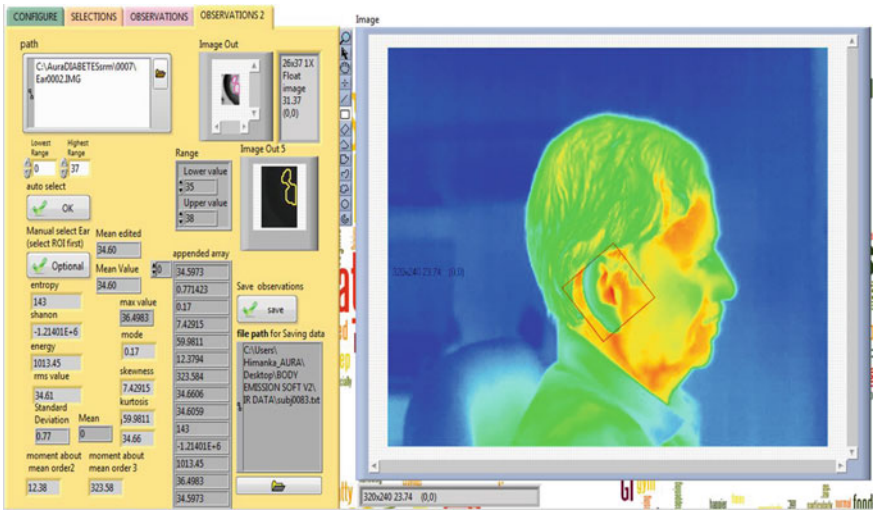


Fig. 1 Data collection and analysis of thermal profile of "ear" zone for diabetic patients

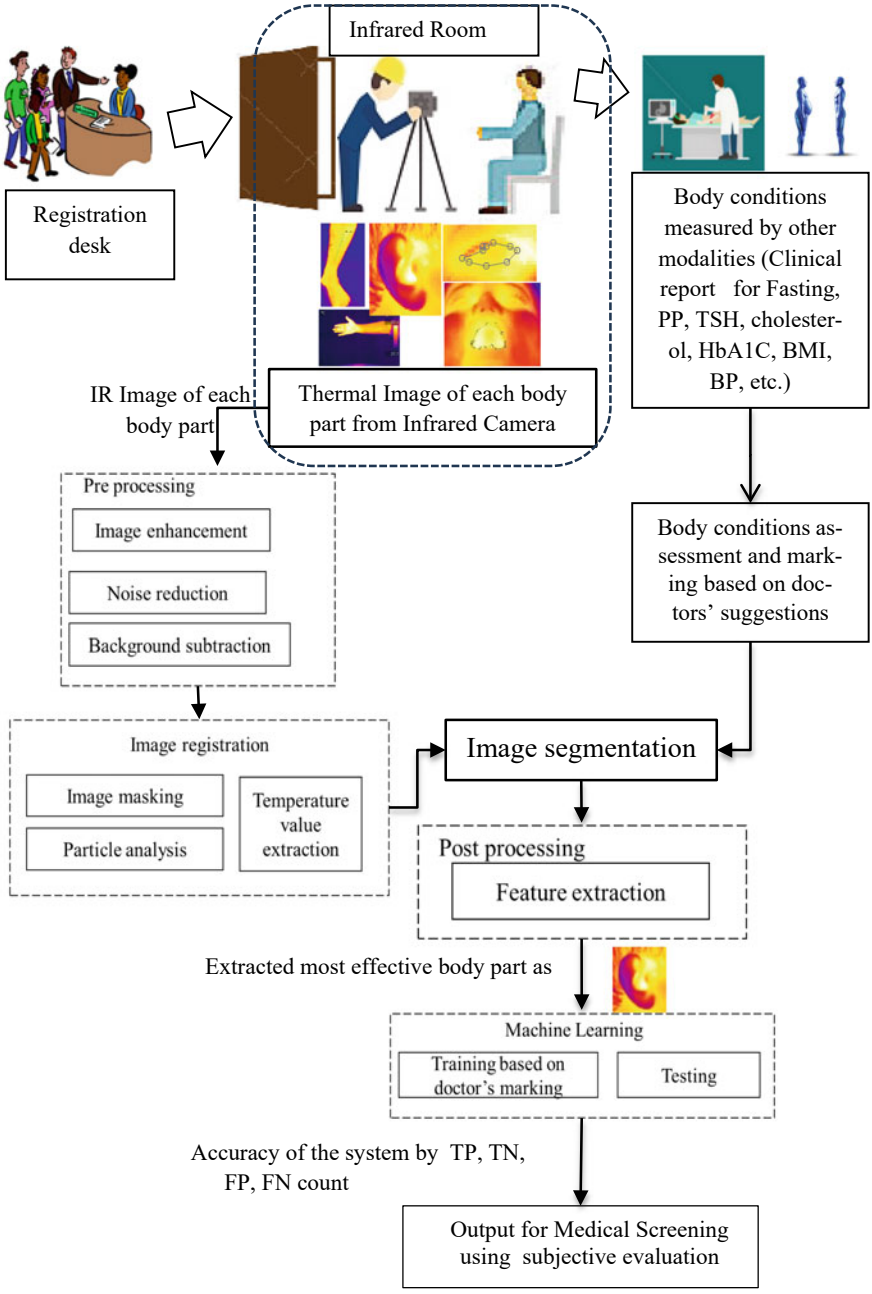


Fig. 2 Diagram for infrared-based image processing and machine vision system for diabetes screening

Machine learning-based classification was performed using the statistical features extracted from the thermal arrays as input. The top three features are selected using a feature ranking algorithm and trained using an SVM learning algorithm for classification and analysis. Other classification algorithms were also tested during the software development phase, including BPNN, RBF NN, and PNN. The SVM algorithm exhibits better and more consistent classification accuracy on image databases.

2.2 *Breast Cancer Screening Methodology*

In the case of breast cancer detection, the data collection techniques employed four pilot studies (PS1, PS2, PS3, and FS) to obtain IR images. Each pilot study utilized specific setups and modifications to optimize the data acquisition process. For example, PS1 and PS2 involved subjects seated in front of cameras with raised hands, while PS3 introduced a modified setup with the camera moving in a semi-circular arc to focus on one breast while covering the other with an infrared-proof barrier. The FS employed rotational thermography, with the patient's breasts positioned one at a time through a breast-shaped grooved hole for IR imaging. Figure 3 shows the breast cancer screening setup using rotational thermography.

Data analysis techniques played a crucial role in breast cancer screening using thermal imaging. The data analysis techniques were integrated into the system at

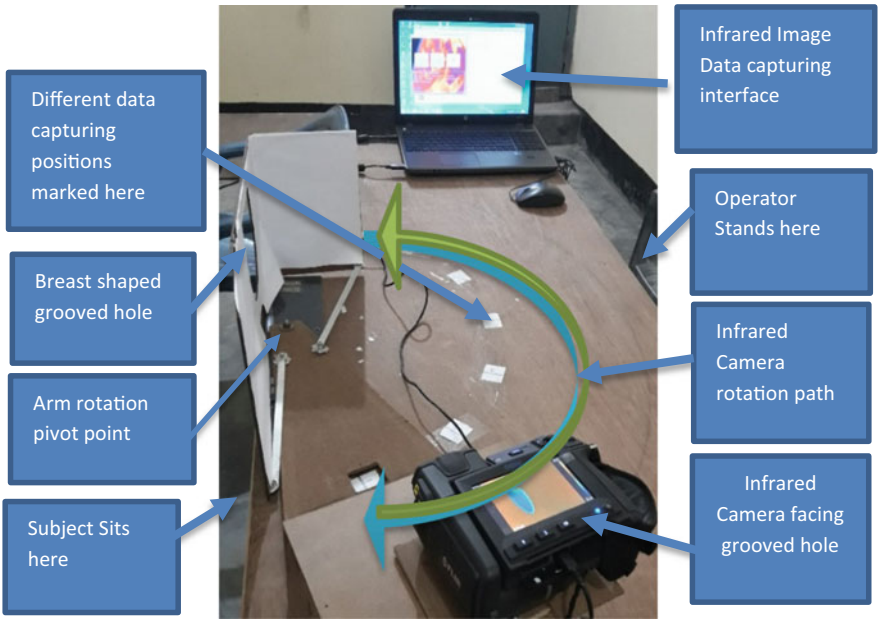


Fig. 3 Breast cancer screening setup for rotational thermography

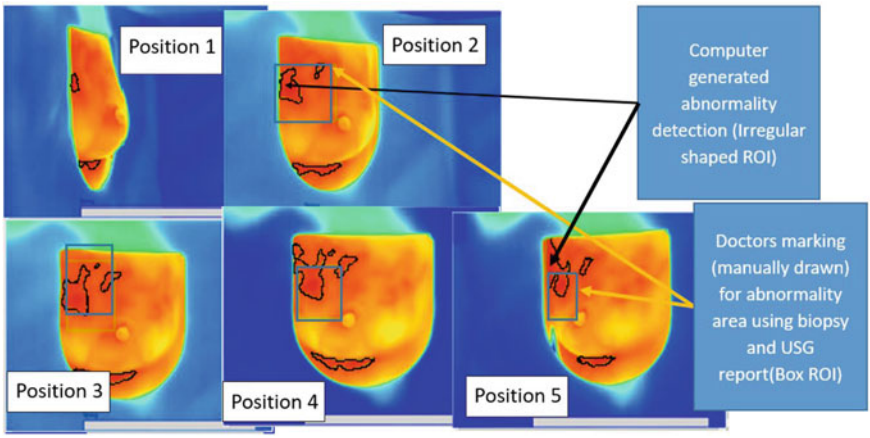


Fig. 4 ROI of detected abnormality on IR breast images captured from different angles

various stages of development to ensure optimal performance. Previous studies needed to integrate data collection protocols and analysis algorithms, resulting in suboptimal system performance. This study's data collection protocols were intelligently designed to integrate with IR image processing techniques.

In the FS pilot study, a novel data collection protocol was developed to overcome limitations in previous studies. The IR image processing techniques were integrated into the data analysis process, where clustering methods were used to segment the images and extract the region of interest (ROI). K-means clustering was applied to other image features using the mean temperature of each ROI as a discriminative feature. The clustering method was optimized based on experimental validation with consulting doctors, and seven clusters were finalized for analysis. Figure 4 shows the ROI of detected abnormality on IR breast images captured from different angles.

Regarding IR image feature-based analysis, PS1 and PS2 conventionally analyzed IR images by extracting features such as mean, median, mode, standard deviation, histogram, and maximum value. However, no reference was made to IR images acquired via other modalities.

In the PS3 pilot study, the primary reference for analysis was ultrasound (USG) and biopsy reports obtained through other modalities. Clustering based on infrared images was used for segmentation, and the average temperature of each ROI was used as a discriminator. The clustering method was gradually improved, and seven clusters were finalized based on experimental validation by consulting doctors.

In the FS pilot study, the analysis was accomplished after integrating the temperature-controlled enclosure into the system. The differences in IR images acquired at different ambient temperatures were used as the key discriminating feature for analysis. This novel analysis technique effectively detected temperature variations associated with breast cancer.

The methodologies for cancer detection using IR imaging incorporated specific data collection techniques tailored for each pilot study. Modifications and optimizations were made to address challenges and improve imaging results. Integrating data collection protocols with IR image processing techniques ensured comprehensive analysis at every stage of development.

By employing clustering methods, extracting key features, and leveraging temperature variations, these methodologies aimed to enhance the accuracy of breast cancer detection using IR imaging. Experimental validation and consultation with medical professionals were crucial in refining the techniques and achieving more reliable results.

Therefore, the methodologies for diabetes and cancer detection using IR imaging encompassed various steps, including imaging setup, data collection, IR image processing, feature extraction, and data analysis. These methodologies aimed to optimize the performance of the IR imaging system, contributing to accurate diabetic screening and early detection of breast cancer. Integrating machine learning algorithms and consultation with domain experts were critical in achieving more reliable and effective results in both research areas. Table 1 briefly details different aspects of the methodology used and compares infrared imaging-based diabetes detection and cancer detection.

3 Results and Discussion

The study highlights the potential of IR imaging-based diabetes screening, where the ear cavity zone takes a significant role and the potential of IR imaging-based screening systems in detecting breast cancer. The research shows that advances in data collection techniques, image analysis, and machine learning algorithms can improve accuracy and efficiency.

3.1 Diabetes Screening Using IR Imaging

The research studied 243 human subjects from different diabetic camps in India. The data was collected with SRM Medical College Hospital and Cachar Cancer Hospital. Ethical clearance and informed consent were obtained from all subjects. The study group comprised 129 males and 114 females aged between 19 and 91.

3.1.1 Feature Extraction and Classification

A statistical image analysis algorithm was developed for the "infrared image analysis module." The algorithm automatically plots the mean, standard deviation, median, mode, skewness, kurtosis, moments, root mean square (RMS), entropy, energy, and

Table 1 Comparison chart of methodology used for infrared imaging-based diabetes and breast cancer detection

Methodology	Diabetes detection	Cancer detection
Imaging setup and data collection	Data collection setup created at SRM Hospital and Cachar Cancer Hospital using IR camera to capture images of different body regions. LabVIEW software used for image analysis	Data collection techniques employed in four pilot studies (PS1, PS2, PS3, and FS) for breast cancer screening. Specific setups and modifications utilized for optimal data acquisition
Temperature control and image capture	Subjects acclimated to temperature-controlled room for 15 min. Skin temperature variations captured in the range of 0.05–0.1 °C	Various setups used in different pilot studies. FS implemented rotational thermography with temperature variations
Data collection plan	Two sessions: fasting blood sample collection and postprandial blood sample collection. Approximately 12–13 subjects per day	Varied data collection techniques in different level (PS1, PS2, PS3, and FS) of studies
Infrared image processing techniques	Manual, semi-automated, and full-automated ROI selection methods. Image processing algorithms (thresholding and morphological operations) used	Integration of data collection protocols and analysis algorithms. IR image processing techniques integrated at every stage of development. Clustering methods used for segmentation and ROI extraction
Analysis and information extraction techniques	Thresholding and shape-based object tracing technique employed. Temperature values stored for further analysis	IR image-based clustering used for segmentation. Mean temperature of ROI used as the discriminating feature
Feature extraction techniques	Generalized "Hough Transform" algorithm used for geometric shape features. Statistical image analysis algorithm recorded various features. Extraction of features like mean, median, mode, standard deviation, and histogram. Gradual improvement and finalization of clustering method	Feature extraction based on mean temperature and clustering methods. Seven discrimination clusters have been extracted
Machine learning classification	Statistical features used as input for machine learning algorithms. SVM algorithm demonstrated better classification accuracy	Analysis based on IR image features and discrimination using clustering, for machine learning algorithm (SVM) is used

(continued)

Table 1 (continued)

Methodology	Diabetes detection	Cancer detection
IR image feature-based analysis technique	Statistical features (mean, median, mode, standard deviation, and maximum value) extracted	Integration of temperature-controlled enclosure. Differences in IR images acquired at different ambient temperatures used as discriminating feature
Overall aim	Development of diabetic screening system using non-invasive, no touch method for improved medical diagnostics and early disease detection	Optimal system performance for breast cancer screening is achieved through an intelligently designed data acquisition protocol and integration with infrared imaging technology

maximum temperature values of a region of interest (ROI) on an infrared image as statistical value. These statistical values are considered as discriminators for classification using machine learning algorithms.

3.1.2 Gender-Specific Classification

The best three significant features for males were identified as standard deviation, mean, and RMS, while for females, they were different. As a result, separate machine learning algorithms were developed for classifying males and females. The experimental database determined the best features based on blood biochemical reports and infrared image output. It was observed that changes in the database might interchange the ranking position of features, but the basic algorithm remained the same.

3.1.3 Thermal Profile Distribution

The thermal profile distribution of diabetic and non-diabetic patients was compared in specific body regions, including the inner center points (eye), palm, ear cavity, and inner posterior buccal cavity (mouth). The temperature difference between diabetic and non-diabetic subjects was approximately 0.5 degrees centigrade. These findings supported previous research using thermography for diabetic patients, prompting further optimization of body regions for better results; here, the earlobe shows a significant differentiation between diabetic and non-diabetic subjects.

Table 2 Comparison of results from each study phase for diabetes screening

	Population-based fasting cross-sectional and case–control studies 243 subjects	Population-based PP cross-sectional and case–control studies 243 subjects
Sensitivity (%)	53.968	53.846
Specificity (%)	81.443	76.344
Positive predictive value (PPV)%	65.385	61.404
Negative predictive value (NPV)%	73.148	70.297
Accuracy (%)	70.625	67.089
FNR	46.032	46.154
FPR	18.557	23.656
LR+	2.908	2.276
LR−	0.565	0.605
Diagnostic odd ratio (DOR)	5.146	3.765

3.1.4 Classification Performance

A population-based cross-sectional and case–control study was conducted to evaluate the performance of the diabetes screening system. Sensitivity, specificity, positive predictive value (PPV), negative predictive value (NPV), accuracy, false negative rate (FNR), false positive rate (FPR), and diagnostic odds ratio (DOR) were calculated for each study group. The overall accuracy rate of diabetes screening with fasting data was 70.62% and that of diabetes screening with postprandial data was 67.08%. Table 2 shows the results of the diabetes screening study phase of a population-based cross-sectional and case–control study of fasting and PP.

3.2 Breast Cancer Detection Using IR Imaging

A novel non-contact and non-invasive breast imaging method was developed for breast cancer detection. This method captured all abnormalities presented by the subjects and introduced a rotational thermography setup for better visualization of the affected area.

3.2.1 Data Collection and Analysis

Data was collected from four pilot studies where PS1 was for 71 subjects, PS2 was done on 10 subjects, PS3 on 37 subjects, and the final study (FS) on 119 subjects. A

double-blind test was conducted during PS3 to FS. Clustering techniques were done using K-median, and a support vector machine (SVM) learning algorithm was used for feature extraction and classification.

3.2.2 Classification Performance

The accuracy of the developed system for screening breast abnormalities and detecting malignant tumors was 93.27% for the FS dataset. The sensitivity and specificity were 86.67% and 97.30%, respectively. These results demonstrated the effectiveness of the proposed IR image acquisition and analysis technique for breast cancer detection. Table 3 shows the results from study phases for breast cancer screening system for population-based cross-sectional and case–control studies PS3 and FS.

Therefore, the research presents promising results for diabetes screening and breast cancer detection using IR imaging. The advanced image processing and machine learning approaches offer accurate and efficient medical diagnosis, indicating the potential for clinical applications in the future. Further research and optimization of the proposed method could improve health care and early detection of diseases.

Table 3 Comparison of results between PS3 study phases and FS for breast cancer screening system

	Population-based PS3 cross-sectional and case–control studies 37 subjects	Population-based Final System FS cross-sectional and case–control studies 119 subjects
	Female	Female
Sensitivity (%)	90.00	86.67
Specificity (%)	82.61	97.30
Positive predictive value (PPV)%	69.23077	95.12
Negative predictive value (NPV)%	95	92.30
Accuracy (%)	84.84848	93.27
FNR	10	13.33
FPR	17.3913	2.70
LR+	5.175	32.06
LR−	0.121053	0.13
Diagnostic odd ratio (DOR)	42.75	234

3.3 Discussion

The comprehensive approach presented in this study provides valuable insights into the use of infrared imaging for diabetes screening and breast cancer detection. The findings contribute to developing a comprehensive framework for data acquisition and automated analysis methods and illustrate the importance of joint research in medical thermography.

4 Conclusion and Future Research Work

The results of this study support the use of infrared imaging as a promising tool for diabetes and breast cancer screening systems using the comprehensive framework. The integrated approach shows the potential of automatic image processing and machine learning techniques to improve the accuracy and efficiency of infrared image-based systems for medical diagnosis. Further studies are needed to validate and refine this comprehensive approach. The development of standardized protocols and more extensive clinical trials could help determine the effectiveness and reliability of infrared imaging for diabetes screening and breast cancer detection. Additionally, efforts should be made to address the challenges associated with quantifying malignancy temperature levels and train qualified healthcare professionals to use infrared imaging techniques.

Acknowledgements We thank Dr. Ashok Salhan from DIPAS, DRDO, and Dr. Subhrangsu Aditya from the J.U. School of Cognitive Sciences on medical and perceptual aspects of imaging. We also thank Dr. M. Anburajan, PhD in Biomedical Engineering, SRM University. Dr J. S. Kumar, Senior Dialectist, SRM Hospital, for their assistance in organizing a diabetes camp to collect IR data. Dr. R. Ravi Kannan and his team of Cachar Cancer Hospital were also recognized for assisting us with data collection and testing facilities. Smt Sayantani Banerjee and Smt Mamata Chakraborty were also recognized for their technical contributions. We sincerely acknowledge Sri Aditya Kumar Sinha, Director, C-DAC, Kolkata, for his kind support and encouragement in implementation of the research Project. The project was funded by Ministry of Electronics and Information Technology (MeitY), Govt. of India.

References

1. Sodi A, Giambene B, Miranda P, Falaschi G, Corvi A, Menchini U (2009) Ocular surface temperature in diabetic retinopathy: a pilot study by infrared thermography. Eur J Ophthalmol 19(6):1004–8
2. Roback K, Johansson M, Starkhammar A (2009) Feasibility of a thermographic method for early detection of foot disorders in diabetes. Diabetes Technol Ther 11(10):663–667
3. Nishide K, Nagase T, Oba M, Oe M, Ohashi Y, Iizaka S, Nakagami G, Kadowaki T, Sanada H (2009) Ultrasonographic and thermographic screening for latent inflammation in diabetic foot callus. Diabetes Res Clin Pract 85(3):304–309 Epub 2009 Jul 7

4. Ring EFJ, Ammer K (2000) The infrared imaging technique in medicine. Thermol Int 10(1):7–14
5. Fujiwara Y, Inukai T, Aso Y, Takemura Y (2000) Thermographic measurement of skin temperature recovery time of extremities in patients with type 2 diabetes mellitus. Exp Clin Endocrinol Diabetes 108(7):463–469
6. Fushimi H, Inoue T, Yamada Y, Matsuyama Y, Kubo M, Kameyama M (1996) Abnormal vasoreaction of peripheral arteries to cold stimulus of both hands in diabetics. Diabetes Res Clin Pract 32(1–2):55–59
7. Sivanandam S, Anburajan M, Venkatraman B, Menaka M, Sharath D (2012) Medical thermography: a diagnostic approach for type 2 diabetes based on non-contact infrared thermal imaging. Endocrine. https://doi.org/10.1007/s12020-012-9645-8
8. EtehadTavakol M, Sadri S, Ng EYK (2010) Application of K- and fuzzy c-means for color segmentation of thermal infrared breast images. J Med Syst 34(1):35–42. https://doi.org/10.1007/s10916-008-9213-1
9. Etehadtavakol M, Ng EYK (2017) Color segmentation of breast thermograms: a comparative study, pp 69–77. https://doi.org/10.1007/978-981-10-3147-2_6
10. Garduño-Ramón M, Vega-Mancilla S, Morales-Henández L, Osornio-Rios R (2017) Supportive noninvasive tool for the diagnosis of breast cancer using a thermographic camera as sensor. Sensors 17(3):497. https://doi.org/10.3390/s17030497
11. Prakash RM, Bhuvaneshwari K, Divya M, Sri KJ, Begum AS (2017) Segmentation of thermal infrared breast images using K-means, FCM and EM algorithms for breast cancer detection. In: 2017 International conference on innovations in information, embedded and communication systems (ICIIECS). IEEE, Mar 2017, pp 1–4. https://doi.org/10.1109/ICIIECS.2017.8276142
12. Venkataramani K, Mestha LK, Ramachandra L, Prasad SS, Kumar V, Raja PJ (2015) Semi-automated breast cancer tumor detection with thermographic video imaging. In: 2015 37th annual international conference of the IEEE engineering in medicine and biology society (EMBC). IEEE, Aug 2015, pp 2022–2025. https://doi.org/10.1109/EMBC.2015.7318783
13. Pramanik S, Banik D, Bhattacharjee D, Nasipuri M, Bhowmik MK, Majumdar G (2019) Suspicious-region segmentation from breast thermogram using DLPE-based level set method. IEEE Trans Med Imaging 38(2):572–584. https://doi.org/10.1109/TMI.2018.2867620
14. Wu M-N, Lin C-C, Chang C-C (2007) Brain tumor detection using color-based K-means clustering segmentation. In: Third international conference on intelligent information hiding and multimedia signal processing (IIH-MSP 2007). IEEE, Nov 2007, pp 245–250. https://doi.org/10.1109/IIHMSP.2007.4457697
15. Kandlikar SG et al (2017) Infrared imaging technology for breast cancer detection—current status, protocols and new directions. Int J Heat Mass Transf 108:2303–2320. https://doi.org/10.1016/j.ijheatmasstransfer.2017.01.086
16. Gonzalez RC, Woods RE (2007) Digital image processing, 4th ed. ISBN 81-7758-168-6
17. EtehadTavakol M, Chandran V, Ng EYK, Kafieh R (2013) Breast cancer detection from thermal images using bispectral invariant features. Int J Therm Sci 69:21–36. https://doi.org/10.1016/j.ijthermalsci.2013.03.001

Precision Agriculture Through Stress Monitoring in Crops with Multispectral Remote Sensing Data



Konumuri Kalyan Suhas, G. Kalyani, and Mandava Venkata Sri Sai Surya

Abstract Although agriculture is regarded as the foundation of the Indian economy, there are times when plants are unable to develop to their full potential owing to a variety of factors, including pests, starvation, and other factors. In these situations, several types of fertilizers are utilized. However, a small-scale farmer is unable to apply fertilizer to hundreds of acres; thus, in these situations, our research is quite helpful for locating the areas where the plant growth is irregular. In order to increase yields, efficiency, and profitability for farmers, precision agriculture can assist farmers in monitoring crop health, soil nutrient levels, and water use, among other factors. The vegetation indices of the crop are calculated using satellite data by the crop stress monitoring system, which is essential for monitoring the developing field. The data was obtained using satellite images with varying levels of geographic and temporal resolution.

Keywords Crop stress · Land classification · Normalized Difference Vegetation Index (NDVI) · Normalized Difference Water Index (NDWI) · Soil-Adjusted Vegetation Index (SAVI) · Enhanced Vegetation Index (EVI)

1 Introduction

Using satellite data to assess crop stress is essential for efficient agricultural management. This technique provides early identification of stress factors including water scarcity, nutritional deficits, pests, illnesses, and extreme weather events by utilizing satellite images. Its extensive coverage enables large-scale agricultural monitoring, making it easier to see patterns and trends [1]. With the use of this data, precision resource management may be used to distribute water, fertilizer, and pesticides in the

K. K. Suhas (✉) · G. Kalyani · M. V. S. S. Surya
Department of Information Technology, Velagapudi Ramakrishna Siddhartha Engineering College, Vijayawada, India
e-mail: kalyan.konumuri@gmail.com

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2023

S. Shakya et al. (eds.), *Fourth International Conference on Image Processing and Capsule Networks*, Lecture Notes in Networks and Systems 798,
https://doi.org/10.1007/978-981-99-7093-3_28

best possible way. As a result of quick decision-making made possible by real-time data, farmers are better equipped to reduce stress and increase crop output.

Using satellite data to detect crop stress has several advantages for agriculture [2]. Early detection of crop stress permits prompt mitigation of the effects of stressors such nutrient deficits, water stress, pests, and diseases. Satellites' extensive coverage enables monitoring of crops across huge areas, making it easier to see patterns and trends [3]. By enabling for exact input application, minimizing waste, and mitigating environmental effects, this information aids in the optimization of resource management. Real-time data enables prompt decision-making, allowing farmers to improve output by adjusting practices.

A useful tool for agricultural stress monitoring systems, Google Earth Engine (GEE), enables users to view and examine a sizable library of satellite images [4]. With the help of GEE, scientists and farmers may preprocess the images, use vegetation indices, and include auxiliary data to evaluate crop health and spot stressed regions. Large-scale datasets may be effectively analyzed using GEE's robust computer infrastructure, allowing for the quick identification of geographical patterns and trends. GEE's user-friendly interface and code editor also make it easier to construct specialized tools and apps for crop stress monitoring, giving farmers the knowledge, and they need to manage resources more effectively and make wise decisions. GEE offers a comprehensive and scalable platform for analyzing satellite images and enhancing agricultural practices, which boosts crop stress monitoring overall.

An integral part of Google Earth Engine (GEE) that enables users to engage with its extensive collection of satellite data and sophisticated geospatial processing capabilities is its JavaScript Web interface [5]. Users may see satellite images, employ analytic methods, and get generated data products using the online interface. Users may explore and analyze geographic data in an easy and simple way by using interactive maps, charts, and visualizations made with JavaScript modules and frameworks [6]. Users are able to customize their analyses using the online interface, search for certain areas of interest, and learn more about various elements of crop stress monitoring, resource management, and decision-making.

The main objective of this project is to identify the stress of the crop with the help of satellite data. Likewise the various objectives of this project are to monitor the crop growth and development over time and assess the overall health of the crop, to estimate yield potential and assess the impact of environmental factors such as temperature, rainfall, and soil moisture, to support precision agriculture by providing detailed information on crop growth and development to inform decision-making, such as fertilizer application or irrigation management.

2 Review of Literature

The study presented an [7] innovative approach for calibrating and validating models for estimating Kc, height, and LAI in cotton based on Sentinel-1 and Sentinel-2 imagery to support irrigation management. The study [8] adopts a popular strategy of using long-lasting burned signatures to map burned areas; however, very few studies have attempted to use complimentary multispectral indices to detect burned areas.

The study of [9] mainly focuses on distribution of plant biophysical and biochemical parameters, spatial structure and variability in the UAV and Sentinel-2 imagery and comparison of UAV and Sentinel-2 data along transect lines. In the article [10], WRASIS was applied at three agriculture project and defining crop patterns, adding crops spectral signatures into software libraries and identify crop types to produce different outputs of crop maps.

The use of satellite data to control crop stress is an area of ongoing study, and several studies have been done on it. In order to manage crop stress, remote sensing techniques have been employed to identify crop stress brought on by water, nutrient, and pest-related issues. Based on satellite data, a crop water stress index (CWSI) has been created to control water stress in crops. This has improved irrigation, boosted agricultural yields, and improved water usage efficiency. Crop stress has been managed using precision agricultural techniques, such as site-specific fertilizer management and variable rate irrigation, which have improved crop growth and decreased stress. Additionally, it has been discovered that employing satellite data for agricultural disease detection and management can lower the need for pesticides and increase crop yields. Finally, crop stress prediction algorithms have been created to accurately forecast crop stress, enabling farmers to take preventative action to minimize stress. Even though using satellite data to control crop stress has shown to be quite beneficial, additional study is needed to provide more precise and reliable methods for identifying and treating crop stress.

3 Methodology

Architecture Diagram

The different lessons that may be drawn from Fig. 1 include: Launch Google Earth and construct a polygon. Think about the options for crop land and non-crop land. In order to improve the image quality, cloud mask must be removed. Following that, training of both the cropland and non-cropland takes place. Then, the land is divided into crop land and non-crop land using the SMILE-CART algorithm. Next, take into account the crop land from the categorized land. Then, figure out the different indices, such as NDVI, NDWI, SAVI, and EVI. Visualize land classification and NDVI, NDWI, SAVI, and EVI, and you are done.

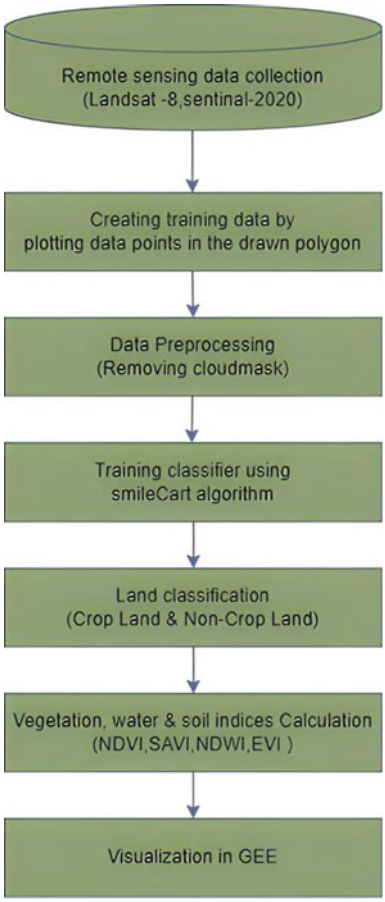


Fig. 1 Architecture diagram

Description of Algorithms

SMILE-CART Algorithm

A decision tree-based machine learning algorithm known as the SMILE (Statistical Machine Intelligence and Learning Engine)-CART (Classification and Regression Tree) method may be applied to classification and regression issues. Recursively dividing the data into subsets according to the input characteristics, the method produces a tree-like structure that may be utilized for prediction.

The optimal feature and split that maximizes the difference in the goal variable between the subsets are chosen to build the decision tree model. The SMILE-CART method is renowned for its readability, interpretability, and simplicity. It can handle missing values and variables that are categorical or continuous. The most crucial characteristics are chosen first by the tree, and hence, the method may be utilized for feature selection.

To prevent over-fitting and enhance the model's generalization capabilities, the tree can also be trimmed. The user must supply a set of labeled training data that comprises the input features and the target variable in order to apply the SMILE-CART method. The user must additionally define the minimum number of samples needed for a split, the maximum depth of the tree, and other parameters. Once trained, the model may be used to generate predictions based on fresh data. The SMILE-CART method is often utilized in several fields, including remote sensing, ecology, business, and medicine.

Algorithm ID3 Input:

- Examples: Instances or observations in the dataset that are used for training the decision tree.
- Target attribute: The class or category that is required to predict or classify for new instances.
- Attributes: The features or characteristics of the instances that are used to make decisions and build the decision tree.

Output: A decision tree that represents learned classification rules based on the given inputs.

Method:

- For the tree, create a root node.
- If every example is positive, return the single-node tree root with the label $= +$.
- If every example is negative, return the single-node tree root with the label $= -$.
- If attributes is empty, return the single-node tree root, with label equal to the most prevalent TargetAttribute value in examples.
- Otherwise Start.
  - A $\leftarrow$ the characteristic that best categorizes examples from the group of attributes
  - The root's decision-making attribute $\leftarrow$ A
  - For every possible value, vi, of A,
  - Below Root, add a new tree branch that corresponds to the test A $=$ vi
  - Let examples, $_{\text{vi}}$ represent the subset of Examples where vi is the value for A
  - If examples, $_{\text{vi}}$ is not filled out

- Then, add a leaf node with label $=$ most frequent TargetAttribute value in Examples underneath this new branch.
- Else include the subtree below this new branch.

ID3(Examples$_{\text{vi}}$, TargetAttribute, Attributes $-$ {A})

- End
- Return Root

4 Experimental Results

Data Collection

The Google Earth Engine and JavaScript were used for all computational and programmable operations. The LandSat-8 satellite is utilized to gather the data for this research. Cropland and non-cropland land from Movva make up the area under consideration. The cloud mask is eliminated using the .filterMetadata() method.

Experimental Metrics

Normalized Difference Vegetation Index (NDVI)

It is used to assess the strength and vitality of plants. The range of NDVI values is -1 to $+$ 1, with higher values indicating more robust and thick vegetation. The numbers 0 and negative denote the absence of vegetation and the presence of water, respectively.

The formula for calculating NDVI is:

$$\text{NDVI} = \frac{(\text{NIR} - \text{Red})}{(\text{NIR} + \text{Red})}$$

where NIR is the reflectance value in the near-infrared band and Red is the reflectance value in the red band.

Soil-Adjusted Vegetation Index (SAVI)

It pinpoints how soil brightness affects vegetation's spectrum response. Higher numbers denote healthier and denser vegetation. SAVI values vary from -1 to $+$ 1. Compared to other indicators, it can offer a more accurate assessment of the health of the plants, especially in regions with high soil moisture or brightness.

The formula for calculating SAVI is:

$$\text{SAVI} = \frac{(\text{NIR} - \text{Red})}{(\text{NIR} + \text{REd})} * (1 + L)$$

where NIR is the reflectance value in the near-infrared band, red is the reflectance value in the red band, and L is the soil adjustment factor, which ranges from 0 to 1.

Normalized Difference Water Index (NDWI)

Evaluates the water content of soil and plants. Higher values signify increased water content. NDWI values range from -1 to $+$ 1. Negative numbers represent soil or plants, whereas a value of 0 implies there is no water present. NDWI is particularly helpful for identifying changes in soil moisture content in agricultural regions as well as for monitoring water bodies including lakes, rivers, and wetlands.

The formula for calculating NDWI is:

$$\text{NDWI} = \frac{(\text{Green} - \text{NIR})}{(\text{Green} + \text{NIR})}$$

where green is the reflectance value in the green band and NIR is the reflectance value in the near-infrared band.

Enhanced Vegetation Index (EVI)

In locations with thick vegetation and high soil background reflectance, it is intended to increase the sensitivity of vegetation indices. Higher values imply denser and better plant cover. The EVI values range from −1 to 1. The EVI is frequently used to track changes in land usage over time, drought conditions, and plant growth.

The formula for calculating EVI is:

$$\text{EVI} = 2.5 * \frac{(\text{NIR} - \text{Red})}{(\text{NIR} + C_1 * \text{Red} - C_2 * \text{Blue} + L)}$$

where NIR is the reflectance value in the near-infrared band, Red is the reflectance value in the red band, Blue is the reflectance value in the blue band, and C_1, C_2, and L are empirical coefficients that depend on the sensor and atmospheric conditions.

Discussion on results

Figure 2 indicates the crop land (indicated in green color) and non-crop land (indicated in pale pink color) classified with help of SMILE-CART algorithm.

Figure 3 indicates the percentage of NDVI values, (i.e.) green color indicates high NDVI values (0.6–0.9), white color indicates average NDVI values (0.2–0.5), and blue color indicates the least NDVI values (≤ 0.1).

Figure 4 indicates the percentage of SAVI values, green color indicates high SAVI values (0.3–0.9), white color indicates average SAVI values (0.1–0.3), and blue color indicates the least SAVI values (≤ 0.1).



Fig. 2 Classification of land into crop and non-crop lands

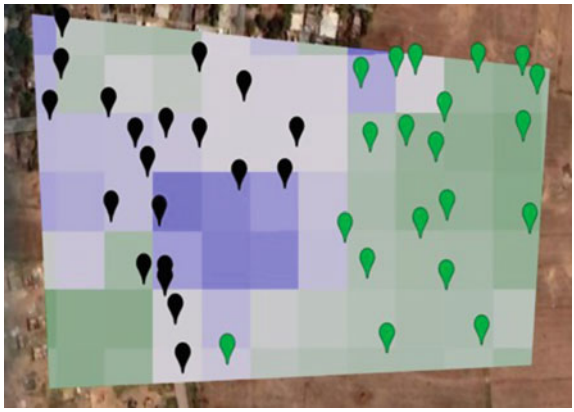


Fig. 3 NDVI values in the selected region

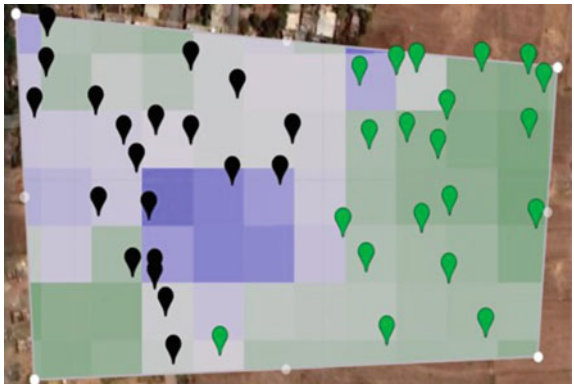


Fig. 4 SAVI values in the selected region

Figure 5 indicates the percentage of NDWI values, green color indicates low NDWI values (≤ -0.3), blue color indicates high NDWI values (0.2–1), and the white color indicates average NDWI values (-0.3 to 0.2).

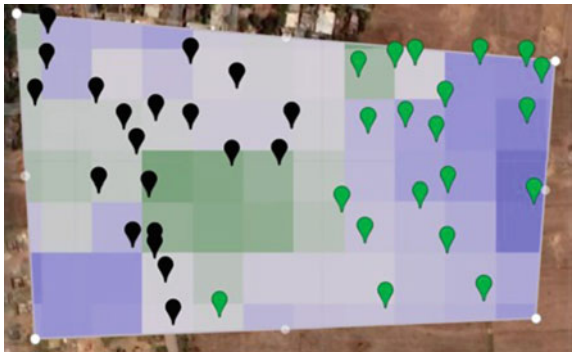


Fig. 5 NDWI values in the selected region

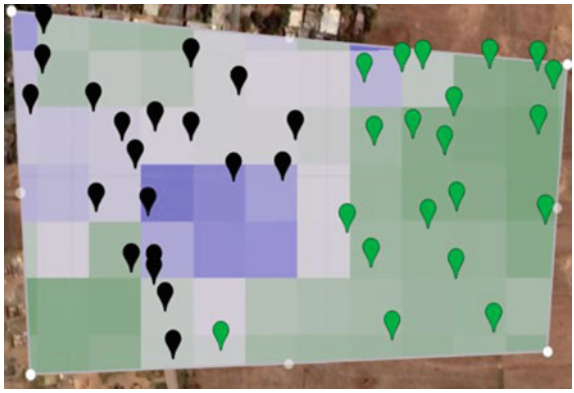


Fig. 6 EVI values in the selected region

Table 1 Index values

Index\values	High	Average	Low
NDVI	0.62135	0.14374	− 0.10977
SAVI	0.53169	0.19965	− 0.41049
EVI	0.59323	0.14924	0.00317
NDWI	0.33997	0.14841	− 0.35734

Figure 6 indicates the percentage of EVI values, green color indicates high EVI values (0.2–0.8), white color indicates average EVI values (0.0–0.2), and blue color indicates the least EVI values (≤ 0.0).

If a sector of agricultural land is considered, the usage of several vegetation indices such as NDVI, NDWI, SAVI, and EVI in order to determine the crop's stress in that specific area. Below is an illustration of one such area.

Table 1 then displays a variety of value ranges for various indices.

5 Conclusion

In this project, Google Earth Engine is used to track the crop's stress using data from satellites. In order to classify the land into two categories, firstly launch the Google Earth Engine and draw a polygon. Then, feature selections such as crop land and non-crop land must be considered, upon which training of both of the lands takes place. Finally, the SMILE-CART algorithm is used to calculate the values of the indexes and monitor the stress of the crop.

After determining the values of the various indexes in this project, the area must be specifically addressed, identify the issue, and then offer a solution. These solutions include different kinds of pesticides or insecticides that should be used on that specific plant, so that the loss will not be significant.

References

1. Sanae H, Jilbab A, Sanad IM (2020) Crop stress monitoring system using satellite data and machine learning techniques. In: 2020 IEEE Jordan international joint conference on electrical engineering and information technology (JEEIT), Amman, Jordan
2. Ahmed S, Yang Z, Zhang J (2020) Satellite-based monitoring of crop water stress using convolutional neural networks. In: 2020 IEEE international geoscience and remote sensing symposium (IGARSS), Waikoloa, HI, USA
3. Zhang P, Li H, Li C (2020) Satellite-based crop stress monitoring system using deep learning and hyperspectral imagery. In: 2020 IEEE international conference on artificial intelligence and computer applications (ICAICA), Wuhan, China
4. Sharma R, Singh R, Tiwari S (2021) Satellite-based crop stress monitoring using convolutional neural networks. In: 2021 IEEE region 10 symposium (TENSYMP), Dhaka, Bangladesh
5. Shrestha S, Zhang T, Li H (2021) Satellite-based crop water stress monitoring using deep learning and multi-temporal Sentinel-2 imagery. In: 2021 IEEE international conference on artificial intelligence and computer applications (ICAICA), Wuhan, China
6. Wang Y, Li H, Zhang L (2022) Crop water stress monitoring using satellite data and long short-term memory networks. In: 2022 IEEE international conference on artificial intelligence and computer applications (ICAICA), Wuhan, China
7. Kaplan G, Fine L, Lukyanov V, Malachy N, Tanny J, Rozenstein O (2023) Using Sentinel-1 and Sentinel-2 imagery for estimating cotton crop coefficient, height, and leaf area index. Agric Water Manage 276
8. Deshpande MV, Pillai D, Jain M (2022) Agricultural burned area detection using an integrated approach utilizing multi spectral instrument based fire and vegetation indices from Sentinel-2 satellite. MethodsX 9
9. Li M, Shamshiri RR, Weltzien C, Schirrmann M (2022) Crop monitoring using Sentinel-2 and UAV multispectral imagery: a comparison case study in Northeastern Germany. Rem Sens 14(17)
10. Zahran SAEl-S, Saeed RA-H, Elazizy IM (2022) Remote sensing based water resources and agriculture spatial indicators system. Egyptian J Rem Sens Space Sci 25(2)

EEG Signal Feature Extraction Using Principal Component Analysis and Power Spectral Entropy for Multiclass Emotion Prediction



S. Babeetha and S. S. Sridhar

Abstract Emotion recognition in artificial intelligence is a challenging factor. Recent years have seen a rise in interest in emotion analysis based on EEG signals due to many factors, including the availability of huge EEG datasets, developments in Brain-Computer interface (BCI) devices, and discoveries in machine learning. Due to its high temporal precision, portability, and relatively low cost of data collection, EEG is a viable option for researching the brain correlates of numerous cognitive activities, including emotion. The user's capacity to generate these signals is indicative of how well their brain processes information across a variety of cognitive, emotional, and physical domains. Important feature values must be extracted from these raw signals, which is still an essential part of the deployment process. In order to better understand the brain's activity, this study explains how to use principal component analysis (PCA) and power spectral entropy (PSE) to extract characteristics from an EEG signal. In this analysis, we evaluate how well-known classification algorithms fare when used on the recovered features: the support vector machine (SVM), the K-nearest neighbors (KNN), and the Naive Bayes (NB) classifier. All of the simulations were carried out in MATLAB using a HEADIT dataset. The results reveal that the suggested classifiers outperform the standard method in predicting emotional states.

Keywords EEG signals · Emotion prediction · Principal component analysis · Power spectral entropy · Support vector machine · K-nearest neighbors · Naive Bayes classifier

S. Babeetha (✉) · S. S. Sridhar
Department of Computing Technologies, College of Engineering and Technology, SRM Institute of Science and Technology, Kattankulathur Campus, Chengalpattu, Tamil Nadu, India
e-mail: bs5393@srmist.edu.in

S. S. Sridhar
e-mail: sridhars@srmist.edu.in

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2023

S. Shakya et al. (eds.), *Fourth International Conference on Image Processing and Capsule Networks*, Lecture Notes in Networks and Systems 798,
https://doi.org/10.1007/978-981-99-7093-3_29

1 Introduction

Electroencephalography (EEG) is a non-invasive technology that specialists often use in a wide variety of contexts due to its composite time series data. Emotional computing encompasses the study of how computers interpret and recognize human emotions; this is known as emotion recognition. Humans rely on their emotions to help them with a wide range of cognitive processes, including thinking, planning, decision-making, and more. Individuals who were emotionally stable and had a healthy perspective on life were more likely to achieve their goals. Moreover, negative emotions might hinder one's decision-making skills and even one's physical health. Some of the non-fun uses of emotion analysis include in areas such as adaptive advertising, an emotion-based music player, and online education. The EEG cap's popularity among clinical analysts may stem from the fact that it is versatile piece of equipment that may be put to work in a variety of contexts requiring emotion recognition. Speech, body language, facial expressions, physiological responses, etc., may all be used for emotion identification. These approaches rely on surface expressions of emotion and are thus inadequate for gauging internal states of mind. This previously hidden information is now available because to EEG-driven signals, which provide a wide range of emotional patterns. As a result, several feature extraction-based methods for analyzing EEG signals have been developed. Nonlinearity and underlying difficulties in the EEG signals may be measured using these properties. Recent advances in machine learning have had far-reaching effects on signal and information processing by making automated feature extraction and selection possible. In light of this, we use principal component analysis and power spectral entropy to forecast across many emotion classes in this article.

The following is a list of this work's most significant contributions:

- Improvement was done on all areas by addressing the problems discovered in earlier research.
- By developing and presenting a machine learning-based predictive modeling emotion recognition system, we have created a system that comprehends nonverbal data such as users' emotions, objectives, and attitudes.
- Individuals' feelings were analyzed using a principal component analysis (PCA) with a partial sums Eigenvector (PSE) and a comparison was made to the support vector machine, the K-nearest neighbors classifier, and the Naive Bayes classifier.

The remaining sections of the paper are laid out as follows. In Sect. 3, we detail the research methodological structure and procedure flow. Section 4 explores more into the findings and discussion. Section 5 gave the last ideas and recommendations for improvement.

2 Related Works

Researchers Misulis et al. [8], who studied the link between EEG and seizure drugs, found that fluctuations in the strength of the EEG signal revealed brain electrical activity. More entropy, or disorder, was seen in the EEG signals of patients with PTSD. More entropy was produced in the frontal lobe of the brain, the region in charge of experiencing, and expressing emotions. To categorize sleep phases in autonomous cars using individual EEG data, Michielli et al. [9] proposed a spiraled long short-term memory (LSTM) back-propagation neural network. Traditional machine learning-based emotion recognition algorithms may benefit from being trained on emotional brainwave data (EEG). Using DEAP's integrated monitored neural classification algorithm, we observed that our average HALA, HVLV, recognition, dominance, and likability scores were 97.39, 97.41, 98.21, and 97.68. Researchers discovered that the model had a 93.74% average success rate in classifying happy and sad faces after testing it on the SEED dataset. They found that the accuracy of the model was enhanced by associating mental states with emotions already present in EEG datasets. Qazi et al. [10] looked at the viability of developing an emotion detection Qazi system employing two-light deep learning in tandem with a CNN model ensemble. An emotion identification system is provided by combining the Mel-frequency cepstral coefficient (MFCC) with bio-signals. Studies of mental stimulation have shown that anxiety, calm, melancholy, and happiness are distinct states of mind. Multilayer perceptrons (MLP) are used as the classifiers. The use of EEG data to assess mental states has been shown in experiments to achieve an accuracy of up to 90%. This indicates that typical emotions may be recognized using scalp EEG data and the MFCC-MLP detection method.

Korotkova et al. [11] measured hippocampal theta alterations to examine their effects on cognitive, behavioral, and personality-based performance. Strong brain rhythms between 5 and 10 Hz, known as theta resonance, have been connected to a wide range of human actions. This article provides a concise overview of the research that has related theta frequency to different mental processes. The electroencephalogram (EEG) of people with mental illness should be studied when they are in a deep slumber, as proposed by Newson and Thiagarajan [12]. In order to better understand the diagnostic and therapeutic implications of PTSD's electrical impulse patterns, this research used EEG. To classify EEG data for moods, Ramzan and Dawn [13] investigated the use of machine learning. The purpose of this study is to determine whether or not certain EEG patterns are linked to the presence and severity of PTSD. The most convincing evidence of time-domain in EEG linked to event-related potentials is an exaggerated forebrain reaction time for task redirection (P3a intensity) and a reduced forebrain response to event-related potentials (P3b amplitude). These findings demonstrate that some individuals with PTSD suffer from attention difficulties that make it challenging for them to focus on common activities and remain dedicated even when faced with potentially harmful distractions. Using measures of emotional reactivity and brain regions associated with locus of control, Fitzgerald et al. [14] show that PTSD symptoms may be predicted in war veterans. It is difficult

to pull off gradients that abruptly disappear or burst. When used to Suhaimi classification and prediction tasks, the MFCC and CNN algorithms utilized by Suhaimi et al. [15] were shown to be imprecise and unreliable. Liu et al. [19] detail the use of a sparse encoder and a DNN with a heap database to create convolutional neural networks (CNNs). Overfitting, exaggerated gradients, and a lack of diversity among classes are the primary problems. Using ResNet50, the SEED dataset, DREAMER, and AMIGOS, Topic and Russo [16] highlight the benefits and drawbacks of convolutional neural networks (CNNs). Although the findings of Wang and Wang's [17] ensemble are more difficult to comprehend and anticipate, they were constructed utilizing SVM, KNN, naive Bayesian, and random forest methods. This strategy is helpful since it improves both the accuracy and performance of forecasts compared to using a single model. In their discussion of CNN with reference to the SEED IV dataset, Pereira et al. [18] note the need of taking into consideration co-occurring symptoms and disorders.

The SVM approach was used to attain an even greater overall accuracy of 92% in the research by George et al. [1]. A box-and-whisker plot and the DCT technique were used to identify the characteristics. By evaluating the statistical features of the Fast Fourier Transform (FFT), researchers found a 92% increase in accuracy for emotion recognition in the DEAP dataset (32 participants). This strategy is preferable to the one proposed by Seeja et al. since it has been shown to be more accurate. Differences in feature extraction and data preprocessing can account for the observed performance gaps. In Alhagry et al. [2] published a research that highlighted the significance of HCI-based emotion detection systems. They found problems on all three fronts, unlike the vast majority of research in this area, which only examine two (arousal and valence). High accuracy of 85.65% was attained for the valence category, 85.45% for the arousal category, and 87.99% for the liking category when the features extracted from the DEAP dataset were classified using LSTM-RNN. These techniques disregarded feature extraction in favor of end-to-end approaches since deep learning algorithms may extract and categorize features in a single step. Yin et al. [3] used graph convolutional neural networks (GCNN) to create an EEG-based emotion recognition model. The experiment used the DEAP data collection. ECLGCNN was applied to the data after it was partitioned, and the differential entropy features were retrieved. The ECLGCNN approach integrates GCNN with LSTM. Valence labeling accuracy was shown to be 85.04% across independent trials, 90.45% across subject-dependent trials, and 90.60% across all trials. The processing cost of this approach may be reduced by discovering novel methods to extract characteristics. To categorize emotions using the CNN model, which has shown to be capable of successful feature extraction, Cimtay et al. [4] used the end-to-end approach. This research enhanced the CNN model by adding extra layers to enhance classification precision. This research made use of the DEAP, LUMED, and SEED datasets. The model performed with an accuracy of 86.56% on the SEED dataset, 78.3% on the DEAP dataset, and 81.8% on the LUMED dataset. Iyer et al. [20] developed the convolution neural network (CNN) and long short-term memory (LSTM) based hybrid model which is developed for accurate emotion detection.

An emotion detection model capable of distinguishing between happy, sad, and neutral expressions was created by Wei et al. [5]. Four characteristics across five frequency bands were used in the SRU models that we constructed using a SEED dataset. Several causes led to the formulation of the concept of SRU. Long-term, it may be the answer to the RNN dependency issue. Using dual-tree compound wave transfer (DT-CWT), we were able to accurately extract time–frequency and nonlinear properties. This strategy is based on the iterative discovery process. The proliferation of EEG data channels has been the subject of recent study in the area of emotion identification, prompting Chao et al. [6] to look into the issue. They developed a novel strategy to solve this issue by combining deep belief-conditional random fields with glia chains. Three different datasets—AMIGOS, SEED, and DEAP—were used to test the model. Overall, the accuracy of these techniques was 76.13%. Using a DEAP dataset and two feature extraction methods—the variational mode decomposition (VMD) and the empirical mode decomposition (EMD)—Pandey et al. [7] examined emotional reactions to stimuli using EEG signals. DNN was also used for emotion categorization. The overall accuracy and valence accuracy for this approach were both 63% and 62%, respectively. According to the results, the deep neural network classifier outperformed the SVM classifiers when used to the emotional classification model. The study's authors claim that using VMD-based features rather than EMD-based techniques improved signal comprehension and performance. Although masking techniques have reduced the amplitude rate between the mono-components, there is room for improvement in EMD's frequency resolution.

According to the analysis, the following are some of the difficulties encountered by methods used to recognize human emotions.

(i) In contrast, the trial-and-error method of choosing network characteristics such as training parameters and the total number of nodes presents the most difficulty.
(ii) When comparing the classification accuracy of the deep learning approach to that of other current methods, it becomes clear that the former is much superior. To improve classification accuracy, using attention processes on different brain areas may help. However, training requires a lot more time. Therefore, a strategy to counteract these drawbacks is required.

3 Proposed Work

Human emotion detection through the suggested technique is a two-stage process that begins with feature extraction and ends with emotion identification. Features such as spectral flatness, logarithmic band power, fluctuation index, spectral decline, spectral roll-off, and relative energy are extracted from the input EEG data using principal component analysis and principal component spectral estimation. Following feature extraction, we utilize KNN, SVM, and NB classifiers to label facial expressions as proud, elated, joyful, satisfied, relieving, hopeful, curious, surprised, sad, embarrassed, ashamed, guilty, envious, disgusted, contemptuous, or furious.

3.1 Data Source

The dataset was obtained from https://headit.ucsd.edu/studies/3316f70e-35ff-11e3-a2a9-0050563f2612. It includes of audio recordings that imply an emotion and prompt participants to visualize a triggering situation or think back on a time when they felt that emotion.

3.2 Feature Extraction

PCA and PSE are two of the methodologies used for feature extraction (Fig. 1).

1. **PCA**

The process of decomposing attached variables into their individual metrics is an integral part of the PCA cycle. Principal component analysis is used to lower the dimensionality of the fused feature (FF) vectors, making them easier to classify accurately. The model is constructed taking into account the features that effectively classify the data. There are n separate dimensions in FF. In the first $F = z_1, z, \ldots, z_n$ has several dimensions, and we must reduce it to k_n. The PCA follows these stages to generate a set of reduced fused features (RFFs):

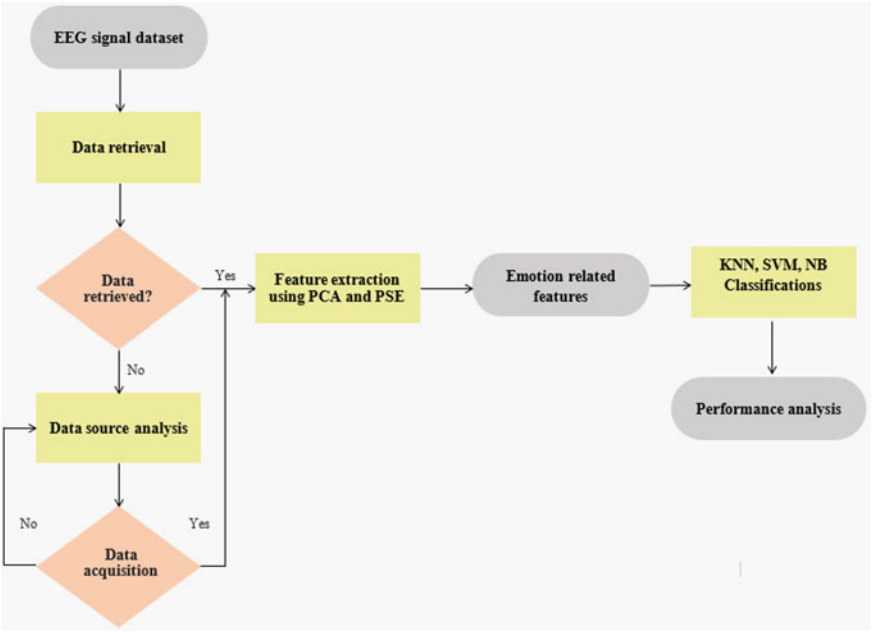


Fig. 1 EEG signal extraction using PCA and PSE

1. Data scaling

$$z_j^i = \frac{z_j^i - \overline{z_j}}{\sigma_j} \tag{1}$$

2. Co-variance matrix computation

$$\sum = \frac{1}{m} \sum_i^m (z_i)(z_i)^T, \sum \in R^{n \times n} \tag{2}$$

3. Third, we calculate the Eigenvector and Eigenvalue

$$w^T \sum = \lambda \mu U = \begin{bmatrix} | & | & & | \\ w_1 & w_2 & \ldots & w_n \\ | & | & & | \end{bmatrix}, w_i \in R^n \tag{3}$$

4. Eigenvalue selection. We used the following formula to choose the top 100 Eigenvalues in K-space for use in our classification system.

$$z_i^{\text{new}} = \begin{bmatrix} w_1^T z^i \\ w_2^T z^i \\ \cdots \\ \ldots.. \\ w_k^T z^i \end{bmatrix} \in R^k \tag{4}$$

5. At last, the feature matrix was constructed.

2. **PSE**

The spectrum complexity of an uncertain system may be quantified using power spectral entropy, which is a kind of information entropy. Let us pretend that the values of states in an uncertain system are determined by a random variable X, with possible outcomes including,

$$Y = \{y_1, y_2, \ldots, y_n\}(n \geq 1) \tag{5}$$

The matching probability is illustrated as follows

$$q = \{q_1, q_2, \ldots, q_n\} 0 \leq q_i \leq 1, i = 1, 2, \ldots, n \tag{6}$$

Under constraints, the values are,

$$\sum_{i=1}^n q_i = 1 \tag{7}$$

As a result, we may describe the system's information entropy as

$$H = -\sum_{i=1}^{n} q_i \ln q_i \tag{8}$$

Power spectral entropy is the entropy of the resulting power spectrum from an FFT transformation of a time series signal. In a nutshell, the algorithm is as follows:

1. The Discrete Fourier Transform (DFT) of a signal may be calculated using the Fast Fourier Transform (FFT) and is represented as $y(\omega_i)$, where i is a frequency. Following is the formula used to determine the power spectral density (PSD),

$$\hat{q}(\omega_i) = \frac{1}{N}|y(\omega_i)|^2 \tag{9}$$

2. Normalize data $\widehat{P}(\omega_i)$, to derive a function of power spectral density,

$$q_i = \frac{\hat{q}(\omega_i)}{\sum_i \hat{q}(\omega_i)} \tag{10}$$

3. The PSE may be calculated using Eq. (10). Time uncertainty in the frequency domain may be quantified using PSE. When the peak of the power spectrum of EEG signals is narrow, the entropy of the signal is low. In other words, if a wave represents orderliness, then the signal's complexity is low; if the spectrum peak represents smoothness, then the signal's entropy value is high. As a result, the PSE can accurately portray the spectral make-up of EEG data.

3.3 Emotion Recognition

The feature extracted from the two-feature extractor are separately given as a input for the three different types of the classifier listed below,

1. **K-Nearest Neighbors**: K-nearest neighbors (KNN) is a supervised machine learning technique that uses similarities to existing data sets for prediction and classification. The distance metric in (11) may be used to measure the degree of emotional feature similarity between points in the test set and the training set.

$$\text{Cosine}(x_i, x = y_i) = \frac{\sum_{i=1}^{n} Y_i X_i}{\sqrt{\sum_{i=1}^{n} (Y_i)^2}\sqrt{\sum_{j=1}^{n} (X_i)^2}} \tag{11}$$

   where X_i are numbers that should be able to categorize their emotional output, and Y_i are values that are known to be legitimate outputs.
2. **Support Vector Machine**: The method is an expansion of the SVM for use in regression analysis. Depending on the kernel function, it might be linear or

nonlinear. Instead of looking for a hyperplane to solve the regression issue, it locates a tube around the function that is insensitive to changes of up to epsilon. To make predictions about feelings, the models were first trained on the training set.

3. **Naive Bayes**: Bayes' theorem has been used here for classification on the premise that predictors have no effect on classification. The Naive Bayes classifier makes the assumption that characteristics inside a given class do not correlate with one another.

The Naive Bayes model may be used to construct and analyze very large datasets. This model is a very effective classification approach despite its apparent simplicity, and it can handle complex circumstances with ease. The following equation may be used to get the posterior probability using Bayes' theorem:

$$v(a/y) = (v(y/a)v(a))/v(y) \tag{12}$$

where $v(a/y)$ is the predictor's prior probability, $v(y)$ is the likelihood, and $v(a)$ is the posterior probability of the class given the predictor's knowledge of the class distribution.

4 Performance Analysis

This section will assess the efficacy of the proposed technique. The whole experiment was carried out using MATLAB as the platform for the operating system. To demonstrate how useful the proposed network is, an examination of its performance in terms of several metrics may be carried out.

The suggested system for emotion recognition and its simulated output was demonstrated in Fig. 2.

Figure 3 displays a loss of 0.1 and a mean squared error of 0.02.

With the suggested method, the average accuracy of EEG-based emotion identification is raised to 98% by the integration of feature extraction and a classification model. Figure 4 shows the test and learn accuracy rates. Accuracy of 0.9826 and MSE of 0.008 are achieved in the training model after 100 epochs.

From the results obtained, both the PCA and PSE have approximately same level of performance over extracting the features as depicted in Fig. 5.

All the projected versus actual charts display the predictions the models made on the test features with the number of features utilized for predictions as depicted in Fig. 6. By comparing the projected and actual values, here all of the models provide a good match to the data and keep the error down to a minimum. For the validation set, the SVM achieved an accuracy of 0.98, showing its efficacy. Almost a straight line appears when plotting test predictions versus test with the KNN and Naïve Bayes, indicating a close match.

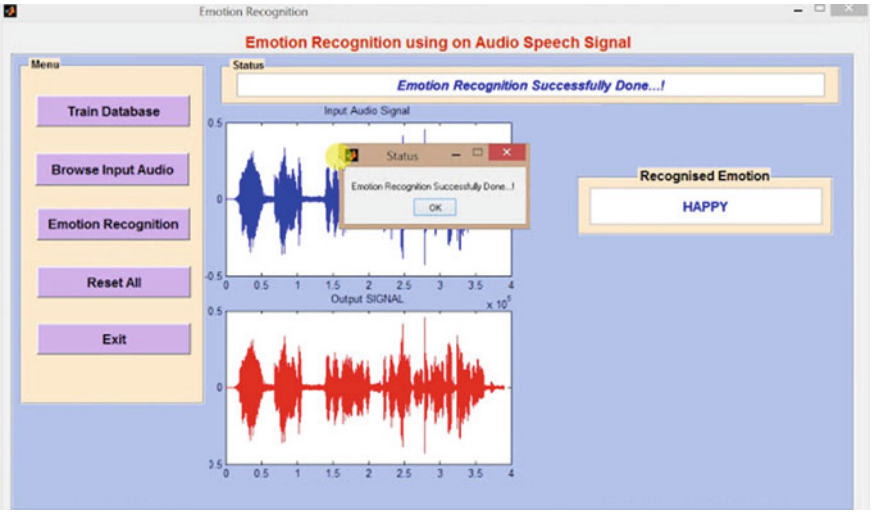


Fig. 2 Emotion recognition on audio speech signal

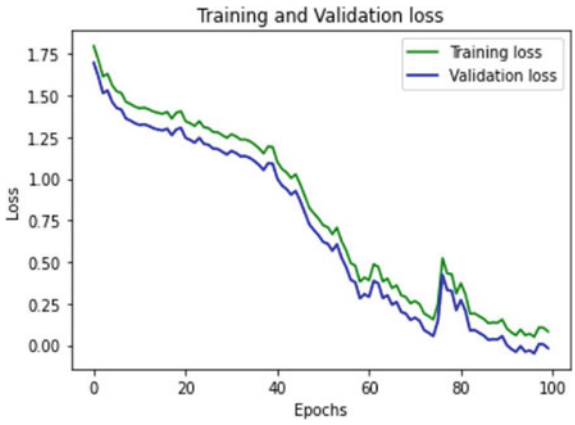


Fig. 3 Epoch versus loss

The effectiveness of the proposed technique may be shown by contrasting it with the current system [20].

Figure 7 shows that when compared to other current mechanisms, the proposed technique achieves more accurate results for emotion prediction.

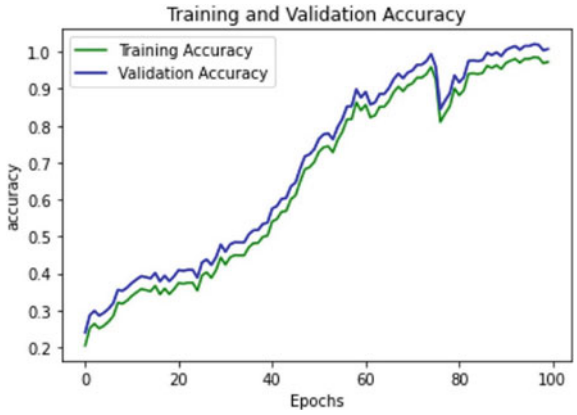


Fig. 4 Epoch versus accuracy

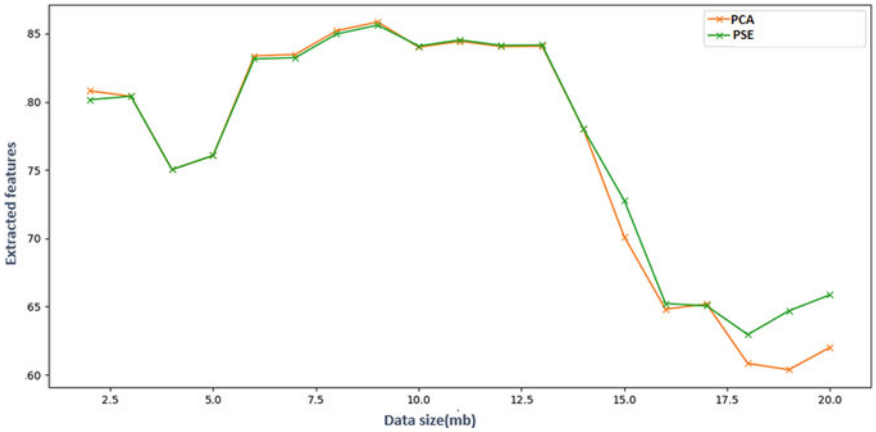


Fig. 5 Data size versus extracted features

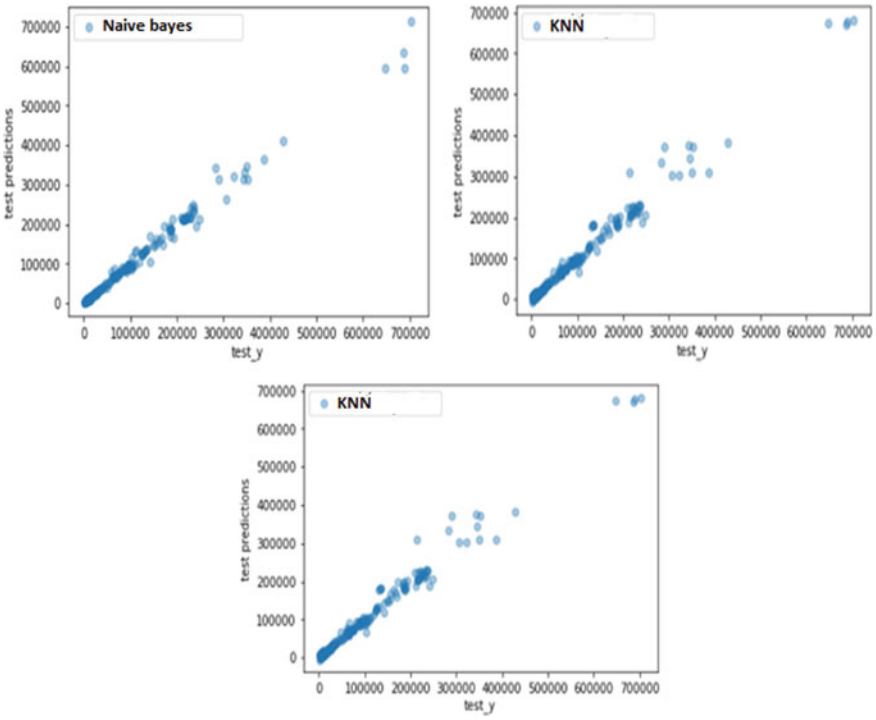


Fig. 6 Prediction rate analysis

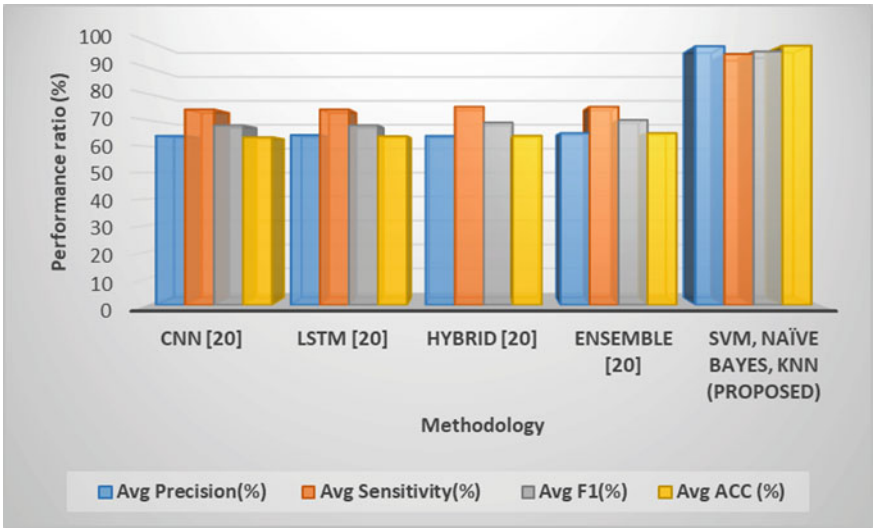


Fig. 7 Performance ratio analysis

Table 1 Comparative performance analysis

Classification methods	Avg precision (%)	Avg sensitivity (%)	Avg F1 (%)	Avg ACC (%)
CNN [20]	64	74	68	63.50
LSTM [20]	64.3	74	68	63.89
Hybrid [20]	64	75	69	64.02
Ensemble [20]	65	75	70	65.00
SVM, Naive Bayes, KNN (Proposed)	98	95	96	98.2

5 Conclusion

The suggested PCA with PSE and machine learning technique using KNN, Naive Bayes, and SVM model has achieved good accuracy in comparison with earlier approaches. The results of contrasting the proposed model with the current methods are given in Table 1. The table shows that the suggested strategy is superior to the current best practices. Future work might broaden the focus of this research to include additional forms of mental illness and analysis of stress factors. Participants may range from active-duty military members to IT specialists to young adults to college students and beyond. There are several evolutionary optimization strategies that might be used.

References

1. George FP, Shaikat IM, Hossain PSF, Parvez MZ, Uddin J (2019) Recognition of emotional states using EEG signals based on time-frequency analysis and SVM classifier. Int J Electr Comput Eng 9:1012–1020
2. Alhagry S, Aly A, Reda A (2017) Emotion recognition based on EEG using LSTM recurrent neural network. Int J Adv Comput Sci Appl 8:8–11
3. Yin Y, Zheng X, Hu B, Zhang Y, Cui X (2020) EEG emotion recognition using fusion model of graph convolutional neural networks and LSTM. Appl Soft Comput 100:106954
4. Cimtay Y, Ekmekcioglu E (2020) Investigating the use of pretrained convolutional neural network on cross-subject and cross-dataset EEG emotion recognition. Sensors 20:2034
5. Wei C, Chen L-L, Song Z-Z, Lou X-G, Li D-D (2020) EEG-based emotion recognition using simple recurrent units network and ensemble learning. Biomed Sig Process Control 58:101756
6. Chao H, Liu Y (2020) Emotion recognition from multi-channel EEG signals by exploiting the deep belief-conditional random field framework. IEEE Access 8:33002–33012
7. Pandey P, Seeja K (2019) Subject independent emotion recognition from EEG using VMD and deep learning. J King Saud Univ Comput Inf Sci 11:1–9
8. Misulis KE, Abou-Khalil B, Sonmezturk H, Ess KC (2022) Atlas of EEG, seizure semiology, and management. Oxford University Press, Oxford
9. Michielli N, Acharya UR, Molinari F (2019) Cascaded LSTM recurrent neural network for automated sleep stage classification using single-channel EEG signals. Comput Biol Med 106:71–81. https://doi.org/10.1016/j.compbiomed.2019.01.013

10. Qazi E-U-H, Hussain M, Aboalsamh H, Ullah I (2019) Automatic Emotion Recognition (AER) system based on two-level ensemble of lightweight deep CNN models. arXiv [Preprint]. https://doi.org/10.48550/arXiv.1904.13234
11. Korotkova T, Ponomarenko A, Monaghan CK, Poulter SL, Cacucci F, Wills T et al (2018) Reconciling the different faces of hippocampal theta: the role of theta oscillations in cognitive, emotional and innate behaviors. Neurosci Biobehav Rev 85:65–80. https://doi.org/10.1016/j.neubiorev.2017.09.004
12. Newson JJ, Thiagarajan TC (2019) EEG frequency bands in psychiatric disorders: a review of resting state studies. Front Hum Neurosci 12:521. https://doi.org/10.3389/fnhum.2018.00521
13. Ramzan M, Dawn S (2019) Learning-based classification of valence emotion from electroencephalography. Int J Neurosci 129:1085–1093
14. Fitzgerald JM, Gorka SM, Kujawa A, DiGangi JA, Proescher E, Greenstein JE et al (2018) Neural indices of emotional reactivity and regulation predict course of PTSD symptoms in combat-exposed veterans. Prog Neuropsychopharmacol Biol Psychiatry 82:255–262
15. Suhaimi NS, Mountstephens J, Teo J (2020) EEG-based emotion recognition: a state-of-the-art review of current trends and opportunities. Comput Intell Neurosci 2020:8875426
16. Topic A, Russo M (2021) Emotion recognition based on EEG feature maps through deep learning network. Eng Sci Technol Int J 24:1442–1454. https://doi.org/10.1016/j.jestch.2021.03.012
17. Wang J, Wang M (2021) Review of the emotional feature extraction and classification using EEG signals. Cogn Robot 1:29–40. https://doi.org/10.1016/j.cogr.2021.04.001
18. Pereira ET, Gomes HM, Veloso LR, Mota MRA (2018) Empirical evidence relating EEG signal duration to emotion classification performance. IEEE Trans Affect Comput 12:154–164. https://doi.org/10.1109/TAFFC.2018.2854168
19. Liu J, Wu G, Luo Y, Qiu S, Yang S, Li W et al (2020) EEG-based emotion classification using a deep neural network and sparse autoencoder. Front Syst Neurosci 14:43. https://doi.org/10.3389/fnsys.2020.00043
20. Iyer A, Das SS, Teotia R, Maheshwari S, Sharma RR (2023) CNN and LSTM based ensemble learning for human emotion recognition using EEG recordings. Multimedia Tools Appl 82(4):4883–4896

Semantic Image Segmentation of Agricultural Field Problem Areas Using Deep Neural Networks Based on the DeepLabV3 Model



Aleksey Rogachev, Ilya Belousov, and Dmitry Rogachev

Abstract In agricultural production, periodic monitoring and determination of the state of crops are required. An additional complexity is the multicomponent structure of the analyzed images. For the purpose of intelligent analysis of agricultural fields, RGB image databases were formed and divided into four classes. An analysis of variants of deep neural network (DNN) architectures based on U-net is carried out. DNN has been developed for the segmentation of problem areas in fields. An experimental quality assessment was carried out on the DeepLabV3 architecture in combination with ResNet50. The constructed DNN family based on DeepLabV3 with ResNet50 showed efficiency of distribution and sufficient speed in segmentation of fragments of agricultural fields and determination of the state of crops. It is established that the increase in accuracy of segmentation of agricultural field images according to the "dice coefficient" criterion is limited by the resolution and quality of manual marking of images.

Keywords Agricultural culture · Neural network recognition · Segmentation · Artificial neural networks · U-net architecture

1 Introduction

The assessment of the state of crops in agricultural fields over large areas is quite a labor-intensive task [1]. Identification of the biological state of agrophytocenoses during their vegetation is possible using artificial intelligence (AI) methods. Among the AI methods that can be used for computer neural network detection of the state

A. Rogachev · I. Belousov
26 Universitetskiy Ave., Volgograd 400002, Russian Federation

D. Rogachev (✉)
VNIIGiM named after Kostyakov, Moscow, Russian Federation
e-mail: rogachev.soft@gmail.com

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2023

S. Shakya et al. (eds.), *Fourth International Conference on Image Processing and Capsule Networks*, Lecture Notes in Networks and Systems 798,
https://doi.org/10.1007/978-981-99-7093-3_30

of crops, it is possible to distinguish the semantic segmentation of their color images [2, 3] by the required classes.

Input information flows are obtained by remote sensing methods and using data from other functional systems, including unmanned aerial vehicles [4, 8]. Earth remote sensing technologies are widely used in the study of the natural environment, soil reclamation, meteorology, and other fields. These technologies are used, along with unmanned aerial vehicles, artificial Earth satellites [4–7]. In the field of agriculture, accurate harvest information on the structure of plantings is important background data for optimizing and adjusting the structure of plantings and the output of agricultural subsidies [10–14].

The specific complexity of the task of assessing the condition and development of crops of various agricultural crops is the multicomponence of the structures of the analyzed multicolor images, which determines the use of a systematic approach in the process of computerized intellectual research, including the formation of databases, their preprocessing, and direct analysis (patents RU 2688234, IPC G06Q 10/06, A01G 9/00; 2723189, IPC A01G 7/00, G06N 5/00, G06N 20/00, G06Q 50/02, etc.). Solving the problems of semantic segmentation of images of agricultural fields provides forecasting of yields and improves the quality of agricultural production management [8, 9, 26].

Formulation and solution of the problem of semantic segmentation, or "pixel-by-pixel classification", allows obtaining the results demanded by the practice of agricultural production management.

At the same time, a number of scientific and methodological problems, including the methodology of selecting optimal sizes and optical resolution of images of agricultural fields, the structure of a database of multiclass images, and the justification of the optimal combination of hyperparameters of deep neural networks (DNN) for intellectual support of decision-making, require modification of known and developed approaches and solutions.

2 Materials and Methods

Methodological approaches to the construction and training of DNN were based on the recommendations of A. Sozykin, set out in the review article [22]. During development, the built-in libraries of the PyTorch framework were used, while the segmentation DNN architecture was selected from the modules available for PyTorch. In the course of research, the accuracy of the compared architectures was evaluated based on the "COCO train2017" dataset (http://cocodataset.org/#home).

To test the hypotheses put forward, a network of the DeepLabV3 ResNet50 architecture was chosen, which is characterized by an accuracy comparable to DeepLabV3 ResNet101 but, at the same time, learns significantly faster. Architecturally, it is a DeepLabV3 based on ResNet50, which contains a sequence of convolutional layers where each subsequent result will be combined with the previous one.

The well-known segmentation quality criteria "Dice coefficient" (1) were used

$$\mathrm{DSC} = \frac{2|X \cap Y|}{|X| + |Y|} \tag{1}$$

and «Jaccard coefficient» (2)

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} \tag{2}$$

DNN's training was implemented on a graphics accelerator using CUDA technology. The Nvidia RTX 2080Ti graphics processor supported by the CUDA Toolkit 11.4 library was used. Various types of image augmentation were used to provide DNN's training; e.g., the method allowed diversification diversify the database and dataset.

3 Results and Discussion

3.1 Formation of Datasets and Architectures for Solving the Problem of Segmentation of Images of Agricultural Fields

To form a database and datasets when solving the segmentation problem, color images of agricultural fields with a dimension of 200 × 200 pixels obtained using UAVs were used. RGB color images were marked up into four classes: high-quality field, defective field, cultivated field (not shown), and non-field (other objects). When marking up the dataset, the segmenting mask was not superimposed on the original image, but was formed in a graphical editor.

To improve the quality of deep DNN training, in 2015, researchers S. Ioffe and Ch. Szegedy proposed the use of special batch normalization regulation layers in neural networks [23]. As is known, learning of the method of error back propagation converges faster in the case when the input data is normalized to the normal distribution, when the mathematical expectation is zero and the variance is one [24]. However, during the propagation of information signals through a neural network, both the mathematical expectation and the variance can change significantly, which negatively affects the learning outcomes. The researchers mentioned above proposed to perform normalization not only at the entrance to the neural network, but also before each of its layers.

When implementing various modifications of the stochastic gradient descent method, it is possible to separately normalize data for each data packet using mini-batch. In the case when the batch contains m elements of input data x_i $\{x_1, \ldots, x_m\}$, the normalized values are determined by the formula (3):

$$\widehat{x_i} = \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 - \upsilon}} \tag{3}$$

where μ_B is the average value of the data in the batch, σ is the variance, and ε is the stabilizing constant.

After that, scaling and displacement are performed according to the dependence (4)

$$y_i = \gamma \widehat{x_i} + \beta, \tag{4}$$

where y_i is the resulting value, γ and β are the parameters that are determined in the learning process.

Batch normalization is implemented in the form of DNN layers, which can be located in the required places of the structure in the neural network, while they can be used multiple times. In addition, a positive consequence of using batch normalization may be the reduction of epochs and training time, as well as the elimination of the undesirable phenomenon of retraining.

Note that the layers of batch normalization, as well as the methods of normalized selection of initial weights when initializing the DNN training process, help to get rid of the problem of disappearing gradient when training neural networks of deep architecture containing dozens of hidden inner layers. Consequently, the problem of the vanishing gradient, according to some researchers, can now be considered solved [24].

Previously, the authors' studies were based on the classification of the specified images with a size of 200 × 200 pixels, which gave only a general idea of the problem, for the case when the detected problem of the field area had a sufficiently large size. The complexity of image processing with this approach increases quadratically with an increase in the linear size of the images, which greatly increases the requirements for the hardware. The use of a neural network based on the segmentation approach (Fig. 1) allowed to reduce the load on processors due to the fact that it became possible to compress images.

The U-Net, Siamese neural network (SNN), and DeepLabV3 architectures use cascades of convolutional CNN layers [17, 25] with 3 × 3 convolution cores to highlight specific "features" of images that can be used to solve the segmentation problem (Fig. 3).

After multiple compressions of images with convolutional layers, both described neural networks form a "new" segmented image from images, obtained at various stages of processing the original image, which once again correlates with the final result of the backbone.

In the experiments conducted, the training of the DNN was carried out both for compressed images and without it, by splitting the original image to 500 × 500 pixels.

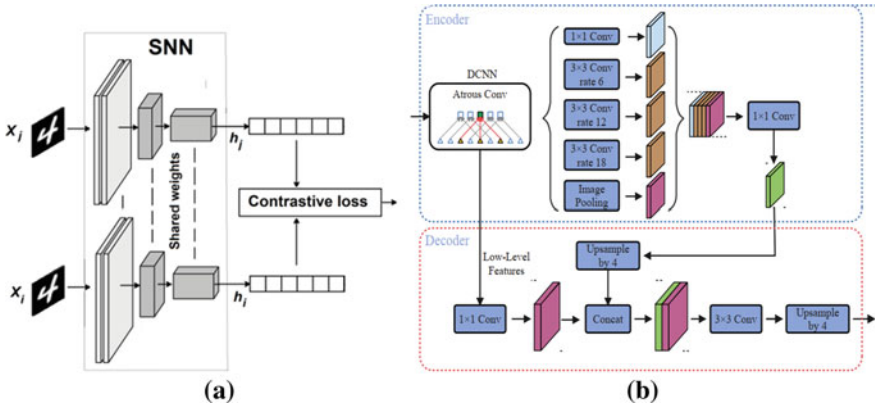


Fig. 1 DNN segmentation architectures: **a** SNN-Net; **b** DeepLabV3

A fragment of a program developed in Python, including loading the used libraries of the PyTorch framework (Fig. 2), as well as marked-up input images, is presented below.

U-Net and DeepLabV3 were used as the architecture in the search experiments (Fig. 3).

3.2 Metrics and Results of Segmentation of Images of Agricultural Fields

Additionally, Intersection over Union (IoU) indicators were used to characterize the quality of recognition on a test sample for various DNNs architectures.

Below is a script fragment (Fig. 4) of the implementing the learning algorithm of the developmental program.

Preliminary experiments have shown that the architecture of DeepLabV3 ResNet50 and DeepLabV3 ResNet101 demonstrates a higher quality of work compared to FCN ResNet101.

Typical graphs of the change in the loss function depending on the number of processed image packets are shown in Fig. 5. The learning process is characterized by a relatively monotonous decrease in the average values of the loss function (Average Loss) with significant amplitude jumps of the described function itself.

This abrupt nature of the change in the loss function indicators during the pro-sizing of various baths is due to the use of stochastic gradient descent and the significant heterogeneity of the colorographic parameters of the processed color images of agricultural fields.

The highest Average Loss values are 1.5 at the beginning of the learning process. After processing approximately 4000 batches, the 0.5 Average Loss values drop to

```
import torch
import numpy as np
import matplotlib.pyplot as plt
from torch import nn
from torch import optim
from torchvision import datasets, transforms

def load_split_train_test(datadir, valid_size=.2):
    train_transforms = transforms.Compose([transforms.Resize(256),
                                           transforms.CenterCrop(224),
                                           transforms.ToTensor(),
                                           transforms.Normalize(mean=[0.5, 0.5, 0.5], std=[0.5,
0.5, 0.5])])
    test_transforms = transforms.Compose([transforms.Resize(256),
                                          transforms.CenterCrop(224),
                                          transforms.ToTensor(),
                                          transforms.Normalize(mean=[0.5, 0.5, 0.5], std=[0.5,
0.5, 0.5])])
    train_data = datasets.ImageFolder(datadir, transform=train_transforms)
    test_data = datasets.ImageFolder(datadir, transform=test_transforms)
    num_train = len(train_data)
    indices = list(range(num_train))
    split = int(np.floor(valid_size * num_train))
    np.random.shuffle(indices)
    from torch.utils.data.sampler import SubsetRandomSampler
    train_idx, test_idx = indices[split:], indices[:split]
    train_sampler = SubsetRandomSampler(train_idx)
    test_sampler = SubsetRandomSampler(test_idx)
    trainloader_t = torch.utils.data.DataLoader(train_data, sampler=train_sampler,
batch_size=128)
    testloader_t = torch.utils.data.DataLoader(test_data, sampler=test_sampler,
batch_size=128)
    return trainloader_t, testloader_t
```

Fig. 2 PyTorch framework

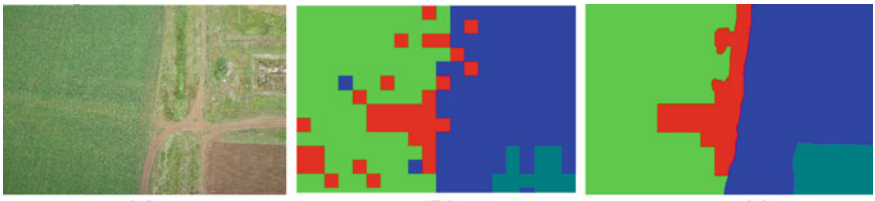


Fig. 3 İmages of agricultural fields. **a** Initial, **b** marked; **c** segmented

```
data_dir = 'fields'
trainloader, testloader = load_split_train_test(data_dir, .2)
device = torch.device("cuda" if torch.cuda.is_available()
                      else "cpu")

print(device)
model = torch.hub.load('facebookresearch/WSL-Images',
'resnext101_32x16d_wsl')
for param in model.parameters():
    param.requires_grad = False

model.fc = nn.Sequential(nn.Linear(2048, 1000, True),
                         nn.ReLU(),
                         nn.Dropout(0.2),
                         nn.Linear(1000, 10),
                         nn.LogSoftmax(dim=1))
criterion = nn.NLLLoss()
optimizer = optim.Adam(model.fc.parameters(), lr=0.003)
model.to(device)

epochs = 6
steps = 0
running_loss = 0
print_every = 10
train_losses, test_losses = [], []
for epoch in range(epochs):
    for inputs, labels in trainloader:
        steps += 1
        inputs, labels = inputs.to(device), labels.to(device)
        optimizer.zero_grad()
        logps = model.forward(inputs)
        loss = criterion(logps, labels)
        loss.backward()
        optimizer.step()
        running_loss += loss.item()

        if steps % print_every == 0:
            test_loss = 0
            accuracy = 0
            model.eval()
            with torch.no_grad():
                for inputs, labels in testloader:
```

Fig. 4 Script fragment

```
                inputs, labels = inputs.to(device), labels.to(device)
                logps = model.forward(inputs)
                batch_loss = criterion(logps, labels)
                test_loss += batch_loss.item()
                ps = torch.exp(logps)
                top_p, top_class = ps.topk(1, dim=1)
                equals = top_class == labels.view(*top_class.shape)
                accuracy += torch.mean(equals.type(torch.FloatTensor)).item()
        train_losses.append(running_loss / len(trainloader))
        test_losses.append(test_loss / len(testloader))
        print(f"Epoch {epoch + 1}/{epochs}.. "
            f"Train loss: {running_loss / print_every:.3f}.. "
            f"Test loss: {test_loss / len(testloader):.3f}.. "
            f"Test accuracy: {accuracy / len(testloader):.3f}")
        running_loss = 0
        model.train()

torch.save(model, 'myFirst-16-5-005.pth')
```

Fig. 4 (continued)

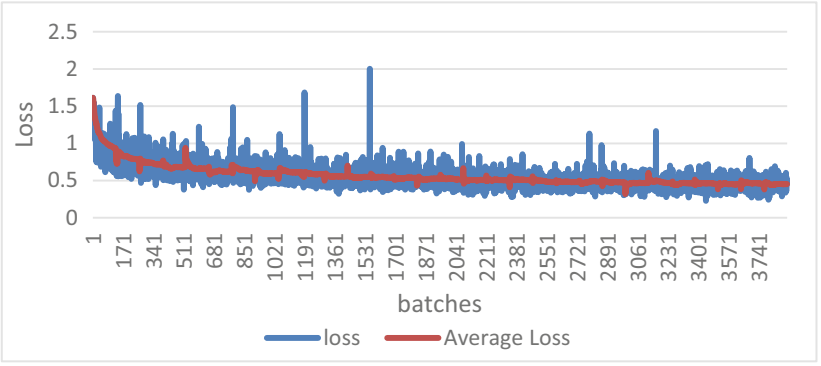


Fig. 5 Typical graphs of changes in the loss function depending on the number of processed image batches

the 0.5 level and stabilize without retraining. However, the presented packet dependence does not very clearly characterize the learning process of the developed neural network, since within each of the epochs of the processing of individual packets, only collectively determines the vectors of changes in the weight coefficients of the neurons of the trained network.

As a result of the segmentation-type DNN construction, it was possible to obtain more detailed masks for the source images (Fig. 5).

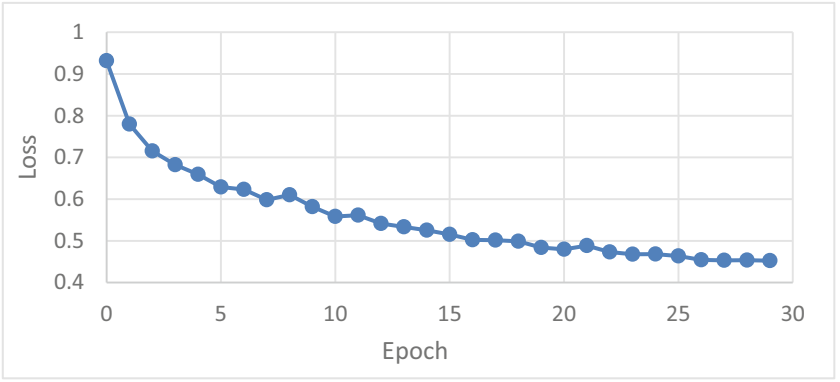


Fig. 6 Diagram of changes in the loss function "Average Loss" depending on the number of training epochs

A visual representation of the dynamics of changes in the average value of the Average Loss function, characterizing the speed of the neural network learning process depending on the number of training epochs, is shown in Fig. 6.

The highest loss values are 0.95 at the beginning of the learning process, and after about 30 epochs, the loss values stabilize without retraining at 0.45.

Further reduction requires in-depth optimization, for example, using the Karas Tuner software that implements various optimization algorithms.

The conducted study of the segmentation of problem areas of agricultural fields confirmed that the use of the considered architectures of the DeepLabV3 ResNet50 type allows us to successfully solve the problems of identifying the reclamation state of agricultural fields by the method of multiclass segmentation, as well as to identify the state of agricultural plants (Fig. 7).

Based on the developed neural network, a SaaS service has been created, the working screen of which is shown in Fig. 8. The service's working screen shows an image of the original snapshot of an agricultural field plot with a color mask superimposed on it.

The red zones of the mask denote areas of unseeded fields and non-agricultural fields in general, translucent red zones—problem areas of agricultural fields. The results of intelligent recognition and segmentation by recognized classes can be used both for expert "manual" analysis and for subsequent use of cascading neural networks or third-party services.

Analysis of the results of the use of classification-type neural networks has shown that DNN can successfully solve the problems of recognizing the reclamation state of fields and the development of crops. Comparing the results obtained with previous versions, it was possible to achieve an increase in accuracy by 9…12%. At the same time, the increase in the accuracy of recognition of classified areas of agricultural fields is limited both by the composition (quality) and the volume of the dataset, as well as its balance [8, 9].

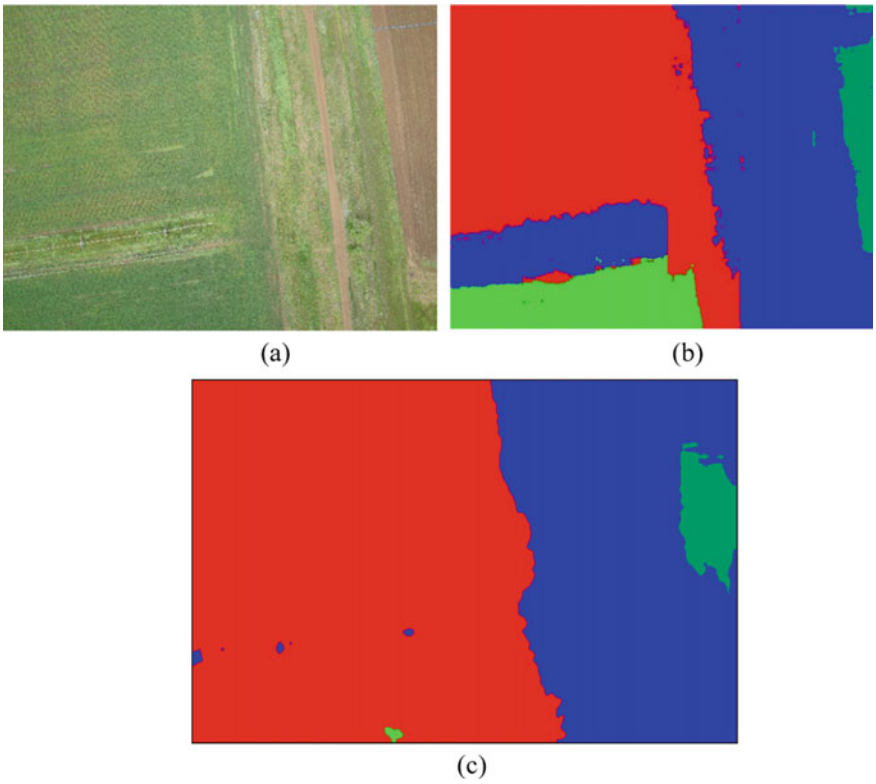


Fig. 7 Segmentation of the image of agricultural fields: **a** the original image; **b** processing with clusters of 500 × 500 size; **c** image processing when scaling up to 500 × 500



Fig. 8 Interface: recognition result with superimposed mask layers

However, when increasing the volume and resolution of images, there are a number of problems: a linearly increasing load on the processor and low accuracy of the boundaries.

The improvement of the accuracy and quality of segmentation of images of recognized agricultural fields at different stages (phases) of the vegetation cycle [9, 16] is constrained by the limited resolution and accuracy of manual markup of the dataset used for training, as well as by the hyperparameters of the DDN [3, 15, 18, 20].

As a direction for continuing research, it is planned to improve the architecture of the segmentation program. In particular, the "attention" mechanism will be implemented. The improvement of the program will provide additional opportunities for analyzing time series (BP) of retrospective images of agro-fields during vegetation [19] in order to intelligently assess the dynamics of plant development [21], including solving the problem of neural network regression.

4 Conclusions

Analysis of the results of the study of the use of neural networks for segmentation of agricultural fields showed that the DNN's of the considered DeepLabV3 architectures in combination with ResNet50 can solve the problems of segmentation and identification of the reclamation state of agricultural fields, as well as the state of plant development, while the mentioned architecture demonstrated advantages.

The increase in the accuracy of segmentation of images of recognized agricultural fields is constrained by the limited resolution and accuracy of the manual markup of the dataset used for training.

Further improvement of the segmentation program architecture can be ensured by the inclusion of the "attention" mechanism in the architecture of the developed DNN's.

The improvement of DNN will provide additional opportunities for the analysis of VR images of agricultural fields during the growing season in order to intelligently assess the dynamics of plant development.

Acknowledgements The publication has been prepared with the financial support of the RFBR under project No. 20-37-90142.

References

1. Saiz-Rubio V From smart farming towards agriculture 5.0: a review on crop data management. https://www.mdpi.com/2073-4395/10/2/207/htm
2. Alberto G-G, Sergio O-E, Sergiu O, Victor V-M, Jose G-R A review on deep learning techniques applied to semantic segmentation. https://doi.org/10.48550/arXiv.1704.06857
3. Badrinarayanan V, Kendall A, Cipolla R SegNet: a deep convolutional encoder-decoder architecture for image segmentation. http://docs.cntd.ru/document/902361843

4. Xu B et al (2004) Remote sensing monitoring on dynamic status of grassland productivity and animal loading balance in Northern China. In: IGARSS 2004. 2004 IEEE international geoscience and remote sensing symposium, Anchorage, AK, USA, vol 4, pp 2306–2309. https://doi.org/10.1109/IGARSS.2004.1369747
5. Pelletier C, Valero S, Inglada J, Champion N, Dedieu G (2016) Assessing the robustness of Random Forests to map land cover with high resolution satellite image time series over large areas. Rem Sens Environ 187:156–168
6. Zhang L, Yu W, Li G, Zhang H (2016) An approach for flood inundated duration extraction based on Level Set Method using remote sensing data. In: 2016 IEEE international geoscience and remote sensing symposium (IGARSS), Beijing, China, pp 1820–1822. https://doi.org/10.1109/IGARSS.2016.7729467
7. Zeng Z-C, Wang Y, Pongetti TJ, Gong F-Y, Newman S, Li Y, Natraj V, Shia R-L, Yung YL, Sander SP (2020) Tracking the atmospheric pulse of a North American megacity from a mountaintop remote sensing observatory. Rem Sens Environ 248:112000
8. Tokarev KE et al (2015) Crops reclamation management based on hybrid neuro-fuzzy systems. IOP Conf Ser-Earth Environ Sci 421:042015. https://doi.org/10.1088/1755-1315/421/4/042015
9. Melikhova EV, Rogachev AF (2019) Computer simulation and optimization of parameters of configuration of the contour of moistening under drip irrigation of agricultures. In: Popkova EG (ed) Ubiquitous computing and the internet of things: prerequisites for the development of ICT. Studies in computational intelligence, vol 826, pp 1193–1201. https://doi.org/10.1007/978-3-030-13397-9_122
10. Soloviev RA, Telpukhov DV, Kustov AG (2017) Automatic segmentation of satellite images based on the modified convolutional neural network UNET. Eng Bull Don (4)
11. Jeong S, Ko J, Yeom JM (2021) Predicting rice yield at pixel scale through synthetic use of crop and deep learning models with satellite data in South and North Korea. Sci Total Environ 802(2):149726
12. Xu J, Yang J, Xiong X et al (2021) Towards interpreting multi-temporal deep learning models in crop mapping. Rem Sens Environ 264:112599. https://doi.org/10.3390/rs13224668
13. Ofori-Ampofo S, Pelletier C, Lang S (2021) Crop type mapping from optical and radar time series using attention based deep learning. Rem Sens 13:4668
14. Kussul N, Lavreniuk M, Skakun S et al (2017) Deep learning classification of land cover and crop types using remote sensing data. IEEE Geosci Rem Sens Lett (99):1–5
15. Alekseev AV, Rozaliev VL, Orlova YA, Zaboleeva-Zotova AV (2016) Context-sensitive image analysis for coloring nature images. Adv Intell Syst Comput 451:133–141
16. He K, Zhang X, Ren X, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the 29th IEEE conference on computer vision and pattern recognition—CVPR, Las Vegas, Nevada, USA, pp 770–778
17. Fezan Review DeepLabv3 (Semantic Segmentation) (2020). https://medium.com/swlh/review-deeplabv3-semantic-segmentation-52c00ddbf28d
18. Adrian R Intersection over Union (IoU) for object detection. https://www.pyimagesearch.com/2016/11/07/intersection-over-unioniou-for-object-detection
19. Seeland M, Mäder P (2021) Multi-view classification with convolutional neural networks. PLoS ONE 16(1):e0245230. https://doi.org/10.1371/journal.pone.0245230
20. Tsang S-H Review: DeepLabv3. Atrous Convolution (Semantic Segmentation). https://towardsdatascience.com/review-deeplabv3-atrous-convolution-semantic-segmentation-6d818bfd1d74
21. Rogachev AF, Belousov IS (2022) Neural network identification of problem areas of the state of crops by methods of artificial intelligence. In: Proceedings of the Lower Volga Agro-University Comp, vol 3(67), pp 459–466. https://doi.org/10.32786/2071-9485-2022-03-52
22. Sozykin AV (2017) Review of deep neural network training methods. Bull SUSU Ser Comput Math Comput Sci 6(3):28–59. https://doi.org/10.14529/cmse170303
23. Ioffe S, Szegedy C (2015) Batch normalization: accelerating deep network training by reducing internal covariate shift. In: JMLR workshop and conference proceedings. Proceedings of the

32nd international conference on machine learning, Lille, France, 06–11 July 2015, vol 37, pp 448–456
24. He K, Zhang X, Ren S et al (2016) Deep residual learning for image recognition. In: 2016 IEEE conference on computer vision and pattern recognition, Las Vegas, NV, USA, 27–30 June 2016, pp 770–778. https://doi.org/10.1109/CVPR.2016.90
25. Utkin L, Kovalev M, Kasimov E (2021) An explanation method for Siamese neural networks. In: Voinov N, Schreck T, Khan S (eds) Proceedings of international scientific conference on telecommunications, computing and control. Smart innovation, systems and technologies, vol 220. Springer, Singapore. https://doi.org/10.1007/978-981-33-6632-9_19
26. Sabour S, Fross N, Hinton GE (2017) Dynamic routing between capsules. In: Neural information processing systems (NIPS)

A Unified Deep Learning Model for Multi-Satellite Image Classification of Land Use and Land Cover



M. S. Babitha, A. Diana Andushia, and A. Mehathab

Abstract Satellite images act as windows to the world, providing us with detailed snapshots of Earth's land surface. Utilizing various spectral bands, sensors aboard satellites detect and record information about different surface features, including vegetation, water bodies, urban areas, and barren land. By analyzing the spatial distribution and spectral characteristics of these features, scientists can classify and map land cover types with remarkable precision. The Landsat 8 satellites capture multispectral imagery with an image feature of 30 m for the optical spectrum, along with a thermal band at 100 m resolution. The revisit time for global coverage is approximately 16 days. The Sentinel program, operated by the ESA, includes multiple satellites such as S2 and S1. Sentinel-2 captures imagery at an analysis rate of 10 m for the optical spectrum, with repeatedly visiting in five days. S1 provides SAR data with an image analysis rate of 10–40 m, allowing imaging regardless of weather conditions or daylight. This paper proposes a deep learning model which will process different satellite imagery, i.e., a single DL model which process different satellite images for LULC identification. Here, the satellite images form S1, S2, and L8 for the year 2020 are processed using a single convolutional neural network model (CNN) for classification of LULC. Caatinga Forest Region, Brazil, was taken as the study area which includes three states Rio Grande Do Norte, Paraiba and Pernambuco. Deep neural networks are complex and powerful computational models inspired by the intricate structure and functionality of the human brain. These networks consist of multiple layers of interconnected artificial neurons, each performing its own calculation and contributing to the overall learning and decision-making process. Accuracy and Kappa analysis was done with each satellite images and combination of different

M. S. Babitha (✉) · A. Diana Andushia
Department of Electronics and Communication Engineering, Karunya Institute of Technology and Sciences, Coimbatore, India
e-mail: babitham@karunya.edu.in

A. Diana Andushia
e-mail: diana@karunya.edu

A. Mehathab
Data Scientist with Specialization in Remote Sensing, Expertz Lab Technologies, Kochi, Kerala, India

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2023

S. Shakya et al. (eds.), *Fourth International Conference on Image Processing and Capsule Networks*, Lecture Notes in Networks and Systems 798,
https://doi.org/10.1007/978-981-99-7093-3_31

satellite images using Google earth engine analysis. The detail study will boost the land cover classification task.

Keywords Deep learning · Convolutional neural network · Sentinel-1 · Sentinel-2 · Landsat 8 · Land cover classification

1 Introduction

Over time, Earth's land cover undergoes a continuous process of transformation. Human activities, natural processes, and climate change contribute to the altering face of our planet. Satellite sensors capture images from space, offering a synoptic view of large areas of land. These images provide consistent and systematic coverage, allowing for regular updates and monitoring of LULC. Satellite images play a vital role in monitoring and detecting land cover changes, enabling us to understand the drivers behind these transformations and their consequences [1]. By comparing images acquired at different times, scientists can identify areas of deforestation, urban expansion, agricultural changes, and shifts in vegetation cover. These observations provide crucial insights into the dynamic interplay between human societies and the environment. LULC identification is a method of dissecting vast landscapes into distinct categories, discerning the diverse elements that compose the terrestrial canvas. By employing advanced technologies and sophisticated algorithms, land cover classification endeavors to decode the language of nature, translating the visual manifestation of land features into actionable information.

Satellite images play a crucial role in land cover classification, offering valuable insights into the Earth's surface and its changes over time [2]. Specifically, Landsat 8, Sentinel-1, and Sentinel-2 satellites have made significant contributions to this field. Landsat 8, operated by NASA and the US Geological Survey (USGS), captures multispectral images of the Earth's surface at high spatial resolution. It collects data across different frequency range, optical spectrum, etc. These wavelength range allow for the discrimination of various land cover types based on their unique reflectance patterns. Sentinel-1, part of the ESA Copernicus program, utilizes synthetic aperture radar (SAR) technology to acquire images. SAR is particularly advantageous in land cover classification because it can penetrate cloud cover and capture data regardless of weather conditions or time of day. SAR images provide information on the backscattered radar signal, which is influenced by surface roughness, moisture content, and vegetation structure [3]. These characteristics can aid in mapping land cover types like forests, urban areas, and water bodies. Sentinel-2, another mission within the Copernicus program, operates with a multispectral sensor and provides high-resolution images across 13 spectral bands. These bands include visible, near-infrared, and shortwave infrared regions, allowing for precise land cover classification [3]. Sentinel-2 data enables the discrimination of vegetation types, identification of water bodies, monitoring of agricultural practices, and assessment of urban development.

In land cover classification, these satellite images are processed using advanced remote sensing techniques and machine learning algorithms. These methods involve preprocessing steps such as atmospheric correction, radiometric calibration, and geometric registration to ensure accurate and comparable data. Machine learning models like RF or CNN are then applied to train models on labeled data, enabling automated classification of land cover types across large areas [4, 5].

DL techniques, such as CNNs, are designed to handle multi-channel inputs, making them suitable for processing diverse satellite images. Satellite images often consist of multiple bands or channels, each representing different parts of the electromagnetic spectrum [6]. For example, satellite cameras can capture images in optical spectrum. These bands provide valuable information about various land cover characteristics, such as vegetation health, soil moisture, or urban development. To incorporate different satellite images into a single deep learning model, you can stack the image bands together to create multi-channel input data. Each band becomes a separate input channel, and the model learns to extract features from all channels simultaneously.

By training a deep learning model on multi-channel satellite images, the model can learn to leverage the complementary information from different bands and capture complex relationships between them. For instance, it can learn to identify how vegetation health (infrared band) relates to land use patterns (visible band) or how surface temperature (thermal band) correlates with urban heat island effects (infrared band) [7]. The advantage of apply a single DL architecture for processing different satellite images is that it allows the model to generalize across diverse data sources and capture broader patterns and features. It can handle various satellite sensors, resolutions, or spectral ranges, making it more versatile in analyzing different types of satellite imagery.

Convolutional neural networks (CNNs) are a powerful and innovative class of deep learning models specifically designed to process and extract meaningful features from grid-like data, such as images and time series. Inspired by the visual cortex of animals, CNNs have revolutionized computer vision tasks and have also found applications in other domains, such as natural language processing and speech recognition. The highlight of a CNN are convolutional layers, which apply filters (also known as kernels) to the input data [8]. These filters perform convolution operations, scanning the input with a small receptive field, capturing local patterns and features. By stacking multiple convolutional layers, CNNs can progressively learn more complex and abstract representations, capturing both low-level features such as edges and textures, as well as high-level features like object shapes and semantic information. In addition to convolutional layers, CNNs typically include pooling layers, which reduce the spatial dimensions of the feature maps, helping to extract the most salient features while improving computational efficiency. Pooling can be performed using operations like max pooling, which select the maximum value within a certain window, or average pooling, which takes the average value. Once the feature extraction is performed, CNNs employ fully connected layers, also known as dense layers, to transform the extracted features into class probabilities or regression outputs [9].

These layers allow the network to learn complex relationships between the extracted features and the target task, enabling accurate predictions or classifications.

The remaining sections are, Sect. 2 explains the methodology, and Sect. 3 gives the results and discussions and suggests a further direction to modify the area. The final part summarizes the conclusion.

2 Methodology

The general structure of this workflow of land cover classification architecture has shown in Fig. 1. The framework consists of individual modules, and each and every module is connected with its specific strategies. The actual modules of this structure are image database, preprocessing, feature extraction, image classification, and image segmentation.

2.1 Data Acquisition

Large number of datasets used to operate a CNN application depends on several factors, including the complexity of the task, the size and diversity of the dataset, and the desired performance of the model [10]. While there is no fixed rule, it is generally recommended to have a sufficient amount of training data to ensure effective learning and generalization of the model. In general, the more data you have, the better, as it allows the model to capture a wider range of patterns and variations in the data. Large datasets help reduce the risk of overfitting, where the model becomes too specialized in the training data and performs poorly on unseen data. This paper mainly consists of three major satellite images, S1, S2, and L8 present in the Google Earth engine dataset catalogs.

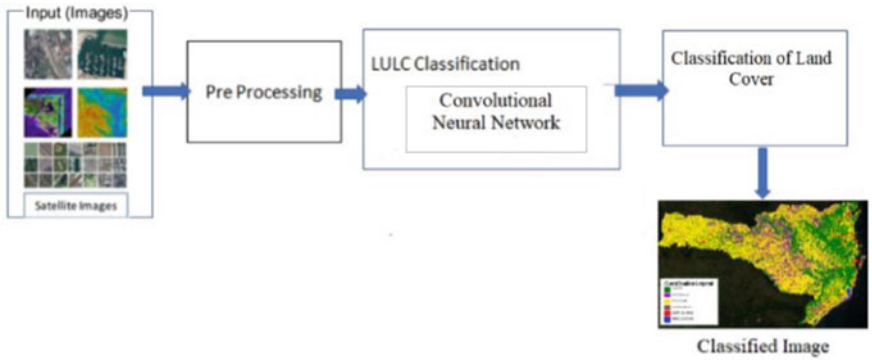


Fig. 1 Flow diagram of land cover classification

Landsat: The Landsat program, initiated by NASA in 1972, currently consists of the Landsat 8 and Landsat 9 satellites. Landsat 8 Tier 1 refers to the highest level of data quality and processing offered by the Landsat 8 satellite mission, which is a joint program between NASA and the United States Geological Survey (USGS). The Tier 1 data from Landsat 8 undergoes rigorous calibration, geometric correction, and atmospheric correction processes to ensure accuracy and consistency. This makes Landsat 8 Tier 1 data highly reliable and suitable for quantitative analysis and long-term monitoring of Earth's land surfaces. The images acquired by Landsat 8 provide valuable information for applications such as LULC identification and monitoring the differences, forest monitoring, and water resource management [11, 12]. The main used spectral bands are,

Landsat: Landsat satellites typically have a broader range of spectral bands, including visible, near-infrared, shortwave infrared, and thermal infrared. This enables precise land cover classification, vegetation monitoring, and thermal analysis for applications such as agriculture, forestry, and urban planning.

Sentinel: Sentinel satellites are equipped with a narrower set of spectral bands, focusing on the visible, near-infrared, and shortwave.

S1 SAR GRD is a satellite mission developed by the European Space Agency (ESA) that employs synthetic aperture radar technology. Unlike traditional optical sensors, SAR operates by emitting microwave pulses and measuring their reflections from the Earth's surface. This enables SAR to overcome limitations imposed by cloud cover and darkness, making it highly suitable for all-weather and day-and-night monitoring.

Sentinel-1 SAR GRD captures high-resolution radar imagery of the Earth's surface, providing valuable information about various features such as land, water bodies, and ice. The SAR images are acquired in "ground range detected" mode, which means that they are processed to produce images with a constant scale, allowing for accurate measurements of distances and surface features. SAR GRD data is widely used for applications such as land cover mapping, disaster monitoring, iceberg detection, and maritime surveillance.

Sentinel-2 MSI (Multispectral Instrument) is another satellite mission led by ESA. It is equipped with a multispectral sensor that captures images in 13 spectral bands, ranging from the visible to the near-infrared spectrum. These bands give important features of the Earth's surface composition, vegetation health, and water quality. Sentinel-2 MSI offers a unique combination of high spatial resolution (10–20 m) and frequent revisits (every five days for the entire Earth), allowing for detailed monitoring and analysis of land and coastal areas [13, 14].

The data from Sentinel-2 MSI is widely used for a variety of applications, including agriculture monitoring, deforestation assessment, urban planning, and natural resource management. The multispectral capabilities of Sentinel-2 MSI enable the creation of detailed and accurate vegetation indices, which are crucial for monitoring vegetation growth, detecting changes, and assessing environmental conditions [2].

In summary, Sentinel-1 SAR GRD offers all-weather, day-and-night imaging using synthetic aperture radar technology, while Sentinel-2 MSI provides multi-spectral data with high spatial resolution and frequent revisits. Landsat 8 Tier 1 data represents the highest quality and accuracy level for the images captured by Landsat 8, enabling detailed analysis and long-term monitoring of Earth's land surfaces. Together, these satellite missions contribute significantly to our understanding of our planet environment and support a collection of information in various fields.

2.2 Preprocessing

Preprocessing involves data cleaning, where missing values, outliers, or errors are identified and handled. This ensures the integrity and quality of the data, reducing the risk of biased or misleading results. It is important to preprocess and normalize the satellite images appropriately before feeding them into the deep learning model. This ensures that the data from different sources are consistent and compatible, allowing the model to learn effectively [15].

Overall, employing a single deep learning model to process different satellite images enables the extraction of valuable insights from multiple data sources, facilitating comprehensive analysis and decision-making in various fields, including environmental monitoring, agriculture, urban planning, and disaster response. Through preprocessing, the neural network can navigate the complexities of the data landscape, uncovering hidden patterns, making accurate predictions, and unleashing the full potential of deep learning in solving real-world problems [16]. For the proposed work, Sentinel-1, Sentinel-2, and Landsat satellite image datasets were used. These datasets are preprocessed for good imagery by Google and the ESA agency beforehand. To get rid of the redundant clouds in the area of study, masking is done by bitmask band which are designed within the dataset for cloud clearing. Masking is done for: Cirrus clouds, opaque clouds, snow/ice filtering, scene classification algorithm (SCL), shadows, and also to control cloud probability after masking and elimination of the above and training process.

2.3 Models for Land Cover Classification

The proposed CNN model with five blocks as shown in Fig. 2 typically refers to a CNN architecture composed of five main components, each representing a block of layers, while the specific configuration and design choices can vary.

Block 1: The first block consists of convolutional layers of 128 units. Convolutional layers apply filters to the input data, extracting local features. This block captures low-level features.

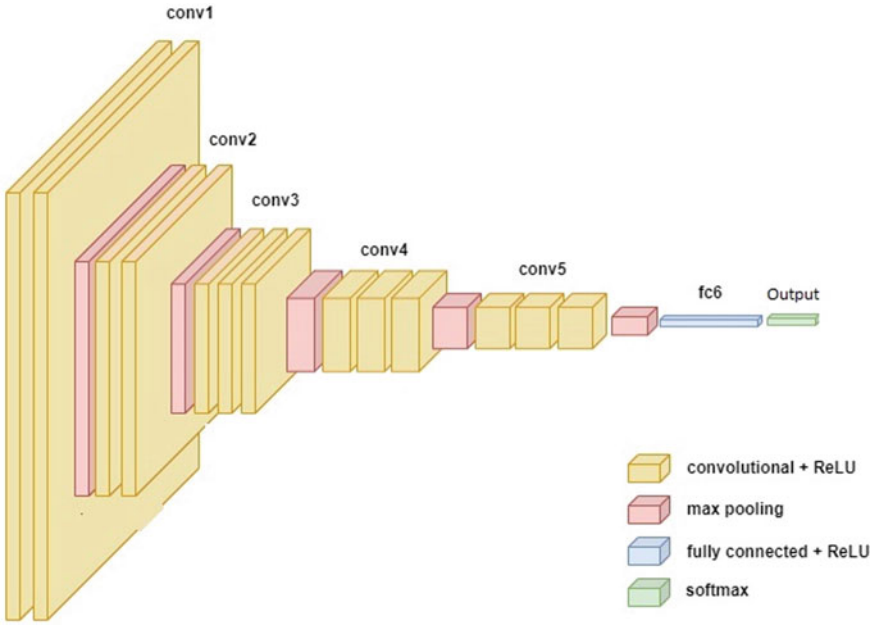


Fig. 2 Block diagram proposed CNN model

Block 2: Similar to the first block, this block contains convolutional layers of 64 units and a pooling layer. It further captures more complex features by building upon the low-level features extracted in Block 1.

Block 3: Continuing the pattern, this block contains convolutional layers of 64 units and a pooling layer. It aims to provide advanced features and patterns from the input data.

Block 4: In this block, additional convolutional layers of 32 units and pooling layers are employed. It further refines the learned features and captures even more complex patterns.

Block 5: This block consists of convolutional layers of 32 units, and these layers take the learned features from the previous blocks and transform them to the next block.

The final block typically consists of fully connected layers, also known as dense layers. These layers take the learned features from the previous blocks and transform them into class probabilities or regression outputs. The developed CNN model incorporated with batch normalization and activation in each block using activation function ReLU, each block with kernel size 3, and max pooling applied with pool size 2. Rectified linear unit (ReLU) is a widely used activation function in neural networks, known for its simplicity and effectiveness in deep learning models. ReLU has played a pivotal role in overcoming the vanishing gradient problem and accelerating the training process of deep neural networks.

In addition to the above blocks, CNN models often incorporate other layers like dropout, batch normalization, and activation functions (e.g., ReLU) for regularization and nonlinearity. The specific architecture and hyperparameters (e.g., number of filters, filter sizes, and pooling window sizes) within each block depend on the task, dataset, and design choices made by the model developer. Manually annotated training points for different regions/classes present in the area. Regions are: forest: represents dense variety of trees; plantations: crops and other type of planted trees; Croplands: land on which crops are to be grown, green in texture; fallow lands: land left without sowing for better crop cultivation, brown/muddy in texture; built-up-area: settlements, buildings, roads, and others; water bodies: sources of water. Labeling done by considering time-bound Sentinel-2 data for better understanding regions.

Classification is achieved by manually looking into the study area and making out different classes of regions. By multiple zooms in and out, confirm that the pixels are indeed of the class. When taking the historical data/previous data, load the respective satellite image with the dates about one year (i.e., for getting 2020 data, take 01-01-2020 to 31-12-2020 data), average those values for each pixel and map them out in the Google Earth engine, and this way the historical/previous year data is available for manually labeling; we have taken a total of 300 training points per classification for the study area, which is then run through the models for classification process. After this step, we apply classification with this trained model across the whole study area, giving us the finally classified/Land Use Land Cover map.

3 Result and Discussion

The wealth of information gathered from satellite images supports evidence-based decision-making and policy formulation. Governments, environmental organizations, and urban planners leverage land cover data to assess the impact of development projects, design effective land use strategies, and implement conservation measures [17]. In this proposed work, the study area taken as the Caatinga Forest Region of the state Rio Grande Do Norte, Brazil, for the year 2020, is shown in Fig. 3. To measure the performance of the proposed algorithms, calculated the validation accuracy and Kappa coefficient.

Manual label is done in Google Earth Engine; each class is marked by a total of 300 points across the state. Bands and data used other than specified are ALOS DSM: Global 30 m; slope and elevation are extracted from this dataset; this is commonly added to all three datasets. The final results of the model are obtained by fine tuning of the hyperparameters. Classification map generated by the developed CNN models with different satellite images for visual comparison is illustrated in Fig. 4.

The training dataset is made for the purpose of training the model and then to classify the whole of the study area using model. Testing dataset is made for the sole purpose of finding the accuracy of the model with the given accuracy metrics. Training points to testing points are: 80% training and 20% testing. Metrics used



Fig. 3 Study area (Caatinga Forest region, Brazil)

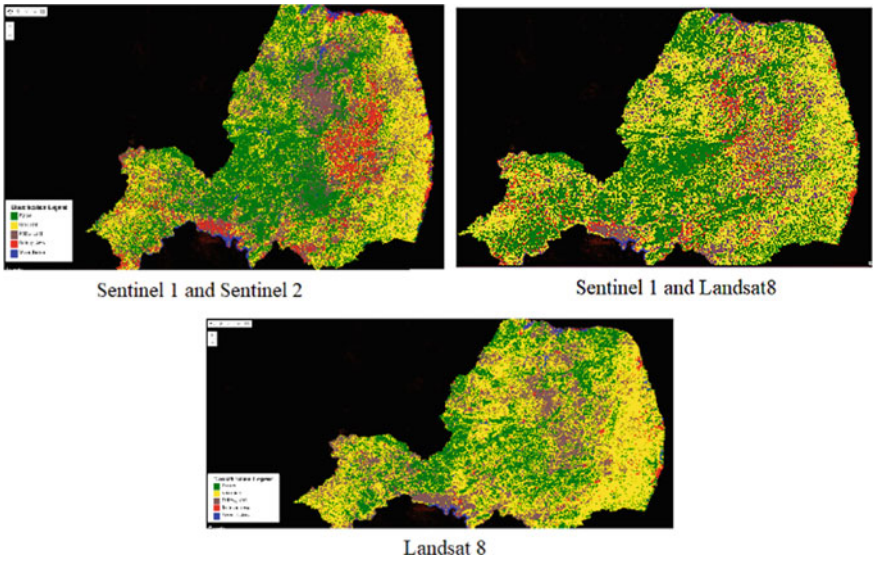


Fig. 4 Classification maps generated by proposed model with different satellite image datasets

for training the model are: optimizer used Adam; learning rate = 0.005; loss function: categorical cross entropy. Resultant metrics are accuracy, Kappa coefficient, producer's accuracy, and consumer's accuracy. Software used in the whole process is the Google Earth engine for dataset collection, manual labeling, and displaying the final classification of the study areal, and Jupyter notebook was used for training the model.

To test the developed CNN model, training and validation accuracy were simulated and observed that the developed model gives high accuracy of 97.07%, and Kappa coefficient of 96.33 is obtained with the combined S1 and S2 datasets in classifying the land cover. With the combined S1 and L8, accuracy of 94.44% and Kappa coefficient 93.04 are obtained. The detail results including producer's accuracy and consumer's accuracy obtained for the proposed model is given in Table 1.

Table 1 Details of results obtained from different datasets

Satellite imagery	Spatial resolution (m)	Year	No. of training points per class	Test accuracy	Producer's accuracy	Consumer's accuracy	Kappa coefficient
Sentinel1 SAR GRD + Sentinel-2 MSI	10	2020	300	97.07	96.49	100	96.33
Landsat 8 SR tier 1	30	2020	300	91.8	95.8	92.48	89.74
Landsat 8 SR tier 1 + Sentinel1 SAR GRD	30	2020	300	94.44	96.15	92.2	93.04

Consumer's Accuracy: Consumer's accuracy, also known as user's accuracy or user's reliability, is an evaluation metric used in classification tasks, particularly in the context of remote sensing or land cover mapping. It measures the probability that a pixel or sample, which has been classified as a specific class, indeed belongs to that class in reality. In other words, consumer's accuracy reflects the reliability of the model's predictions from the perspective of the end-user or consumer of the classification results. It helps determine the trustworthiness and usefulness of the model's outputs for decision-making and practical applications.

Producer's Accuracy: Producer's accuracy, also referred to as producer's reliability, is another evaluation metric commonly used in classification tasks, especially in remote sensing applications. It quantifies the probability that a pixel or sample, which truly belongs to a specific class, is correctly identified and classified as that class by the model. Producer's accuracy provides insights into the model's ability to accurately detect and label specific classes or land cover types. It helps assess the reliability and precision of the model's classifications from the perspective of the data producer or the one responsible for generating the ground truth data [18, 19].

Test accuracy measures the overall characterization of a model on unseen test data, consumer's accuracy indicates the reliability of the model's predictions from the user's perspective, and producer's accuracy assesses the accuracy and precision of the model's classifications from the data producer's viewpoint. Together, these metrics provide a comprehensive evaluation of the model's effectiveness, usability, and reliability in classification tasks, particularly in the field of remote sensing and land cover mapping [5].

Figure 5 shows the teaching and verification accuracy of the developed NN architecture for categorizing the land cover. The increase in training and validation accuracy over multiple epochs gives strong evidence that the proposed model has effectively learned the patterns and features necessary for accurate classification of land cover.

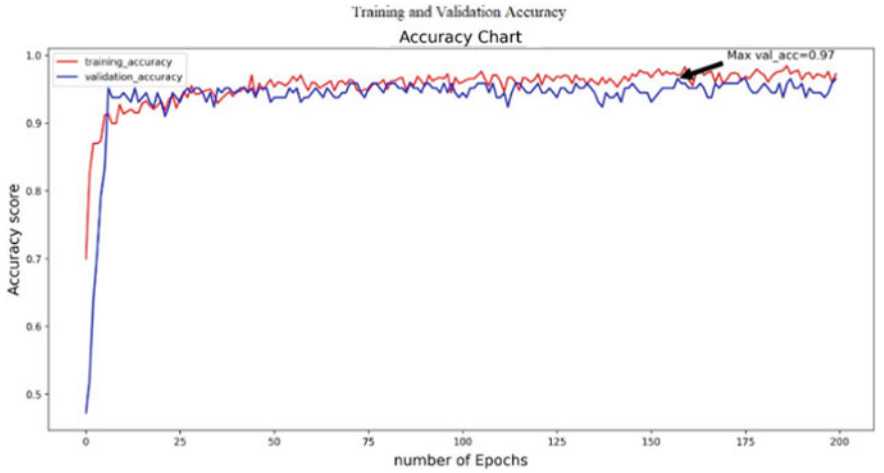


Fig. 5 Training and validation accuracy

Validation Accuracy

The validation accuracy is a measure of how well a machine learning model is performing on a validation set. The formula for calculating validation accuracy is as follows:

$$\text{Validation Accuracy} = \frac{(\text{Number of Correct Predictions on Validation Set})}{(\text{Total Number of Predictions on Validation Set})} \quad (1)$$

Validation accuracy is always expressed as percentage.

4 Conclusion

Satellite images provide us with a unique perspective on Earth's land cover, unlocking a wealth of knowledge about our changing planet. From the obtained results, we can say that a single DL model designed to handle different satellite images can generalize well across various datasets. It can learn common patterns and features from different sources, making it versatile and adaptable to different satellite sensors or data acquisition platforms. This reduces the need to develop and maintain multiple specialized models for each satellite source. Building a single DL model eliminates the need for redundant model development and deployment efforts. It saves computational resources, memory, and storage space that would otherwise be required for maintaining multiple models for different satellite images. As we continue to explore the frontiers of technology, satellite imagery will undoubtedly remain an invaluable tool in unraveling the secrets of Earth's land cover tapestry.

References

1. Ienco D, Interdonato R, Gaetano R, Minh DHT (2019) Combining Sentinel-1 and Sentinel-2 satellite image time series for land cover mapping via a multi-source deep learning architecture ISPRS J Photogramm Rem Sens 158:11–22
2. Talukdar S, Singha P, Mahato S, Shahfahad, Pal S (2021) Land-use land-cover classification by machine learning classifiers for satellite observations—a review. Rem Sens 12:1135. https://doi.org/10.3390/rs12071135
3. Shafique A, Cao G, Khan Z, Asad M, Aslam M (2022) Deep learning based change detection in remote sensing 48 images: a review. Rem Sens 14:871
4. Helber P, Bischke B, Dengel A, Borth D (2017) Eurosat: a novel dataset and deep learning benchmark for land use and land cover classification. arXiv preprint arxiv:1709.00029
5. Leitloff J, Riese FM (2018) Examples for CNN training and classification on Sentinel-2 data. Zenodo, Geneve, Switzerland
6. Balaniuk R, Isupova O, Reece T (2020) Article mining and tailings dam detection in satellite imagery using deep learning. Sensors 20:6936. https://doi.org/10.3390/s20236936
7. Xu X, Li W, Ran Q, Du Q, Gao L, Zhang B (2017) Multisource remote sensing data classification based on convolutional neural network. IEEE Trans Geosci Rem Sens 56:937–949
8. Lei T, Zhang Y, Lv Z, Li S, Liu S, Nandi AK (2019) Landslide inventory mapping from bitemporal images using deep convolutional neural networks. IEEE Geosci Rem Sens Lett 2019(16):982–986
9. Bischke B, Helber P, Folz J, Borth D, Dengel A (2019) Multi-task learning for segmentation of building footprints with deep neural networks. In: Proceedings of the 2019 IEEE international conference on image processing (ICIP), Taipei, Taiwan, 22–25 Sept 2019, pp 1480–1484
10. Dong S, Zhuang Y, Yang Z, Pang L, Chen H, Long T (2020) Land cover classification from VHR optical remote sensing images by feature ensemble deep learning network. IEEE Geosci Rem Sens Lett 17(8)
11. Li H, Gong M, Zhang M, Wu Y (2021) Spatially self-paced convolutional networks for change detection in heterogeneous images. IEEE J Sel Top Appl Earth Obs Rem Sens 14:4966–4979
12. Yu Q (2014) Object-based detailed vegetation classification with airborne high spatial resolution remote sensing imagery. Photogramm Eng Rem Sens 72(7):799–811
13. Chughtai AH, Abbasi H, Karas IR (2021) A review on change detection method and accuracy assessment for land use land cover. Rem Sens Appl Soc Environ 22:100482
14. Maladkar K (2018) Overview of convolutional neural network in image classification. CNN Reference Jan 25, 2018. [Online]. Available https://www.analyticsindiamag.com/convolutionalneural-network-image-classification-overview/. Accessed 10th Feb 2019
15. Kerner HR, Wagstaff KL, Bue BD, Gray PC, Bell JF, Amor HB (2019) Toward generalized change detection on planetary surfaces with convolutional autoencoders and transfer learning. IEEE J Sel Top Appl Earth Obs Rem Sens 12:3900–3918
16. Sarker IH, Kayes ASM, Watters P (2019) Effectiveness analysis of machine learning classification models for predicting personalized context-aware smartphone usage. J Big Data 6(1):1–28
17. Deng L, Hinton G, Kingsbury B (2013) New types of deep neural network learning for speech recognition 2013, and related applications: an overview. In: Proceedings of IEEE international conference on acoustics, speech, and signal processing (ICASSP), pp 8599–8603
18. Mu L, Wang L, Wang Y, Chen X, Han W (2019) Urban land use and land cover change prediction via self-adaptive cellular based deep learning with multi sourced data. IEEE J Appl Earth Observ Rem Sens
19. Ma L, Liu Y, Zhang X, Ye Y, Yin G, Johnson BA (2019) Deep learning in remote sensing applications: a meta-analysis and review. ISPRS J Photogramm Rem Sens 152:166–177

Multi-class Plant Leaf Disease Classification on Real-Time Images Using YOLO V7



P. Sajitha, Diana A Andrushia, and S. S. Suni

Abstract Agriculture faces a significant challenge with leaf disease, which can lead to substantial decreases in crop productivity. Deep learning techniques, particularly the YOLO (You Only Look Once) algorithm, have gained popularity for effectively detecting and categorizing plants diseases in real time. This research presents a leaf disease detection and classification system utilizing YOLO v7, which benefits from a diverse and current dataset obtained from real-time field images. This dataset allows for precise training and evaluation of plant disease detection models in real-world scenarios, offering valuable insights into the effectiveness of the YOLO v7 algorithm. By training on a large collection of images of plants, the system achieves high accuracy (96%) in identifying various leaf diseases such as leaf spot, rust, and early blight. Utilizing this technology, farmers will be able to take preventative measures and decrease crop losses through early disease diagnosis and intervention due to the quick and effective object identification capabilities of YOLO v7.

Keywords Leaf disease detection · YOLO (You Only Look Once) · Deep learning

1 Introduction

Agriculture is incredibly important to a nation's economy and is essential to its entire growth. It supports human civilization by giving billions of people throughout the world food, clothes, and a place to live. Agriculture has always been essential to the development of societies because it will be helpful for people to settle down and

P. Sajitha (✉) · D. A. Andrushia
Karunya Institute of Technology and Sciences, Coimbatore, India
e-mail: sajithap@karunya.edu.in

D. A. Andrushia
e-mail: diana@karunya.edu

S. S. Suni
Ilahia College of Engineering and Technology, Ernakulam, India
e-mail: drsuniss@icect.ac.in

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2023

S. Shakya et al. (eds.), *Fourth International Conference on Image Processing and Capsule Networks*, Lecture Notes in Networks and Systems 798,
https://doi.org/10.1007/978-981-99-7093-3_32

create an ecofriendly environment. It has evolved from a basic means of sustenance to a sophisticated system that uses innovation, science, and technology [1].

The agriculture field has a number of challenges, including food insecurity, climate change, and an overreliance on synthetic insecticides. The biggest issue facing the farming sector is the spread of plant diseases, which have a big effect on crop production. If they are not recognized and treated, plant diseases that can be brought on by microorganisms, fungal infections, or additional infectious agents can spread swiftly. For efficient disease management and control, early identification and detection of leaf diseases are essential. Visual examination and manual diagnosis methods have traditionally been used by farmers and agronomists to diagnose plant diseases, but these methods can be labor-intensive and arbitrary [2, 3]. Automation of plant disease diagnosis has become more feasible because of developments in computer vision algorithms. Deep learning algorithms have shown significant achievements in the area of image-based plant disease identification, offering improved efficiency and accuracy for identifying variety of diseases in plants [4].

The YOLO algorithms play a critical role in plant leaf disease detection by efficiently and accurately identifying diseased regions on leaves. YOLO, in contrast to conventional approaches, performs bounding box regression and object detection in a single neural network run [5]. Through its real-time object detection capabilities, YOLO divides the input image into a grid, predicting bounding boxes and corresponding class probabilities for each grid cell in a single pass through the deep learning neural network. Trained on extensive datasets of healthy and diseased plant leaves, YOLO learns to recognize specific patterns associated with different diseases, enabling it to swiftly and precisely distinguish between healthy and infected areas. This speed and accuracy empower farmers and researchers to promptly identify and address leaf diseases, leading to improved crop management, reduced losses, and enhanced agricultural productivity. YOLO's ability to generalize and adapt to diverse plant species and environmental conditions further enhances its effectiveness in safeguarding plant health. In this study, a novel method for identifying leaf diseases is presented using the YOLO v7 algorithm. Using this YOLO v7 algorithm, we trained large collection of images and identified different diseases with high accuracy.

This research work has been structured as follows: The existing approach surveys are covered in Sect. 2, and fundamentals of the YOLO algorithm are covered in Sect. 3. The proposed methodology is discussed in Sect. 4. The experimental results and analysis are presented in Sect. 5. Finally, Sect. 6 marks the end of this study.

2 Literature Survey

This section provides a review on different versions of the YOLO algorithm (ranging from YOLO version 1 to version 5) which have been utilized for disease detection in plants.

The study by Morbekar and Parihar [6] used YOLO (You Only Look Once) as a real-time image recognition system to recognize different crop diseases. Their

primary goal was to develop a reliable method of early disease identification that would aid in the successful management of agricultural diseases. The approach's drawback, however, is that it now only focuses on detecting diseases in entire crops, ignoring the possibility of recognizing diseases in certain crop sections or at various phases of disease development.

Srinivasan et al. [7] used YOLO v2 algorithm for detection of rice crop diseases. It employs the Darknet-53 network with 53 convolutional layers, utilizing K-means clustering to create nine anchor boxes. The 75 layered design and the use of skip connections allow for detection at various scales while reducing localization errors and maintaining a high level of accuracy with low complexity. Its increased depth and complexity may demand more computational resources and longer training time.

Shill and Rahman [8] showed that their computer vision method, which included YOLO V3 and YOLO V4 pretrained on the MSCOCO dataset, successfully addressed the difficult task of manual disease detection in thirteen plant species, encompassing seventeen disease classes from the PlantDoc dataset, yielding promising and accurate results for disease detection and classification. The MSCOCO dataset is heavily relied upon pretraining in the proposed computer vision approach using YOLO V3 and YOLO V4 for manual disease detection, which may lead to decreased performance and accuracy for certain plant species and disease classes.

Deep learning was applied in a study by Theresa and Midhun [9] to identify diseases in bell pepper plants. This work used the promising object identification model YOLO V5, which is well-known for its quickness, accuracy, and simplicity of training. They quickly detected diseases using YOLO V5, even minuscule spots on leaves, and provided bounding boxes and class probabilities. Their study's key goal was to make it easier for farmers to quickly treat diseases by identifying bacterial areas at an early stage.

In their research, Chen et al. [10] developed a disease recognition miniature on YOLO v5 in rubber trees. It detected two diseases, such as powdery mildew and anthracnose, with excellent accuracy (above 85%). Accuracy is increased by 5.4% from the initial YOLOv5 network to 70%. As a result, the augmented YOLOv5 network showed better accuracy in recognizing and categorizing rubber tree diseases, providing insightful information for disease prevention and control. Despite these successes, the enhanced YOLO v5 network's detection accuracy still has space for improvement.

Janibul et al. [11] proposed an artificial intelligence-based solution for leaf disease detection in tea plants. This research focused on using YOLO v7, the quickest single-stage method for object detection, to address the decreasing tea production caused by climate change and pest infestation. The study utilized a dataset of 4000 images, comprising five types of diseases. The detection and identification performance of YOLO V7 were evaluated using statistical metrics such as detection accuracy, precision, recall, MAP value, and F1 score. One drawback is that the suggested method presently does not take into account additional data sources, including as temperature, humidity, soil details, and information on pathogenic spores, which might improve the precision of disease identification and provide early warnings.

For the purpose of identifying diseases in plants, the YOLO algorithm is a well-liked computer vision method. In order to function, the image is divided into a grid, and bounding boxes and class probabilities are predicted for each grid cell. By training the system on sizable datasets of both healthy and diseased plant images. From the review, YOLO models (1–6) have always focused on real-time object detection by trading off some accuracy for faster inference times. YOLO v7 might introduce further architectural reforms to improve the accuracy and detection speed.

The goal of this research is to use the YOLOv7 object detection framework to create a multi-class leaf disease diagnosis system for plants. A helpful tool for early identification and efficient management of plant health, the system seeks to precisely identify and categorize several forms of leaf diseases in plants [12, 13].

3 Basics of YOLO Algorithms

This section discusses several fundamental YOLO algorithm principles. The YOLO algorithms are an emerging series of object recognition algorithms that is developed for real-time object detection and classification. It includes YOLO v1, v2, v3, v4, and v5, which have progressively increased accuracy and speed [14–17]. Table 1 highlights some key specifications of versions (1–5) of the YOLO algorithms.

Different YOLO algorithm models can accurately locate and classify various types of plant diseases, enabling farmers and researchers to swiftly identify and address issues, ultimately aiding in crop protection and management [18–20]. YOLO v6 and YOLO v7 are emerged recently. It not only excels in identifying targets but also demonstrates its usefulness in, for instance, segmentation and human posture assessment. YOLO v7 utilizes nine anchor structures, which are preconfigured boxes with various aspect ratios. This strategy enables YOLO v7 to encompass a broader range of object shapes and sizes while reducing false positives compared to earlier versions. YOLO v7, which can process images at a remarkable speed of 155 frames per second, offers a significant speed advantage [21, 22].

YOLO v7 introduces a novel attention mechanism called “Adaptive Contextual Fusion,” which has the ability to capture contextual information and improves object detection performance in complex scenes. With YOLO v7, the model adopts a multi-scale prediction strategy, enabling it to detect objects at various scales. This leads to better localization accuracy and improved handling of objects of different sizes. YOLO v7 utilizes a progressive training approach, where the model undergoes multiple stages of training. This process involves gradually increasing the input resolution and refining the detection performance, leading to improved accuracy and better localization of objects [23].

YOLO v7 Architecture

An innovative object identification approach that makes use of deep neural networks is the YOLOv7 architecture. It incorporates enhancements to attain greater accuracy and quicker inference speed, building on the success of earlier YOLO iterations.

Table 1 Key features of different YOLO algorithms

Specification	YOLO v1	YOLO v2	YOLO v3	YOLO v4	YOLO v5
Year of release	2015	2016	2018	2020	2020
Model architecture	Custom	Darknet-19	Darknet-53	CSPDarknet53	CSPDarknet53
Number of parameters	~ 45 million	~ 60 million	~ 62 million	~ 27.6 million	~ 46.5million
Training time	Several hours to days (Slow training due to custom architecture)	Hours to a day (Faster than YOLOv1 with Darknet-19)	Days to several days (Longer training due to deeper architecture)	Several days to a week (Deep architecture with advanced optimizations)	Few hours to a day (Faster training with streamlined architecture)
Anchor boxes	Manual selection	Predefined	Predefined	K-means clustering	K-means clustering
Training strategy	Batch gradient descent	SGD with momentum	Adam	Adam	Adam
Key features	Single neural network for object detection	Anchor boxes, pyramid feature networks, multi-scale feature extraction	Darknet-53 architecture, feature pyramid network, multi-scale feature	CSPDarknet53 backbone, optimization techniques	Backbone + neck + head structure, efficient detection of the D7 backbone

YOLOv7 is a well-liked option for real-time object recognition applications due to its simplified design and effective bounding box prediction [22]. Efficient layer aggregation network (ELAN) serves as the foundation for YOLO v7 architectural design. In order for deeper networks to converge and train efficiently, ELAN takes into account constructing an efficient network by managing the shortest and longest gradient paths. Figure 1 shows the architecture design of YOLO v7.

In the architecture, input channel is fed into the bottom block with 1×1 convolution, while three other connections to the block were obtained by convolving the input by 3×3 convolution with similar channel multiplier. In the lower block, features are concatenated with 1×1 convolution which is used to gain in-depth knowledge. YOLO v7 architecture is designed through scaling by increasing width of input channel and depth of convolution layers.

YOLO v7 modifies the ELAN architecture called E-ELAN. E-ELAN, which stands for extended efficient layer aggregation network, is the computing unit of the YOLO v7 backbone. With the help of the "expand, shuffle, merge cardinality" technique, the E-ELAN architecture of YOLOv7 makes it possible for the model to learn more effectively while maintaining the original gradient route. E-ELAN uses group convolution to expand the computing block's channels and cardinality.

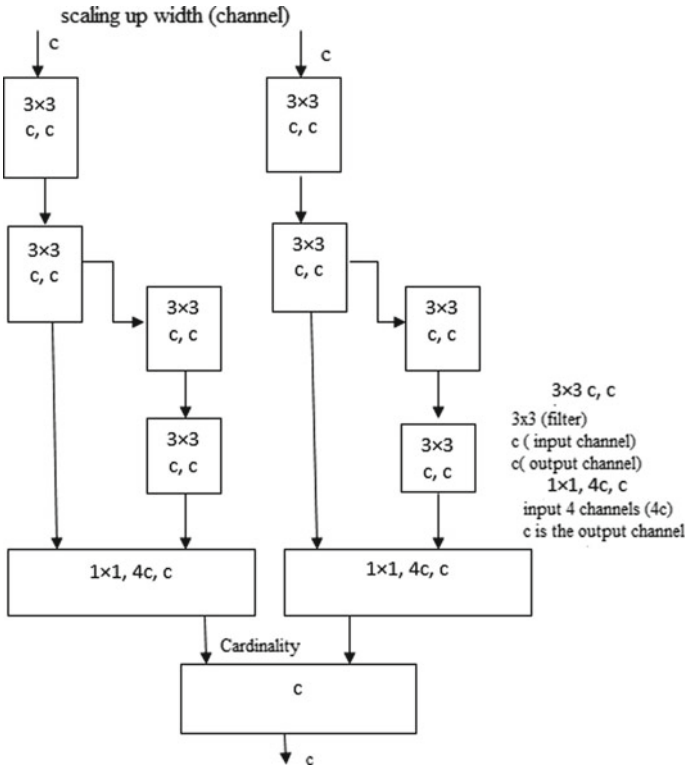


Fig. 1 Architecture design of YOLO v7

The same group parameter and channel multiplier are applied to each computational block in a computing layer. The feature maps of each computational block will be sorted into groups according to size and then concatenated. A shuffled group feature map will then be used to conduct merge cardinality.

Increasing or decreasing the number of layers in the model is referred to as scaling depth. In the same way, scaling the number of channels in the model architecture correlates with breadth. The model architecture files include scaling factors for depth and width. Because YOLOv7 architecture is concatenated with the other layers, we must determine how the output kernels will vary as we scale the depth parameter of a computational block. Then, taking into account scaling width by the computed change in kernels, therefore, this compound scaling strategy will keep the characteristics of the architecture's initial design for YOLO v7.

4 Proposed Method

The complete block diagram of proposed methodology is shown in Fig. 2.

- **Image acquisition and dataset building**

A collection of images featuring diverse plants and their associated diseases was compiled from real-world events to train and evaluate an algorithm. The dataset, named "Plantleaf," encompasses multiple diseases such as rust, leaf spots, and early leaf blight.

- **RoboFlow Tool**

RoboFlow is a flow-based visual programming language that enables the creation of general-purpose mobile manipulation tasks. The purpose of RoboFlow is to (i) provide a reliable low-level implementation of program operations on a mobile manipulator and (ii) limit the high-level programming as much as possible to prevent user mistakes while permitting expressive programs that incorporate branching, looping, and nesting [24]. The dataset was meticulously labeled and subsequently annotated using the RoboFlow tool. Bounding boxes were added around the affected regions in each image.

An important stage in classifying and labeling various sorts of data, such as images, videos, or text files, is image annotation. Its goal is to make it possible for machines to comprehend and analyze input data. For supervised machine learning (ML), an annotated dataset is essential since it acts as the starting point for the model training process, which interprets the data and produces reliable results. Here, we systematically annotated leaf images using RoboFlow by highlighting and labelling key areas connected to several leaf diseases. These annotations entailed precisely identifying lesions, disease spots, and impacted areas and supplying ground truth labels that aided the YOLO v7 model's learning process. Accuracy and consistency were guaranteed with the use of RoboFlow's annotation tools, enabling the model to accurately identify and categorize leaf diseases during inference. Figure 3 displays the dataset's specifics.

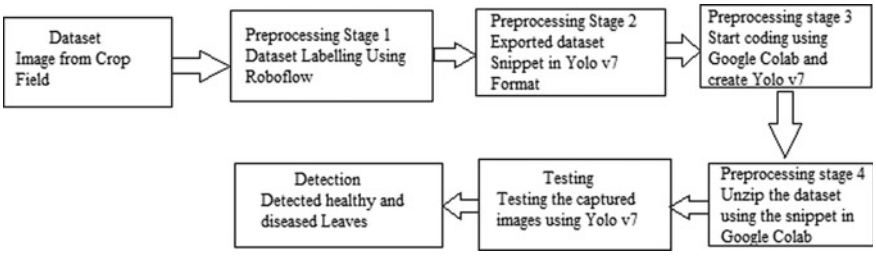


Fig. 2 Architecture of proposed leaf disease detection using YOLO v7

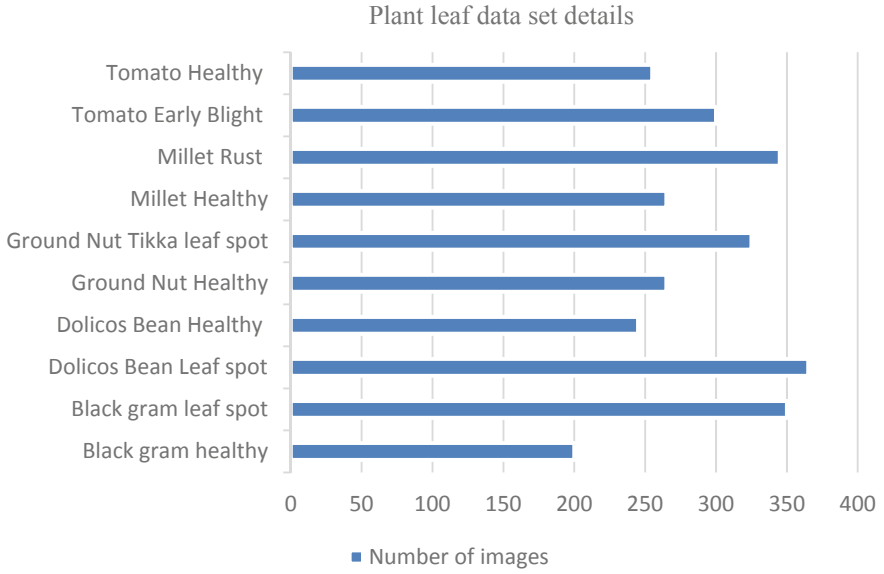


Fig. 3 Plant leaf dataset details

- **Training Procedure**

  The following procedures were used to train a customized YOLO v7 model:

  Step 1: Environment setup: Download the YOLO v7 library and install the required plugins to configure the YOLO environment.
  Step 2: Google Collab training: Using the free Google Collab training environment, possibly based on the Tesla P100 GPU.
  Step 3: Data collection formatting: Real-time images were collected to create a new data set called "Plantleaf". RoboFlow was used to load datasets for annotation.
  Step 4: Preprocessing and scaling: To optimize the dataset, RoboFlow was used to apply preprocessing levels and select appropriate scaling approaches.
  Step 5: Data partitioning: The dataset was automatically partitioned into training, testing, and validation sets by RoboFlow.
  Step 6: Fragment: The proposed switch or package uses RoboFlow PIP, as shown in Fig. 4.

  To train a custom YOLO v7 model, it is recommended to create a data.yaml file that contains details of the input dataset. The above steps offer a broad explanation of the training process for a customized YOLO v7 model.

  Achieving high accuracy and improving the performance of YOLO v7 relies on several key strategies. Firstly, carefully curating a diverse and large-scale dataset with

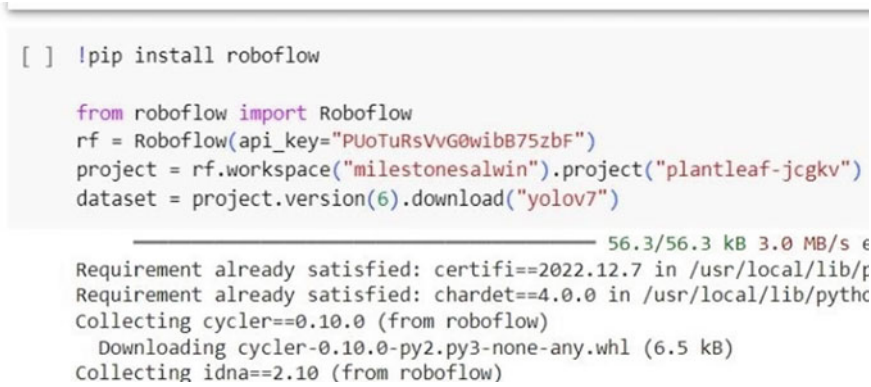


Fig. 4 RoboFlow key

Table 2 Hyperparameters

S. No	Hyperparameters	Value
1	Learning rate	0.001
2	Momentum	0.938
3	Epoch	250
4	Weight decay	0.0005
5	Optimizer	Adam
6	Batch size	16
7	Activation	Mish

accurate annotations enables the model to learn robust representations. Secondly, optimizing the model architecture and hyperparameters through techniques like grid search or automated tuning enhances its detection capabilities. Lastly, utilizing hardware accelerators, such as GPUs, during training and inference can significantly speed up the process and lead to improved accuracy and efficiency. This model has 37 million parameters. Some hyperparameters used in this experiment is given in Table 2.

5 Results and Discussion

This section describes and explains the implementation results as well as the performance of our proposed system. In addition, comparative results from the existing work are presented.

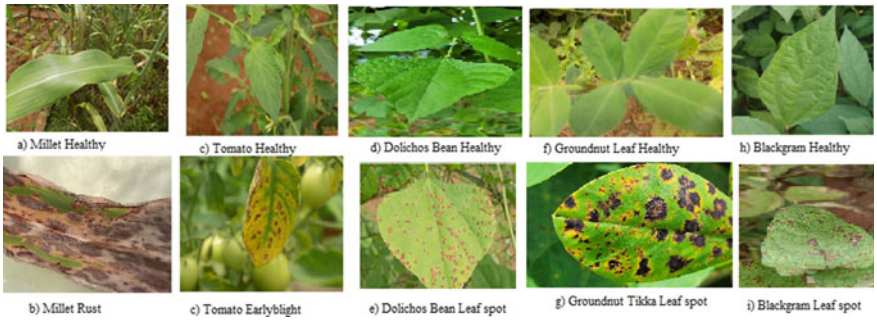


Fig. 5 Sample images from dataset

5.1 Dataset Description

A dataset called Plantleaf was built with the help of plant pathologists of Agricultural College, Vellayani, Trivandrum that contains images of various plants leaves such as tomato, dolichos beans, millet, groundnut, black gram, along with the diseases that correspond to them. The diseases include tomato early blight, millet rust, black gram leaf spot, groundnut tikka leaf spot, and dolichos bean leaf spot. Figure 5 displays a few of the sample images.

To ensure diversity and complexity within the dataset, images of leafs were captured using sophisticated cameras such as Nikon CoolPix 340 and two i-phones 13pro and Samsung S20 under varying lighting conditions and from different angles. Size of all images is 416×416. The final dataset comprises 2500 images, with 2000 images allocated for training, and 500 images designated for testing phase.

In this study, Google Colab was used to train YOLOv7 networks using Plantleaf datasets.

Each training period lasts for 7 h, 25 min, and 40 s, and batch size is 16.

Specifically, this research explores the GPU utilization feature in Google Colab which provides "NVIDIA Tesla K80" with 12 GB RAM.

5.2 Performance Metrics

Performance indices are required for an experiment to assess an algorithm model. This research evaluates average accuracy, recall rate, F1 score, and precision in accordance with the evaluation indices of the neural network model. Equations (1) through (3) display the results of the computations for precision, recall, and F1 score. Figure 6 indicates the performance metrics of the proposed work.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{1}$$

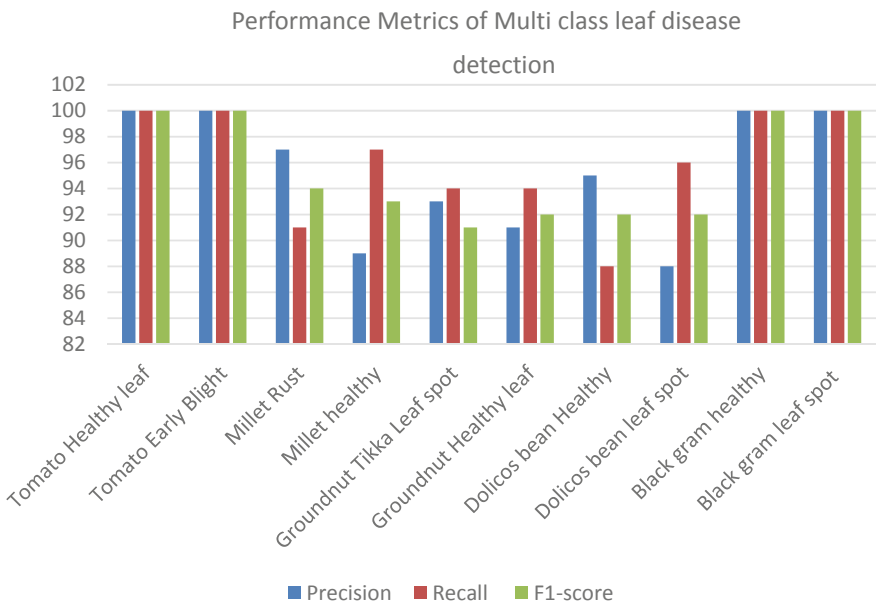


Fig. 6 Performance metrics of multi-class leaf disease detection and classification

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{2}$$

$$\text{F1 score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3}$$

A confusion matrix contrasts anticipated disease classes with actual disease classes to produce the system's performance. The system's strengths and drawbacks may be thoroughly examined, as well as true positives, true negatives, false positives, and false negatives. Figure 7 illustrates the confusion matrix of this model.

The effectiveness of the generated model is displayed in graphs that display various measures of the training and validation sets' performance. Figure 8 shows three different forms of loss: box loss, objectness loss, and classification loss. The box loss calculation allows the algorithm to determine an exact location of an object's center and calculate its bounding box. "Objectness" measures how probable it is that an object will be located in a specific location. An item is likely to be present in the viewable portion of an image, according to high objectivity. Classification loss is a measure of how accurately an algorithm can identify an object's correct class. The model's parameters change value throughout the span of 0–200 iterations and consistently improve as the number of iterations increases in the range from 200 to 250. The loss of objectness is minimal.

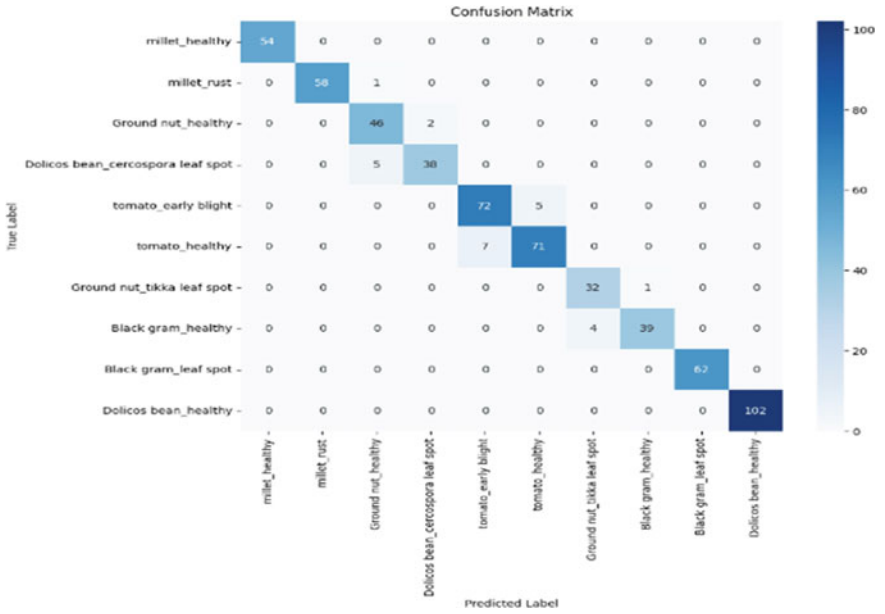


Fig. 7 Confusion matrix of multi-class leaf disease detection and classification

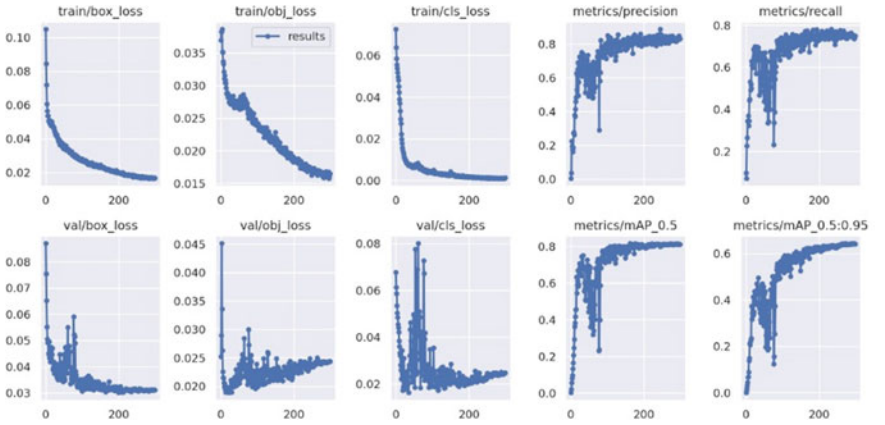


Fig. 8 Evaluation indicators visual analysis

Figure 9 shows sample images after image detection, and Fig. 10 shows a comparative analysis of YOLO v7, YOLO v5, and YOLO v3. The work findings demonstrate the remarkable efficiency of YOLO v7 for detecting leaf diseases, achieving an impressive accuracy rate of 96%. This highlights the strong performance and reliability of the YOLO v7 algorithm in accurately identifying and classifying diseases affecting leaves.

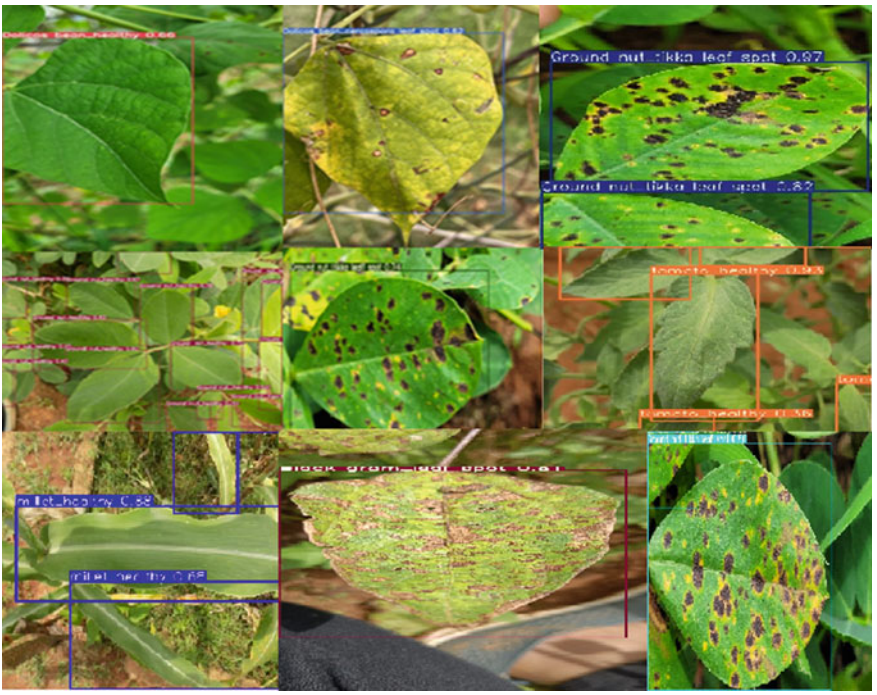


Fig. 9 YOLO v7 based disease detection for different plant leaf diseases

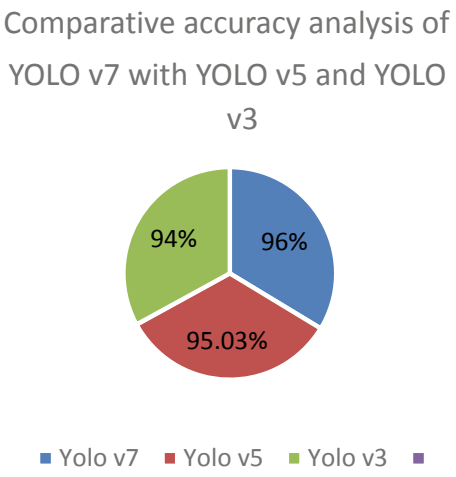


Fig. 10 Comparative accuracy analysis

The experimental comparison of YOLO v7 with YOLO v5 and YOLO v3 reveals that YOLO v7 exhibits improved object detection and classification capabilities, resulting in higher precision and recall rates for detecting plant leaf diseases. Moreover, YOLO v7 demonstrates superior efficiency, with optimizations enhancing inference speed and computational efficiency, enabling real-time or near real-time disease detection. The findings underscore the significance of using the latest version of the YOLO algorithm, as YOLO v7 represents a substantial advancement in accuracy and efficiency compared to its predecessors. Overall, the experimental work confirms that YOLO v7 outperforms YOLO v5 and YOLO v3, emphasizing the importance of leveraging state-of-the-art object detection algorithms for more effective and efficient plant leaf disease detection systems.

6 Conclusion

Deep learning-based plant disease detection is an efficient tool for early diagnosis and disease detection in crops. Among these methods, YOLO v7 algorithm is viable option for precisely identifying and localizing leaf diseases in plant images. By leveraging its deep learning architecture, this model demonstrates exceptional proficiency in identifying various disease types with high precision and recall. Notably, this research highlights the significant improvements in accuracy rate 96% made by YOLO v7 when compared to previous YOLO algorithms. It efficiently utilizes the available data to learn representative features and patterns relevant to plant diseases. It can generalize well from the limited dataset and adapt to variations in leaf diseases. As future, it is essential to explore and develop correction methods to further enhance the performance of the YOLO v7 algorithm in plant disease detection.

References

1. Arjun KM (2013) Indian agriculture-status, importance and role in Indian economy. Int J Agric Food Sci Technol 4(4):343–346. ISSN 2249-3050
2. Patel S, Jaliya UK, Patel P (2020) A survey on plant leaf disease detection. Int J Modern Sci Technol 06(04). ISSN 2455-3778
3. Wu Y, Lihong, Goodman ED Tomato leaf disease identification and detection based on deep convolutional neural network. Intell Autom Soft Comput. https://doi.org/10.32604/iasc.2021.016415
4. Zhang Q, Liu Y, Gong C, Chen Y, Huihui Y (2020) Applications of deep learning for dense scenes analysis in agriculture: a review. Sensors. https://doi.org/10.3990/s20051520
5. Bochkovskiy A, Liao H-YM, Wang C-Y YOLO v7: trainable bag-of-freebies sets new state-of-the-art for real-time object detectors
6. Morbekar A, Parihar A, Jadhav R (2020, June) Crop disease detection using YOLO. In: 2020 International conference for emerging technology (INCET). IEEE, pp 1–5
7. Srinivasan R, Prakash Reddy LB, Badashah SJ (2022) Detection of rice crop disease using the YOLO algorithm. Int Res J Modernization Eng Technol Sci 04(03)

8. Shill A, Rahman MA (2021, July) Plant disease detection based on YOLO v3 and YOLO v4. In: 2021 International conference on automation, control and mechatronics for industry 4.0 (ACMI). IEEE, pp 1–6
9. Mathew MP, Mahesh TY (2022) Leaf-based disease detection in bell pepper plant using YOLO v5. Sig Image Video Process 1–7
10. Chen Z, Wu R, Lin Y, Li C, Chen S, Yuan Z, Chen S, Zou X (2022) Plant disease recognition model based on improved YOLO v5. Agronomy 12(2):365. https://doi.org/10.3390/agronomy12020365
11. Soeb MJA, Jubayer MF, Tarin TA, Al Mamun MR, Ruhad FM, Parven A, Meftaul IM (2023) Tea leaf disease detection and identification based on YOLO v7 (YOLO-T). Sci Rep 13(1):6078
12. Mia J, Bijoy HI, Uddin S, Raza DM (2021, July) Real-time herb leaves localization and classification using YOLO. In: 2021 12th International conference on computing communication and networking technologies (ICCCNT). IEEE, pp 1–7
13. Ponnusamy V, Coumaran A, Shunmugam AS, Rajaram K, Senthilvelavan S (2020, July) Smart glass: real-time leaf disease detection using YOLO transfer learning. In: 2020 International conference on communication and signal processing (ICCSP). IEEE, pp 1150–1154
14. Liu J, Wang X (2020) Tomato diseases and pests detection based on improved YOLO V3 convolutional neural network. Front Plant Sci 11:898
15. Mathew MP, Mahesh TY (2021, June) Determining the region of apple leaf affected by disease using YOLO V3. In: 2021 International conference on communication, control and information sciences (ICCISc), vol 1. IEEE, pp 1–4
16. Xue Z, Xu R, Bai D, Lin H (2023) YOLO-tea: a tea disease detection model improved by YOLO v5. Forests 14(2):415
17. Jhatial MJ, Shaikh RA, Shaikh NA, Rajper S, Arain RH, Chandio GH, Bhangwar AQ, Shaikh H, Shaikh KH (2022) Deep learning-based rice leaf diseases detection using YOLO v5. Sukkur IBA J Comput Math Sci 6(1):49–61
18. Li J, Zhu X, Jia R, Liu B, Yu C (2022, June) Apple-YOLO: a novel mobile terminal detector based on YOLO v5 for early apple leaf diseases. In: 2022 IEEE 46th annual computers, software, and applications conference (COMPSAC). IEEE, pp 352–361
19. Haque ME, Rahman A, Junaeid I, Hoque SU, Paul M (2022) Rice leaf disease classification and detection using YOLO v5. arXiv preprint arXiv:2209.01579
20. Cao L, Zheng X, Fang L (2023) The semantic segmentation of standing tree images based on the YOLO V7 deep learning algorithm. Electronics 12(4):929
21. Gillani I, Munawar M, Talha M, Azhar S, Mashkoor Y, Uddin M, Zafar U (2022) YOLO v5, YOLO-x, YOLO-r, YOLO v7 performance comparison: a survey, pp 17–28. https://doi.org/10.5121/csit.2022.121602
22. Zhou S, Cai K, Feng Y, Tang X, Pang H, He J, Shi X (2023) An accurate detection model of Takifugu rubripes using an improved YOLO v7 network. J Mar Sci Eng 11(5). https://doi.org/10.3390/jmse11051051
23. Wang CY, Bochkovskiy A, MarkLiao H (2022) YOLO v7: trainable bag of feebies sets new state of the art for real time object detectors. Comput Vis Pattern Recogn. https://doi.org/10.48550/arXiv.2207.02696
24. Floriana C, Zuppichini F, Paul G, Mark M, Jacob S (2022) Roboflow 100: a rich, multi domain object detection benchmark. Comput Vis Pattern Recogn. https://doi.org/10.48550/arXiv.2211.13523

City Road Anomaly Alert for Autonomous Vehicles: Pothole Dimension Estimation with YOLOv5

Varun Vinod Kulkarni, S. Vishal, Monish Mohanty, and Peeta Basa Pati

Abstract The problem addressed in this chapter is the pervasive issue of road defects, specifically potholes, on Indian roads. These defects pose significant challenges for both human drivers and autonomous driving technology. For human drivers, potholes can lead to vehicle damage and even pose risks to human life due to accidents caused by sudden manoeuvers to avoid these defects. For autonomous driving technology, the detection and accurate classification of such road defects is a critical task. The solution proposed involves a deep learning model that performs three tasks: classification, pothole detection, and area and distance estimation. The model is trained to detect and classify road defects into three categories. A novel aspect of this work is the estimation of the area and distance of the detected potholes. By calculating the size of the bounding box surrounding the pothole that was spotted in the image, the pothole's area may be determined. The distance of the pothole is estimated from the bottom of the image, assuming a certain scale between the pixels and actual distance. The results obtained surpass those reported in existing literature, indicating the effectiveness of the proposed methodology.

Keywords Pothole · Detection · Visual geometry group 16 (VGG16) · Classification · You only look once v5 (YOLOv5) · Distance and area estimation

V. V. Kulkarni · S. Vishal · M. Mohanty · P. B. Pati (✉)
Department of Computer Science and Engineering, Amrita School of Computing Bengaluru, Amrita Vishwa Vidyapeetham, Bengaluru, India
e-mail: bp_peeta@blr.amrita.edu

V. V. Kulkarni
e-mail: BL.EN.U4AIE20033@bl.students.amrita.edu

M. Mohanty
e-mail: BL.EN.U4AIE20074@bl.students.amrita.edu

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2023

S. Shakya et al. (eds.), *Fourth International Conference on Image Processing and Capsule Networks*, Lecture Notes in Networks and Systems 798,
https://doi.org/10.1007/978-981-99-7093-3_33

1 Introduction

Potholes detection remains a major challenge in Indian roads. These road defects vary in size, shape, and appearance, causing a great discomfort during travel. Sometime, potholes may damage the vehicle and increase the maintenance costs for vehicle owners. Potholes pose serious risks to human life. For human drivers, unexpected potholes detection can lead to accidents.

In the context of autonomous driving, the detection and accurate classification of such road defects remains as a critical task. Autonomous vehicles must be capable of identifying these defects in real-time to make appropriate decisions, such as slowing down or safely traveling around the defect. However, the complexity of this task is increased by the varying the nature of potholes and the different environmental conditions under which the detection system must operate. Various solutions have been proposed to address this issue. Many of these involve the use of machine learning techniques, utilizing the capabilities of these algorithms to make predictions or judgements based on data without being expressly programmed to do so.

This research study proposes a novel solution that leverages the power of deep learning in the detection and classification of road defects, specifically potholes. The proposed approach involves a model that performs three tasks: classification, pothole detection, along with area and distance estimation. Unlike many existing methods, the proposed model is designed to handle the diverse and unpredictable nature of road defects. It is trained on a diverse dataset of images, representing various road defects and environmental conditions.

The novelty of the proposed solution lies in the area and distance estimation of the detected potholes. This feature, not commonly found in existing methods, provides valuable information that can be used by autonomous driving systems or drivers to make informed decisions. For instance, knowing the size and distance of a pothole can help an autonomous vehicle decide whether it needs to slow down, change lanes, or take another appropriate action.

Our proposed solution aligns with the United Nations Sustainable Development Goals (UNSDGs). It contributes to promoting good health and well- being (Goal 3) by lowering the possibility of accidents brought on by potholes. It also supports the goal of building resilient infrastructure (Goal 9) by aiding in the identification and subsequent repair of road defects.

The remainder of this chapter is organized as follows: we first discuss the data collection process, which involved capturing images under various real-world scenarios and different environmental conditions. This is followed by a detailed explanation of our methodology including how our deep learning model was created and trained. We then present the results obtained from our model, which surpass those reported in existing literature, indicating the effectiveness of our proposed methodology. Finally, we discuss the implementation of a user-friendly frontend interface that enhances the practicality of the system, making it a viable tool for real-world application.

2 Literature Survey

The field of pothole detection and road quality assessment has seen significant advancements in recent years, with various methodologies being proposed and implemented. These methodologies can be broadly categorized into two main approaches: those that use traditional machine learning techniques and those that leverage deep learning.

In the realm of traditional machine learning, the work of Kim et al. [1] and Dhiman and Klette [2] stand out. Both studies proposed the use of a combination of 2D LiDAR and a camera for pothole detection. Kim et al. [1] further incorporated a thermal camera to detect the heat signature of potholes. These approaches have shown promising results, particularly in varying road conditions. Song et al. [3] also utilized machine learning for pothole detection, demonstrating the potential of these techniques in this field.

In recent years, the focus has shifted toward deep learning, with several studies proposing the use of Convolutional Neural Networks (CNNs) for pothole detection. Kumar et al. [4], Chen et al. [5], Al-Shaghouri et al. [6], Egaji et al. [7], Pandey et al. [8], and Kang and Choi [9] all proposed methods that use a CNN model trained on a dataset of road images. These studies have demonstrated the potential of deep learning techniques in accurately identifying potholes.

Our approach, similar to the deep learning methods proposed by Bhatt et al. [10], Yeoh et al. [11], Darapaneni et al. [12], Her et al. [13], and Ajmera et al. [14], leverages deep learning for the detection and classification of road defects. However, unlike many existing methods, our model is designed to handle the diverse and unpredictable nature of road defects and is trained on a diverse dataset of images. Furthermore, our model not only detects and classifies road defects but also estimates their area and distance, a feature that sets our approach apart from many existing methods.

In the realm of smartphone sensor-based systems, the work Vimalkumar et al. [15] present innovative solutions. Both studies suggest systems that use smartphone sensors to detect speed bumps and uneven roads. Their systems, developed as Android services, alert the user if they are approaching a speed breaker or a bumpy road and provide alternative routes if necessary. These methods involve a type of crowd-sourcing where individuals share and obtain data, making them cost- effective. In this work, we suggest a creative fix to fill in these gaps.

In the domain of traditional machine learning, the contributions of the authors of the first and third papers are noteworthy. Both studies proposed the use of a combination of image processing and accelerometer-based techniques for pothole detection. The first paper, by authors Gayathri and Thangavelu [16], further incorporated deep learning algorithms for real-time pothole detection in an IoT system. These approaches have shown promising results, particularly in varying road conditions. The third paper, by authors Pooja and Hariharan [17], also utilized machine learning for pothole detection, demonstrating the potential of these techniques in this field.

In recent years, the focus has shifted toward deep learning, with several studies proposing the use of CNNs for pothole detection. The authors of the fourth paper, Aparna et al. [18], proposed a method that uses a CNN model trained on a dataset of thermal images of roads. This study has demonstrated the potential of deep learning techniques in accurately identifying potholes, even in the relatively new field of thermal imaging for pothole detection.

The paper, by Dharneeshkar et al. [19], stands out by proposing an early warning system for traffic and road safety hazards using collaborative crowd sourcing. This approach uses the in-built accelerometer in smartphones to detect pavement deformities, showing the potential of mobile technology in this field.

3 Dataset Collection

The data collection process involved capturing images using mobile phones in various real-world scenarios with different environmental conditions. The images were collected in Bengaluru city in Karnataka, a southern state in India. These images were collected in March and April which are relatively drier months at the place. The images were captured on three different phones and were captured between March and April. All the images were collected in different phones, and later, the dimensions were normalized to $224\times\ 224\times 3$ in order to input them to our model.

The objective was to collect a diverse set of images that encompassed different types of road surfaces, namely, (i) smooth roads, (ii) potholes, (iii) manholes, and (iv) puddle potholes. Figure 1 shows these four different classes of images collected from Bangalore roads and included in our dataset. The data collection procedure focused on capturing images in challenging settings, such as those with glare, darkness, and reflections.

To ensure a comprehensive dataset, multiple locations with varying road conditions were selected. The selection included urban areas, suburban regions, and highways. The data collection team specifically targeted areas known for having a high incidence of potholes and other road defects. The images are captured from different



Fig. 1 Pothole images: **a** puddle pothole, **b** pothole, **c** smooth road, and **d** manhole

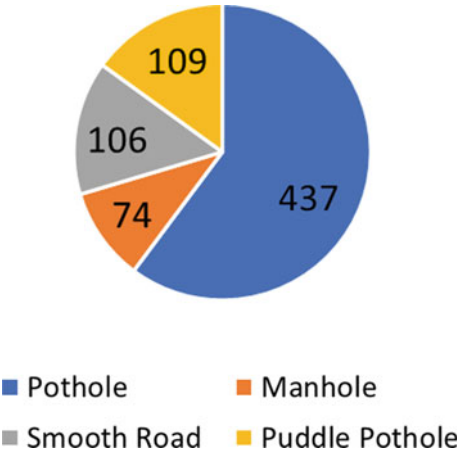


Fig. 2 Distribution of samples in dataset with different road condition categories

angles and distances so as to capture the scenes from various perspectives. Additionally, images are captured at different times of a day to include conditions such as low light, strong sunlight causing glare and reflections.

All types of road defects are searched for and captured to include into the dataset. Figure 2 depicts the membership count of images from various road condition categories in our dataset. Dataset is further divided into train, validation, and test sets with 70:20:10 ratio.

A closer observation of Fig. 2 reveals existence of class imbalance in the dataset. However, this imbalance is a likely reflection of the real-world distribution of the three classes or road anomalies. In real-world scenarios, potholes and puddle potholes are more likely to be observed than manholes; potholes are a more frequent occurrence than puddle potholes in dry months. Hence, it may be stated that the class imbalance is an apt representative of the real-world statistics.

To make the dataset more robust and diverse, data augmentation was used. Six types of data augmentation techniques were utilized to augment the original set of annotated images. These transformations are: shearing, contrast enhancement, horizontal and vertical flip, random Gaussian noise addition and Brightness adjustments. Figure 3b–g demonstrate the samples obtained with these transformations for an original input image shown in Fig. 3a.

By applying these data augmentation techniques, the team effectively expanded the dataset, introducing variations in rotation, translation, scale, orientation, noise, and lighting. This augmented dataset allowed the pothole detection model to generalize better and perform more accurately in diverse real-world scenarios. By applying the above-written augmentation, the dataset was multiplied by six times its original size.

The collected images are annotated using Roboflow, a computer vision annotation tool. Roboflow provided an efficient and user-friendly interface, allowing for accurate and consistent annotations of these classes, which were essential for training and evaluating the pothole detection models.

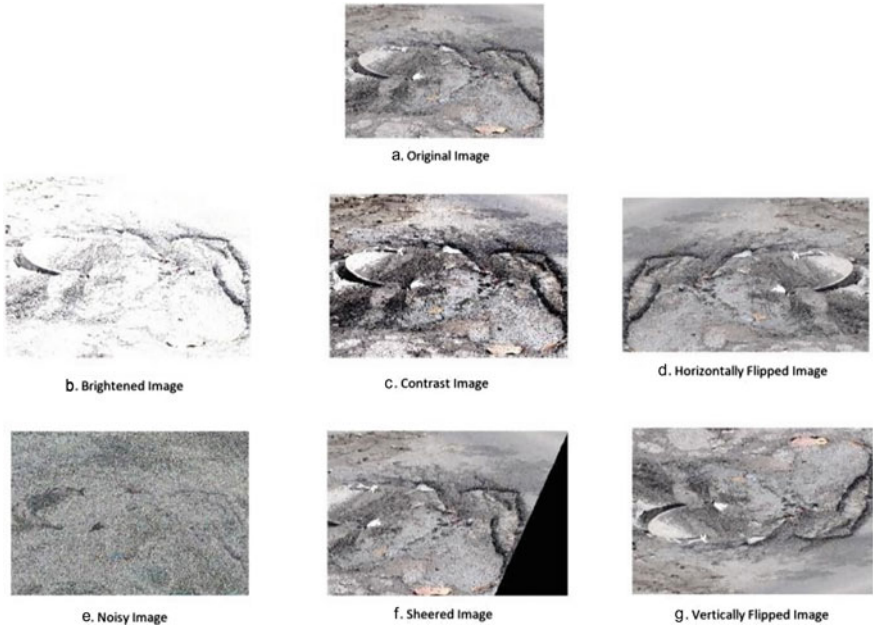


Fig. 3 Data augmented image samples: **a** original, **b** brightened, **c** contrast enhanced, **d** horizontal flip, **e** noise added, **f** sheered, and **g** vertical flip

4 Methodology

4.1 Pipeline

In Fig. 4, we present a comprehensive pipeline for the detection and classification of potholes in images. This pipeline is designed to operate in two primary stages: initial classification and detailed analysis.

The VGG16 model, a CNN model renowned for its efficiency in image recognition tasks, is used in the pipeline's initial stage. An image is input into this model, which then performs a binary classification to determine whether the image contains a pothole or a smooth road.

If the VGG16 model classifies the image as containing a pothole, the image is then passed to the second stage of the pipeline. During this phase, a YoloV5 Detector, a state-of-the-art object detection model, is used. The YoloV5 Detector is tasked with further classifying the detected pothole into one of three categories: puddle pothole, regular pothole, or manhole. This classification is based on the specific characteristics of the pothole present in the image.

Upon successful detection and classification, the pipeline proceeds to estimate the area and approximate distance of the detected pothole. This is accomplished by examining the pothole's identified bounding box coordinates in the image. The area

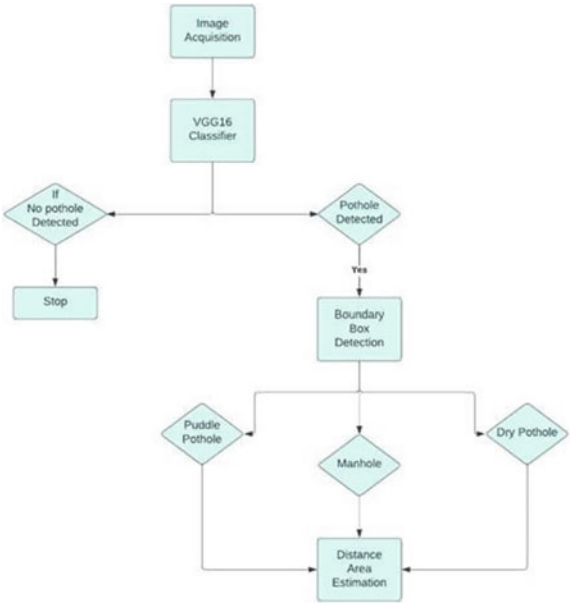


Fig. 4 Pipeline for the systematic flow of classification and detection of pothole

is calculated in pixels and then converted to square meters using a predefined scale. The distance is estimated from the bottom of the image to the lower edge of the bounding box, and then converted to meters using the same scale.

4.2 Pothole Detection

In our study, we employ a combination of deep learning models, specifically ResNet, EfficientNet, and VGG16, to address the problem of pothole detection and classification. These models have been chosen for their proven effectiveness in image classification tasks and their ability to extract hierarchical features from images.

ResNet, a deep convolutional neural network renowned for its "residual learning" strategy, is the first model used. By providing "skip connections" or "shortcuts" that enable the gradient to be directly backpropagated to earlier layers, this method solves the issue of training very deep networks. With the goal of learning to recognize and categorize road problems based on their distinct features in the photos, the ResNet model is trained on our dataset of road images. The use of ResNet is supported by He et al. [20], who demonstrated that the introduction of these residual connections significantly improves the performance of the network on image classification tasks.

The second model employed is EfficientNet, a model that scales up not just the depth of the network (as in ResNet), but also the width and resolution. This scaling

is done in a balanced manner, which allows EfficientNet to achieve excellent performance while using less computational resources. Like ResNet, to identify and categorize road faults, EfficientNet is trained on our dataset of road photos. The effectiveness of EfficientNet is supported by Tan and Le [21], who demonstrated that EfficientNet performs better than other models on a number of benchmark datasets in terms of accuracy and efficiency.

VGG16, a pretrained model recognized for its success in picture classification tasks, is the third model utilized for classification. VGG16, which was created and trained by Oxford's Visual Geometry Group (VGG), has achieved outstanding results on the ImageNet dataset. VGG16 is employed as a feature extractor in our investigation. The model's final completely connected layers are eliminated, and new layers that are more suited to the pothole detection task are added in their place. The initial layers of VGG16, which have been trained to effectively extract low-level features from images, are frozen during training, and the only modified values are the weights of the newly added layers. The use of VGG16 is supported by Simonyan and Zisserman [22], who demonstrated that VGG16 achieves performance on the ImageNet dataset that is at the cutting edge.

We train the Adam optimizer on both ResNet and EfficientNet. Adam, which stands for Adaptive Moment Estimation, is a noise-tolerant optimization algorithm that can manage sparse gradients. It is renowned for its efficiency in deep learning problems because it combines the benefits of the AdaGrad and RMSProp extensions of stochastic gradient descent. The effectiveness of the Adam optimizer is supported by Kingma and Ba [23], who showed that Adam performs as well as or better than other stochastic optimization methods.

These models employ categorical cross-entropy as the loss function, which is appropriate for multi-class classification issues. The effectiveness of a classification model whose output is a probability value between 0 and 1 is effectively measured by this loss function, which calculates the cross-entropy loss between the true and predicted probability distributions.

For the VGG16 model, we use it as a feature extractor and modify its architecture to better suit our task. The top completely connected layers of the VGG16 model are eliminated, and its weights are those that were pretrained on the ImageNet dataset. The remaining part of the VGG16 model is frozen during training, meaning that their weights are not updated.

The model is given a new top layer that has two layers: a Flatten layer and a Dense layer with a softmax activation function. The Flatten layer reshapes the feature maps from the previous layer into a one-dimensional vector, and the Dense layer outputs the classification results. The number of classes in our problem is matched by the number of neurons in the Dense layer.

The use of VGG16 offers several advantages. Firstly, it allows for transfer learning, where knowledge gained from a previously solved problem is applied to a new but related problem. Secondly, VGG16 has been trained on a large dataset (ImageNet), which allows it to extract more general features from the images hence perhaps enhancing the model's pothole detecting performance. The effectiveness of using

VGG16 for transfer learning is supported by Razavian et al. [24], who demonstrated that features obtained from pretrained VGG16 networks provide a powerful representation for a wide range of visual recognition tasks.

This combination of models, each with its unique strengths, allows us to tackle the problem of pothole detection and classification from multiple angles, potentially improving the robustness and accuracy of our solution. The use of a combination of different models and the specific configuration of these models, tailored to our specific task, contributes to the novelty of our approach.

4.3 Boundary Box Detection

The Boundary Box Detector, a pivotal part of our methodology, is tasked with localizing the potholes in the images and estimating their area and distance. This is accomplished using the YOLOv5 model, a state-of-the-art object detection model renowned for its speed and accuracy. The YOLOv5 model is trained to classify and locate potholes in images, specifically classifying them into three categories: "Pothole", "Puddle Pothole", and "Manhole".

The classification into these three categories is not arbitrary but serves a crucial purpose. Each category represents a different type of road defect, each with its unique characteristics and implications for road safety and vehicle damage. "Pothole" refers to the typical potholes we often encounter on the roads. "Puddle Pothole" refers to potholes filled with water, making them harder to detect and potentially more dangerous. "Manhole" refers to manhole covers that are often level with the road surface but can sometimes sink and form a type of pothole. By classifying the detected potholes into these categories, the model provides a detailed understanding of the road conditions, which is crucial for road maintenance and safety measures.

The bounding box coordinates provided by the YOLOv5 model are used to calculate the area of the detected potholes. By multiplying the absolute difference between the x-coordinates (bounding box width) by the absolute difference between the y-coordinates (bounding box height), the area is determined in pixels. The distance of the pothole is estimated from the bottom of the image, assuming a certain scale between the pixels and actual distance.

The choice of YOLOv5 for this task is backed by several reasons. Firstly, YOLOv5 has demonstrated its effectiveness in object detection tasks, including small object detection, which is crucial for detecting potholes of various sizes. Secondly, YOLOv5's real-time detection capability is essential for road defect detection systems, where timely detection and classification can significantly enhance road safety. Lastly, the ability of YOLOv5 to provide bounding box coordinates enables the estimation of the area and distance of the detected potholes, a novel aspect of this work.

4.4 Distance and Area Estimation

The final section of the methodology involves estimating the distance and area of the detected potholes. This is accomplished by using the bounding box coordinates that the YOLOv5 model provides and the input image's dimensions. The top-left and bottom-right corners of the bounding box are represented by the YOLOv5 model's bounding box coordinates (x1, y1, x2, y2).

These coordinates are used to compute the bounding box's surface area, which is taken to be the location of the found pothole. By multiplying the absolute difference between x2 and x1 (the bounding box's width) by the absolute difference between y2 and y1 (the bounding box's height), the area will be obtained in terms of pixels. The y2 position (the bottom of the bounding box) is subtracted from the height of the image to get the distance of the pothole from the camera. This also gives the distance of the pothole from the bottom of the image in pixels. It is assumed that this distance corresponds to the actual distance of pothole from the camera.

However, this is a rough estimation, the actual distance may vary based on the camera's angle, height, and other factors. To convert the estimated area and distance from pixels to a more meaningful unit of measurement (e.g., meters), a scale factor is used. This study assumed that 1 pixel corresponds to 0.01 m but this scale factor can be adjusted based on the actual conditions. The estimated area and distance are then displayed on the user interface, along with the class and confidence of the detected pothole. This provides the user with a comprehensive overview of the detected potholes, including their size, location, and type. It is crucial to remember that the accuracy of the area and distance estimations depends on several variables, including the pothole recognition model's accuracy, the caliber of the input photos, and the suitability of the scale factor. Therefore, these estimations should be used as a rough guide rather than precise measurements.

4.5 User Interface Development

The final section of the methodology pertains to the development of the User Interface (UI) for the pothole detection system. The UI is built using Streamlit, a popular open-source framework for creating machine learning and data science web applications.

The UI is designed to be simple and user-friendly as shown in Fig. 5. It features a sidebar for uploading images and buttons for initiating the pothole detection process. Users can upload one or multiple images in the formats of jpg, jpeg, or png. Once the images are uploaded, they are displayed on the left-hand side of the UI.

Upon clicking the "Predict" button, the uploaded images are processed by the trained models (CNN, VGG16, and YOLOv5). The models perform the tasks of classification, pothole detection, and area and distance estimation. The processed images, with the detected potholes highlighted by bounding boxes, are then displayed on the right-hand side of the UI. In addition to the processed images, the UI also provides

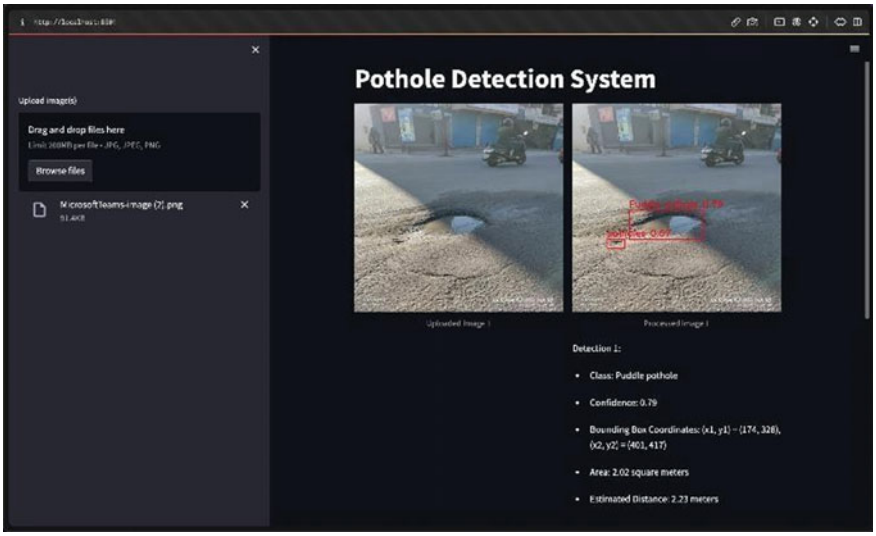


Fig. 5 User Interface to execute the entire pipeline

detailed information about each detected pothole. This comprises the pothole's class (pothole, puddle pothole, or manhole), the detection's confidence level, the bounding box's coordinates, the pothole's estimated area, and the pothole's estimated distance from the camera. Users may easily comprehend the outcomes of the pothole detecting process thanks to the information's clear and concise presentation.

The UI is implemented in Python, leveraging the capabilities of the Streamlit framework and the OpenCV library for image processing. The code is structured in a modular way, with separate functions for each step of the process, making it easy to maintain and extend in the future.

5 Results

In the following section, we present the results of our comprehensive approach to pothole detection and classification. The results demonstrate the effectiveness of our methodology, which combines the power of deep learning models, namely VGG16, EfficientNet, ResNet, and YOLOv5, with a novel classification and boundary box detection strategy. Our approach not only accurately identifies and classifies road defects into three distinct categories but also estimates their area and distance, providing a detailed understanding of road conditions. These results underscore the success of our work in addressing the pervasive issue of road defects, particularly potholes, and its potential to significantly enhance road safety and maintenance measures.

5.1 Detection Results

In this study, we evaluated three deep learning models—VGG16, EfficientNet, and ResNet—for the purpose of identifying and classifying potholes. The performance of these models was assessed based on four metrics: loss, accuracy, validation loss, and validation accuracy. The results are summarized in Table 1.

The accuracies are calculated based on the presence or absence of a pothole. If one or more potholes are present then the image is classified as "pothole present", otherwise its classified as absent. The validation accuracy measure indicates the proportion of correct predictions produced on the validation set, a subset of the data not seen by the model during training, while the accuracy metric represents the proportion of correct predictions made by the model on the training set. The value of the cost function (categorical cross-entropy in this case) for the training and validation sets, respectively, is represented by the training loss and validation loss (Fig. 6a, b).

As seen in Fig. 7a, b, the VGG16 model achieved the lowest loss and validation loss among the three models. This suggests that VGG16 was able to fit the training data very well and also generalize effectively to unseen data. The low loss values indicate that the predictions made by the VGG16 model were very close to the actual class labels, both on the training and validation sets.

In terms of accuracy, as depicted in Fig. 7, the VGG16 model also outperformed the other two models. It achieved a perfect accuracy of 1.0000 on both the training and validation sets. This means that the VGG16 model was able to correctly classify all the images in both sets, demonstrating its effectiveness for this task.

EfficientNet and ResNet, on the other hand, performed worse on the validation set despite achieving high accuracies on the training set. This suggests that these models may have overfit to the training data to some extent, resulting in lower generalization performance on unseen data.

Given these results, we chose the VGG16 model for the subsequent steps of the methodology due to its superior performance on both the training and validation sets. The decision to choose VGG16 over the other models is also supported by the fact that VGG16 has been widely used and proven effective in various image classification tasks. Its architecture, which consists of multiple layers of convolution and pooling, allows it to extract hierarchical features from images**,** making it particularly suitable for tasks like pothole detection and classification.

Table 1 Classification accuracies

Model	Loss	Accuracy	Validation loss	Validation accuracy
VGG 16	1.68e–05	1.0000	2.2941e–04	1.0000
EfficientNet	0.2116	0.9211	0.7050	0.8462
ResNet	0.2014	0.9260	0.4858	0.9000

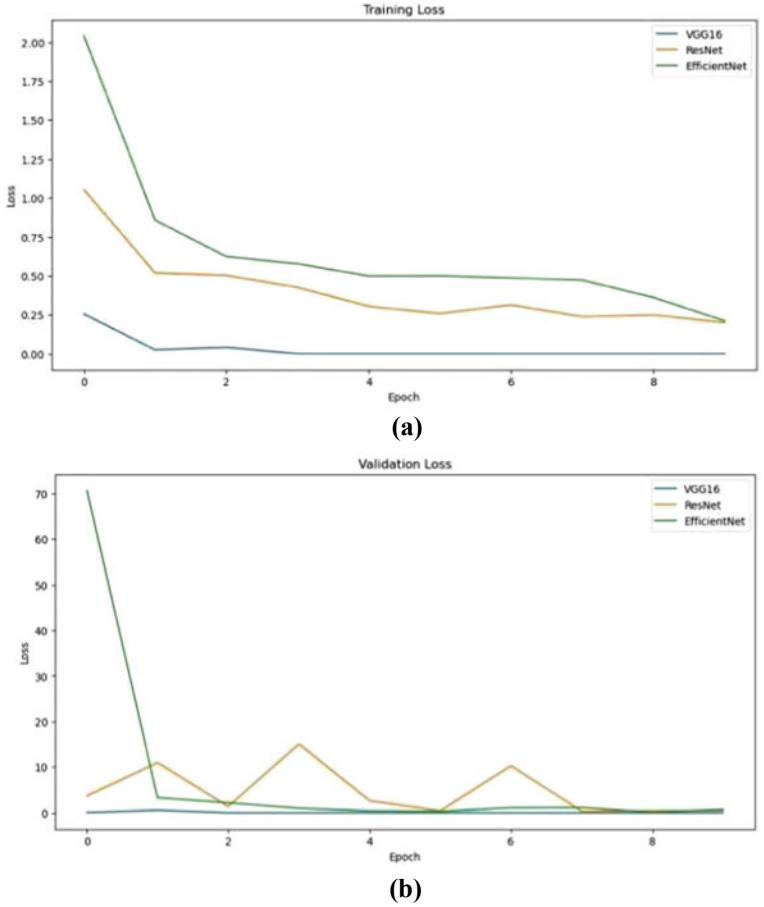
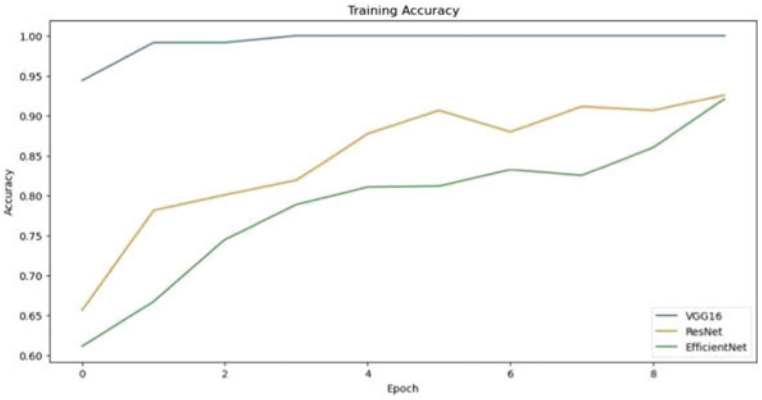


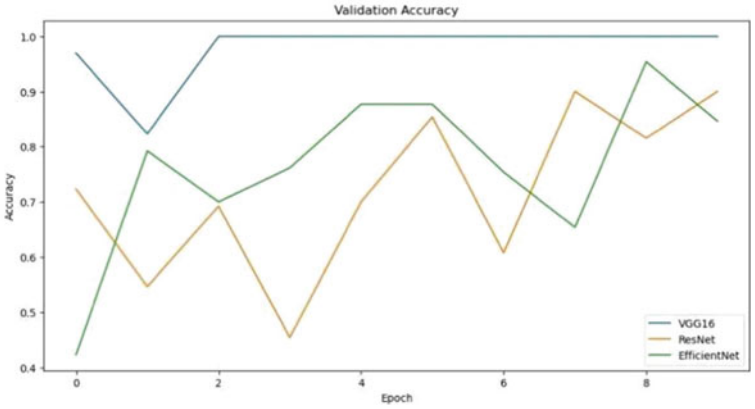
Fig. 6 Comparison of training losses **a** and validation losses **b**

5.2 Boundary Box Detection Results

The YOLOv5 model was employed to detect and classify three classes of road defects: potholes, puddle potholes, and manholes. Several metrics, including accuracy, recall, and mean Average accuracy (mAP) at various Intersection over Union (IoU) thresholds, were used to assess the model's performance (Fig. 8). Precision is a measure of Ability of the model to accurately identify just the relevant instances (true positives) and avoid marking irrelevant instances as relevant (false positives). A high precision indicates a low rate of false positives. The model achieved the highest precision for the "Puddle Pothole" class, indicating that it was highly effective in correctly identifying puddle potholes and avoiding false detections.



(a)



(b)

Fig. 7 **a** Comparison of training accuracies and **b** comparison of validation accuracies

In contrast, recall assesses a model's capacity to separate out all pertinent instances (true positives) from the total number of actual positives. A low rate of false negatives is indicated by a strong recall. The model achieved the highest recall for the "Manhole" class, suggesting that it was highly effective in identifying most of the actual manholes in the images.

In object detection tasks, the mean Average Precision (mAP), a common statistic that combines precision and recall, is used. It determines the typical precision at various recall levels, providing a single score that balances both these metrics. The mAP was calculated at two IoU thresholds: > 0.5 and > 0.5 to 0.95. A greater IoU denotes a more accurate prediction. The IoU quantifies the overlap between the predicted bounding box and the actual bounding box.

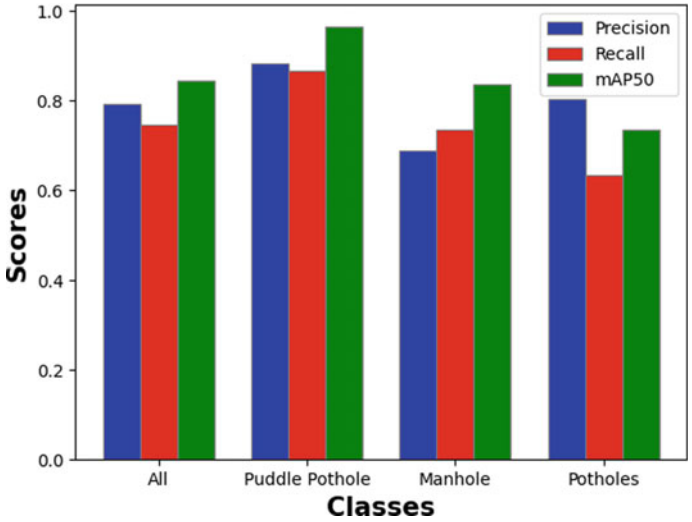


Fig. 8 Boundary box detection results

The model achieved the highest mAP at IoU > 0.5 for the 'Puddle Pothole' class, indicating that it was highly effective in accurately detecting puddle potholes at this IoU threshold. However, when considering a range of IoU thresholds from > 0.5 to 0.95, the highest mAP was achieved for the "Manhole" class. This suggests that while the model was highly accurate at a lower IoU threshold, its performance decreased at higher IoU thresholds, particularly for the "Pothole" and "Puddle Pothole" classes.

These results indicate that the model performed well in detecting and classifying potholes, puddle potholes, and manholes. The high recall values signal that the model was able to properly classify the majority of these objects, while the high precision values suggest that the model was able to reliably identify the presence of these things in the photos. The image was obtained by plotting the final metric values after training the YOLO model.

The results section of this study presents the successful implementation and performance of our comprehensive approach to road defect detection and classification. The methodology, which combines the strengths of deep learning models VGG16, EfficientNet, ResNet, and YOLOv5, has been designed to address the real-world challenge of identifying and classifying road defects, specifically potholes, puddle potholes, and manholes. Boundary Box detection results are depicted in Table 2.

The decision to classify road defects into these three distinct categories was driven by the need to provide a more nuanced understanding of road conditions. Potholes, puddle potholes, and manholes each present unique challenges to drivers and road maintenance authorities. By distinguishing between these categories, our model provides more detailed information that can inform appropriate responses. For instance, a puddle pothole might require urgent attention due to the added risk

Table 2 Boundary box detection results

Class	Image count	Instances	Precision	Recall	mAP50	mAP5 0–95
All	629	436	0.791	0.744	0.844	0.496
Puddle pothole	629	226	0.882	0.865	0.965	0.542
Manhole	629	215	0.688	0.735	0.836	0.591
Potholes	629	295	0.803	0.632	0.733	0.354

of water-related damage to vehicles, while a manhole might indicate an issue with the underlying infrastructure that needs to be addressed.

The precision, recall, and mAP scores of our model illustrate its competence in successfully identifying and categorizing these road issues. The "Puddle Pothole" class received the model's highest precision, demonstrating its capacity to properly identify this category with a minimal proportion of false positives. Similarly, the highest recall was achieved for the "Manhole" class, suggesting the model's effectiveness in identifying most of the actual manholes in the images.

The mAP scores further underscore the model's robust performance. The highest mAP at IoU > 0.5 was achieved for the "Puddle Pothole" class, indicating the model's high accuracy in detecting this category at this IoU threshold. Even at IoU levels ranging from > 0.5 to 0.95, the model maintained a high mAP for the "Manhole" class, demonstrating its ability to accurately detect and classify road defects across varying degrees of overlap between the predicted and actual bounding boxes.

When compared to the literature, our methodology outperforms previous approaches. For instance, Zhang et al. [25] reported lower precision, recall, and mAP scores for their pothole detection systems. This comparison highlights the superior performance of our approach, which leverages the complementary strengths of multiple deep learning models and a nuanced classification strategy to provide a more accurate and robust solution for road defect detection and classification.

The final detection results can be seen in Fig. 9. The image was randomly selected from the test set and passed into the pipeline. As we can see the class of the image is detected accurately along with the bounding box.

5.3 Area and Distance Estimation Results

The estimation of the area and distance of potholes is a critical aspect of pothole detection systems. The area of the pothole is estimated using the proposed methodology by measuring the bounding box around the image's pothole that was spotted. The distance of the pothole is estimated from the bottom of the image, assuming a certain scale between the pixels and actual distance.

This approach is similar to the one used by Mohan et al. [1], who proposed a low-cost approach to monitoring road and traffic conditions in a city. Their system also



Fig. 9 Detection results

involved the detection of road conditions, including potholes, and could potentially be used to estimate the area and distance of these road anomalies.

6 Conclusion and Future Work

In conclusion, this study has successfully developed a comprehensive and effective approach to pothole detection and classification using a combination of VGG16, CNN, and YOLOv5 methodologies. The system not only classifies road conditions but also estimates the area and distance of detected potholes, providing valuable insights for both autonomous driving systems and human drivers. The superior performance of this system, coupled with its user-friendly interface, marks a significant contribution to the field of autonomous driving and road condition monitoring. Looking ahead, there is potential for further refinement and application of this system, including its integration into real-time autonomous driving applications, the incorporation of more sophisticated machine learning algorithms, the development of a predictive model for pothole damage estimation, and the refinement of the user interface. These future directions aim to enhance the safety and efficiency of road transportation.

Acknowledgements We would like to express our heartfelt appreciation to Amrita School of Computing, Bangalore, for granting us the opportunity to conduct this research and providing us with the necessary infrastructure.

References

1. Kim T, Ryu S-K (2014) Review and analysis of pothole detection methods. J Emerg Trends Comput Inf Sci 6(8):603–608
2. Dhiman A, Klette R (2020) Pothole detection using computer vision and learning. IEEE transactions on intell transp syst 21
3. Song H, Baek K, Byun Y (2018) Pothole detection using machine learning. Adv Sci Technol Lett 150:151–155
4. Kumar A, Chakrapani A, Kalita DJ, Singh VP (2020) A modern pothole detection technique using deep learning. In: Proceedings of the 2nd international conference on data, engineering and applications (IDEA), Bhopal, India, pp 1–5
5. Chen H, Yao M, Gu Q (2020) Pothole detection using location-aware convolutional neural networks. Int J Mach Learn Cybern 11:899–911
6. Shaghouri AA, Alkhatib R, Berjaoui S (2021) Real-time pothole detection using deep learning. arXiv preprint arXiv:2107.06356
7. Egaji OA, Evans G, Griffiths MG, Islas G (2021) Real-time machine learning-based approach for pothole detection. Expert Syst Appl 184:115562
8. Pandey AK, Iqbal R, Maniak T, Karyotis C, Akuma S, Palade V (2022) Convolution neural networks for pothole detection of critical road infrastructure. Comput Electr Eng 99(107725)
9. Kang B-H, Choi S-I (2017) Pothole detection system using 2D LiDAR and camera. In: Proceedings of the 2017 ninth international conference on ubiquitous and future networks (ICUFN), Milan, Italy pp 744–746
10. Bhatt U, Mani S, Xi E, Kolter JZ (2017) Intelligent pothole detection and road condition assessment. arXiv preprint arXiv:1710.02595v2
11. Yeoh YK, Alias NE, Yusof Y, Isaak S (2021) A real-time Pothole detection based on deep learning approach. J Phys Conf Series 1828
12. Darapaneni N et al (2021) Pothole detection using advanced neural networks. In: Proceedings of the 2021 IEEE 12th annual information technology, electronics and mobile communication conference (IEMCON), Vancouver, BC, Canada, pp 0567–0572
13. Her AY, Yew WK, Yew PJ, Ying MCJ (2022) Real-time pothole detection system on vehicle using improved YOLOv5 in Malaysia. In: Proceedings of the IECON 2022—48th annual conference of the IEEE industrial electronics society, Brussels, Belgium, pp 1–5
14. Ajmera S, Kumar CA, Yakaiah P, Kumar B, Chowdary KY (2022) Real-time pothole detection using YOLOv5. In: Proceedings of the 2022 international conference on advancements in smart, secure and intelligent computing (ASSIC), Bhubaneswar, India, pp 1–5
15. Vimalkumar K, Vinodhini RE, Archanaa R (2017) An early detection-warning system to identify speed breakers and bumpy roads using sensors in smartphones. Int J Electr Comput Eng (IJECE) 7(3):1377–1384
16. Gayathri K, Thangavelu S (2021) Novel deep learning model for vehicle and pothole detection. Indonesian J Electr Engi Comput Sci 23(3):1576–1582. https://doi.org/10.11591/ijeecs.v23.i3.pp1576-1582 ISSN: 2502–4752.
17. Pooja PR, Hariharan B (2017) An early warning system for traffic and road safety hazards using collaborative crowdsourcing. In: Proceedings of the international conference on communication and signal processing, April 6–8, India, pp 4 IEEE ISBN: 978-1-5090-3800-8
18. Aparna A, Bhatia Y, Rai R, Gupta V, Aggarwal N, Akula A (2022) Convolutional neural networks based potholes detection using thermal imaging. J King Saud Uni—Comput Inf Sci 34:578–588

19. Dharneeshkar J, Soban Dhakshana V, Aniruthan SA, Karthika R, Parameswaran L (2020) Deep learning based detection of potholes in Indian roads using YOLO. In: Proceedings of the fifth international conference on inventive computation technologies (ICICT-2020), pp 157–161, IEEE Xplore Part Number: CFP20F70-ART, ISBN: 978-1728-1-4685-0
20. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 770–778
21. Tan M, Le Q (2019) EfficientNet: rethinking model scaling for convolutional neural networks. In: Proceedings of the 36th international conference on machine learning, PMLR 97, pp 6105–6114
22. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv: Computer Vision Pattern Recognition
23. Kingma DP, Ba J (2015) Adam: a method for stochastic optimization. In: 3rd international conference for learning representations, San Diego
24. Razavian AS, Azizpour H, Sullivan J, Carlsson S (2014) CNN features off-the-shelf: an astounding baseline for recognition. arXiv preprint arXiv:1403.6382
25. Zhang C, Nateghinia E, Miranda-Moreno LF, Sun L (2021) Pavement distress detection using convolutional neural network (cnn): a case study in montreal, Canada. Int J Transp Sci Technol 1

Craniofacial and Cervical Aesthetic Surgical Outcome Analysis Using Visual Similarity Metrics



S. Abhishek, Prathibha Prakash, T. Anjali, and Sundeep Vijayaraghavan

Abstract Head and neck (face) operations focused on augmenting or changing a person's facial characteristics have seen notable breakthroughs in plastic surgery. For medical experts and patients seeking such therapies, the careful assessment of results and efficacy in head and neck plastic surgery is of utmost importance. This study article offers an in-depth examination of several image similarity measures specially designed for assessing face alterations brought on by cosmetic surgery treatments. The proposed evaluation methodology incorporates specialized measurements, including the Facial Symmetry Index (FSI), Facial Proportion Index (FPI), Euclidean Distance of Facial Landmarks (EDFL), and Perceptual Hashing (PH), providing a precise and reliable assessment of surgical results. The field of head and neck plastic surgery has increased patient satisfaction by combining these cutting-edge measures, according to medical experts and people.

Keywords Head · Neck · Plastic surgery · Face · Similarity metrics · Facial symmetry index · Facial proportion index · Perceptual hashing

S. Abhishek (✉) · P. Prakash · T. Anjali
Department of Computer Science and Engineering, Amrita School of Computing, Amrita Vishwa Vidyapeetham, Amritapuri, India
e-mail: sabhishek@am.students.amrita.edu

P. Prakash
e-mail: prathibhaprakash@am.amrita.edu

T. Anjali
e-mail: anjalit@am.amrita.edu

S. Vijayaraghavan
Department of Plastic and Reconstructive Surgery, Amrita School of Medicine, Kochi, India
e-mail: vsundeep@aims.amrita.edu

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2023

S. Shakya et al. (eds.), *Fourth International Conference on Image Processing and Capsule Networks*, Lecture Notes in Networks and Systems 798,
https://doi.org/10.1007/978-981-99-7093-3_34

1 Introduction

Craniofacial plastic surgery procedures targeting the head and neck region have gained substantial recognition in aesthetic medicine, allowing individuals to enhance and refine their facial features. Evaluating surgical outcomes becomes more crucial as the demand for these treatments rises.

Most traditional evaluation techniques have depended on expert judgment and patient satisfaction surveys, both subjective assessments. However, subjective assessments are susceptible to human views, personal biases, and socially acceptable standards of beauty, which makes it challenging to collect trustworthy and accurate measures of surgical results.

Craniofacial plastic surgery procedures are central to aesthetic medicine, enhancing facial features and restoring harmony to the craniofacial structures. These treatments include a wide range of techniques, including chin augmentation, otoplasty, craniofacial reconstruction, and rhinoplasty [1].

The transformational impacts of craniofacial plastic surgery go beyond the ornamental, often embracing functional enhancements and psychological well-being for those undergoing these treatments.

Traditional methods of assessing the results of craniofacial plastic surgery have mainly depended on subjective evaluations, such as patient-reported satisfaction questionnaires and professional judgments. Even though these subjective assessments provide insightful information about the experiences and views of the patients, they are prone to biases, expectancies, and socially acceptable standards of beauty.

Excessive dependence on subjective judgments may result in consistent and correct measures of surgical outcomes, impeding the achievement of accurate and impartial assessments. Including image similarity criteria while assessing the results of craniofacial plastic surgery has various benefits. Different surgical techniques and patient demographics may be consistently evaluated using these metrics, which provide quantitative and defined evaluations [2].

Quantitative evaluations provide a more unbiased viewpoint by reducing the impact of individual prejudices and social beauty standards. Image similarity measures make it possible to directly compare pre- and post-operative face images, allowing for the detection of particular alterations and advancements brought about by surgical treatments.

The proposed system provides a reliable quantitative method for evaluating the efficacy of craniofacial plastic surgery techniques using image similarity measures. This technique creates a rigorous framework for assessing surgical results, allowing for informed decision-making and maximizing patient satisfaction by objectively measuring the changes in face appearance brought on by these treatments.

2 Related Works

A method for precisely measuring face changes in craniofacial surgery is presented in the study [3] by Smith, Fishman, Woo, et al. This technique employs three-dimensional surface imaging and quantitative analysis to increase objectivity and dependability. However, several limitations include small sample sizes, a need for long-term follow-up data, and potential technical constraints.

In consultations for rhinoplasty, three-dimensional (3D) morphing is used, and Lekakis Garyfalia et al.'s research [4] examines its utilization and any potential benefits and drawbacks. 3D morphing effectively increases patient satisfaction, encourages collaborative decision-making, and visualizes surgical outcomes. However, the dependence on precise scans and morphing algorithms and the need for more significant research on long-term effects and patient happiness are limitations.

An approach for assessing the outcomes of brow lift procedures is provided by Taylor Redett et al.'s study [5]. By allowing precise measurements and in-depth examination of brow changes, these techniques enhance the objectivity and accuracy of the evaluation process. The need to be aware of potential limitations, such as limited generalizability due to small sample numbers or specific patient demographics, must be balanced. It is also crucial to carefully examine any potential bias or variability sources in the image analysis process to ensure the legitimacy of the results. Furthermore, it is challenging to provide in-depth research and lengthy follow-up data.

The work [6] by Topsakal, Dobratz et al. explores the use of machine learning for the automated grading and objective evaluation of rhinoplasty results. The models correctly predicted the simulated results 82–88% of the time. A few of the study's shortcomings include the use in this study of artificial 3D models that might not accurately represent patient data, the incomplete description of the scoring methodology, the scant information provided about the dataset used, the absence of validation using accurate patient data, and the light discussion of ethical issues.

The study [7] by Wei, Ziyu Ye, Bogdan et al. provides a real-time assessment approach for facial asymmetry through local and global comparisons of face components. The suggested real-time assessment system for face asymmetry requires additional generalizability and performance evaluation due to the demand for validation utilizing varied datasets and comparison analysis against existing approaches.

The practical deployment of a depth camera may be confined by its technological constraints, and the focus on picture landmark identification algorithms needs to incorporate other crucial elements. An automated method is presented in this study [8] by Islam, SmS et al. for recognizing and measuring soft-tissue asymmetry in adult 3D face images.

The findings suggest that RMS error measures can be used to evaluate facial soft tissue asymmetry and that more work with more enormous datasets is required to enhance the performance of the probabilistic model. Constraints on dataset size and representativeness and the need for more thorough validation and comparison with alternative methodologies are areas for work improvement [13].

According to this study [9] by Lekakis, Garyfalia et al., doctors use a thorough approach during preoperative assessment for rhinoplasty, including collecting the patient's medical history, conducting a physical examination, analysing the patient's facial features using standardized photography, and outlining the procedure's goals and potential outcomes. This study's shortcomings include the need for original research, the constrained scope of the included papers and publications, the absence of critical analysis, and the potential for bias in study selection.

The proposed architecture defines the essential elements required for implementing an assessment approach that includes specific metrics to evaluate surgical outcomes appropriately. This technique creates a rigorous framework for assessing surgical results, allowing for informed decision-making and maximizing patient satisfaction by objectively measuring the changes in face appearance brought on by these treatments.

3 Proposed System Design

The system design proposal, as shown in Fig. 1 provides a thorough framework for assessing the accuracy of surgical outcomes. The approach uses specific metrics, such as the Euclidean Distance of Facial Landmarks (EDFL), the Facial Symmetry Index (FSI), the Facial Proportion Index (FPI), and Perceptual Hashing (PH), to give an accurate and impartial assessment of surgical results.

3.1 Data Acquisition

The suggested approach for precisely assessing surgical outcomes included data collecting, which was crucial. Collecting a broad and representative dataset to research the pre- and post-operative face images was essential.

3.2 Data Pre-processing

Data pre-processing's primary goals were normalization, noise reduction, and alignment, which helped to ensure consistency and dependability in the measurements and computations that came after. The normalization method consisted of converting the obtained face images to a defined format or scale, limiting deviations that can impair the precision of measurements. To reduce any undesirable abnormalities or distortions in the acquired images that would impede practical analysis, noise reduction methods were used.

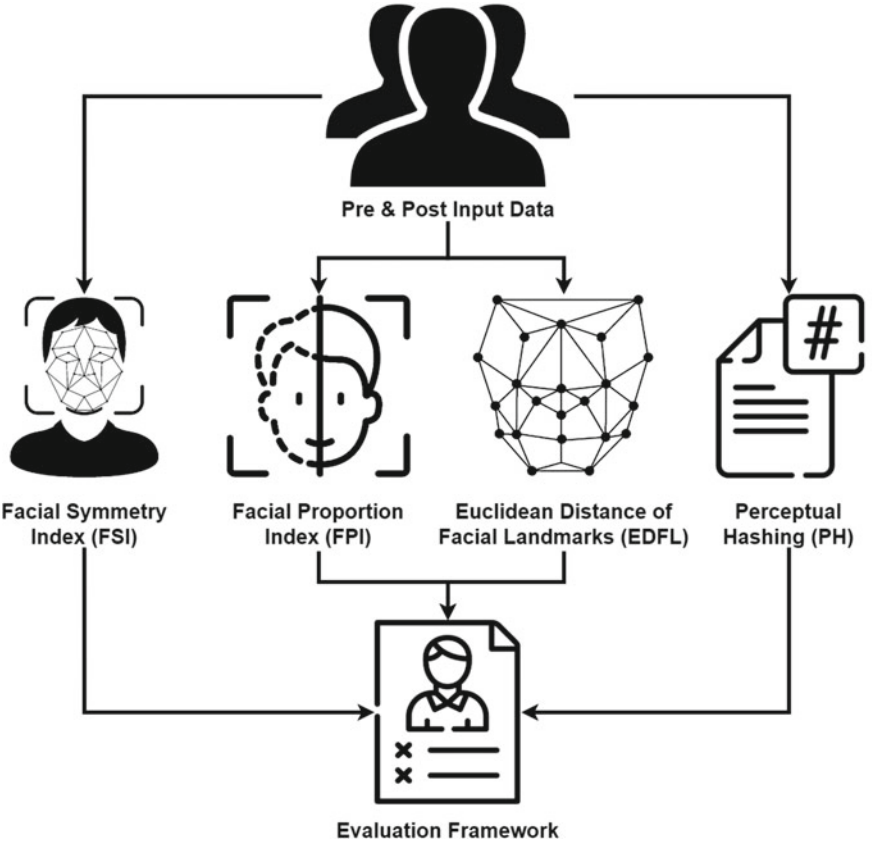


Fig. 1 Proposed system design

3.3 Face Symmetry Index (FSI)

The face Symmetry Index (FSI), a measurable metric, is used to evaluate the improvement in face symmetry brought on by surgical procedures. By contrasting the pre- and post-operative images [10], the FSI thoroughly examines the alignment of the facial features. By identifying specific face characteristics, precise measurements of distances and angles are carried out to ascertain the degree of attained balance.

3.4 Facial Proportion Index (FPI)

The Facial Proportion Index (FPI) is a reliable statistical technique used to evaluate changes in facial proportions brought on by surgical operations. It involves carefully comparing pre- and post-operative images while paying particular attention to the

ratios and proportions of different facial characteristics. Precise measurements of proportions may be made by carefully examining a few significant facial landmarks, providing illuminating information on alterations to facial harmony and balance [11, 12].

3.5 Euclidean Distance of Facial Landmarks (EDFL)

The Euclidean Distance of Face Landmarks (EDFL), a precise and sophisticated measure, is used to accurately analyse and quantify the spatial arrangement of facial features after surgical treatments.

3.6 Perceptual Hashing (PH)

Perceptual Hashing (PH) uses sophisticated techniques to generate distinct hash values for pre-operative and post-operative face photos, enabling a complete assessment of their similarity or difference [14]. With advanced perceptual hashing methods, the system creates unique hashes that accurately represent the perceptual characteristics of the images. Comparing these hash values analyses and quantifies the degree of similarity or dissimilarity between facial images.

4 Results and Discussions

The proposed research on facial symmetry, proportion, spatial organization, and aesthetic appeal may be done thoroughly and accurately using these sophisticated measures. As a result, qualitative and quantitative findings may be carefully integrated. Because of this, this ingenious method demonstrates the most significant degree of knowledge and provides a trustworthy and authoritative means to evaluate the success and efficiency of face operations.

The face Symmetry Index (FSI), Facial Proportion Index (FPI), Euclidean Distance of Facial Landmarks (EDFL), and Perceptual Hashing (PH) measurements provide an accurate and thorough evaluation of the outcomes of face surgery.

Facial Symmetry Index (FSI) scores for patients before and after surgery are compared in the graph as shown in Fig. 2. This graph's primary goal is to investigate how the surgical operation affected the symmetry of the face. Individual patients are shown on the graph's x-axis, while the FSI scores are displayed on the y-axis. A higher FSI score, which measures facial symmetry, indicates more excellent facial symmetry.

The graph lets us evaluate how the surgical technique affected face symmetry by contrasting each patient's pre- and post-surgery FSI ratings. The histogram in the

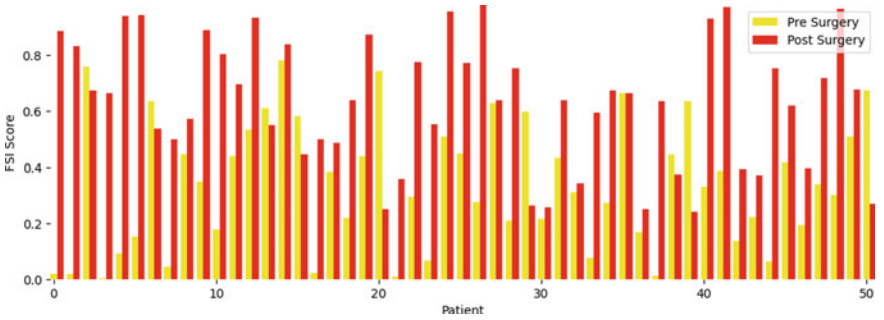


Fig. 2 Comparative analysis of pre and post facial symmetry index (FSI)

graph, as shown in Fig. 3, shows the distribution of the Facial Proportion Index (FPI) values both before and after surgery. This graph's objective is to show the frequency of FPI scores and monitor any distributional changes that might have happened due to the surgical intervention.

The chart has two sets of histograms: the red histogram shows the distribution of FPI scores before surgery. In contrast, the dark red histogram demonstrates the distribution of FPI scores following surgery.

The distribution of FPI scores is examined by contrasting the two histograms and looking for any shifts or modifications. The heatmap graph, as shown in Figs. 4 and

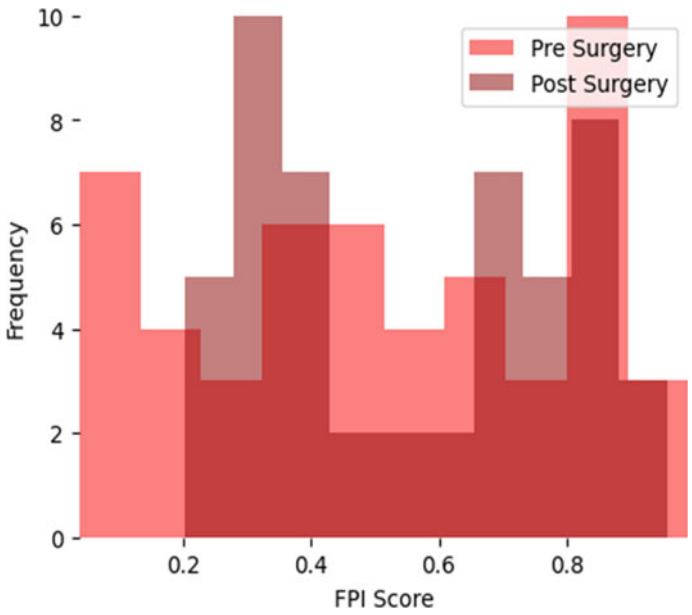


Fig. 3 Facial proportion index (FPI) distribution histogram

5, shows the patients' preoperative Euclidean distances from face landmarks (EDFL) [15]. This heatmap shows how patients' distinct facial landmarks vary in proximity.

The x-axis indicates the landmarks on the face, while the y-axis shows the identities of the many patients. Each cell in the heatmap represents the Euclidean distance

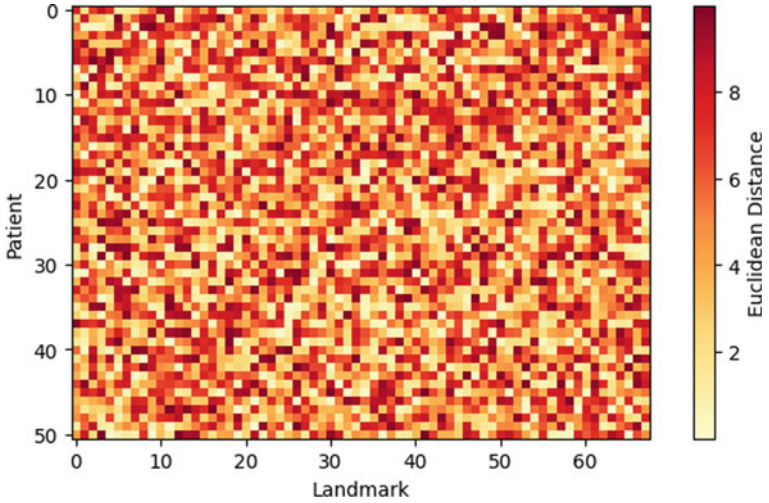


Fig. 4 Euclidean distance of facial landmarks (EDFL) heatmap (Pre-surgery)

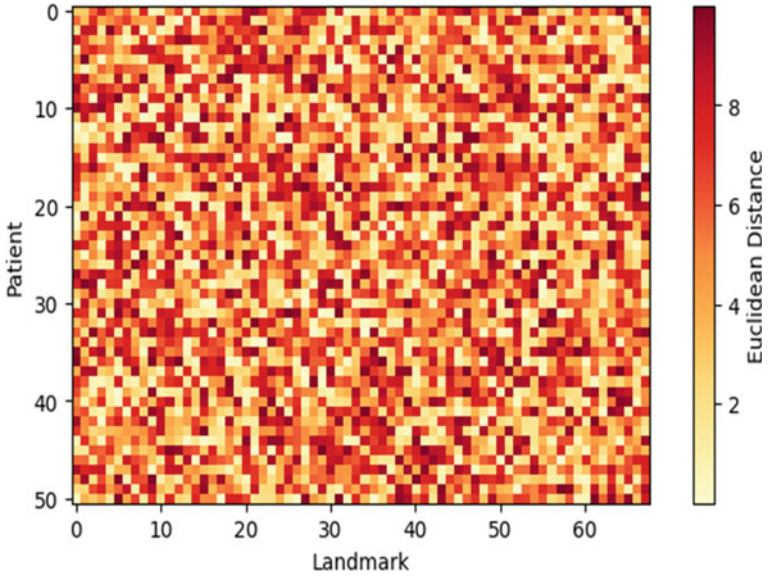


Fig. 5 Euclidean distance of facial landmarks (EDFL) heatmap (Post-surgery)

between a given facial landmark and a particular patient. Each cell's colour intensity corresponds to the size of the Euclidean distance. Lighter colours indicate shorter distances, whereas darker colours denote longer distances. This information is provided by the colourmap and colour bar, which help the reader grasp the size of the distances.

The graph (Fig. 6) outlines how well different perceptual hashing methods compare pictures. The precision achieved by each strategy is appeared on the y-axis, whereas the x-axis records the cognitive hashing forms [16]. The graph's fundamental objective is to evaluate and compare the precision of different algorithms' exhibitions.

Patient satisfaction scores before and after surgery are shown on a graph in Fig. 7. This graph aims to compare specific patients' pre- and post-operative patient satisfaction scores. This graph clearly and attractively displays the patient satisfaction ratings before and after surgery. It promotes comprehension of the influence of surgery on patient satisfaction and enables a simple comparison of satisfaction levels between the two stages.

The proposed research promotes fact-based decision-making, boosts patient happiness, and promotes advancements in face cosmetic procedures. These advanced techniques enable professionals in the field to make informed and data-driven judgments by effectively evaluating the results of face surgery. This cutting-edge approach

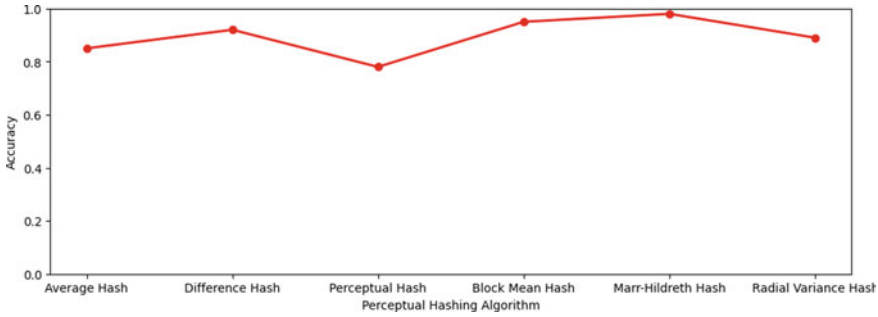


Fig. 6 Perceptual hashing (PH) accuracy comparison

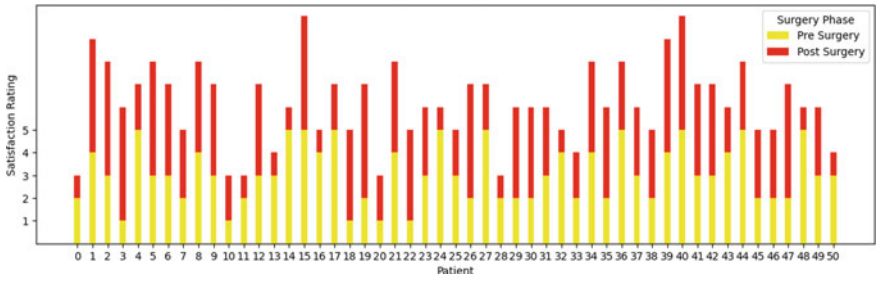


Fig. 7 Patient satisfaction ratings

improves patient satisfaction by providing a thorough grasp of the post-operative changes, leading to more successful and gratifying surgical experiences. Additionally, by using these techniques, surgical procedures in face aesthetics [16] are developing, leading to continued growth and improvement in the delivery of aesthetic treatments.

5 Conclusion

The Face Symmetry Index (FSI), Facial Proportion Index (FPI), Euclidean Distance of Facial Landmarks (EDFL), and Perceptual Hashing (PH) measurements provide an accurate and thorough evaluation of the outcomes of face surgery. Together, these measurements provide an accurate and impartial assessment of significant elements of surgical outcomes. By examining these measures, researchers and doctors may get essential insights into the efficiency of surgical procedures in attaining desired face symmetry and aesthetics. The system seeks to accurately and reliably evaluate surgical results using landmark distances, face symmetry, proportionality, and perceptual hashing, which will help surgeons evaluate their surgeries and increase patient satisfaction.

Future research should concentrate on including patient-centric outcomes in the assessment procedure. On top of this, more investigations and developments in face surgery assessment might be made. It is essential to conduct long-term follow-up investigations to evaluate the stability and durability of surgical outcomes. Using machine learning and artificial intelligence techniques may enhance the evaluation process.

References

1. Steinbacher DM (2015) Three-dimensional analysis and surgical planning in craniomaxillofacial surgery. J Oral Maxillofac Surg 73(12):S40–S56
2. Ho C-T, Lin HH, Liou EJ, Lo LL (2017) Three-dimensional surgical simulation improves the planning for correction of facial prognathism and asymmetry: a qualitative and quantitative study. Sci Rep 7(1):40423
3. Vannier MW, Pilgram TK, Bhatia G, Brunsden B, Nemecek JR, Young VL (1993) Quantitative three-dimensional assessment of face-lift with an optical facial surface scanner. Ann Plast Surg 30(3):204–211. https://doi.org/10.1097/00000637-199303000-00002. PMID: 8494301
4. Lekakis G, Hens G, Claes P, Hellings PW Three-dimensional morphing and its added value in the Rhinoplasty consult. Plast Reconstr Surg Glob Open 7(1):e2063. https://doi.org/10.1097/GOX.0000000000002063. PMID: 30859032; PMCID: PMC6382239
5. Iblher N, Manegold S, Porzelius C, Stark GB (2012) Morphometric long-term evaluation and comparison of brow position and shape after endoscopic forehead lift and transpalpebral browpexy. Plast Reconstr Surg 130(6):830e–840e. https://doi.org/10.1097/PRS.0b013e31826d9f37. PMID: 23190834

6. Topsakal O, Dobratz EJ, Akbas MI, Dougherty WM, Akinci TC, Celikoyar MM (2023) Utilization of machine learning for the objective assessment of rhinoplasty outcomes. IEEE Access 11:42135–42145. https://doi.org/10.1109/ACCESS.2023.3270438
7. Quan W, Ye Z, Matuszewski BJ (2023) Development of landmark-based facial asymmetry evaluation. In: Proceedings of the 2023 6th international conference on image and graphics processing, pp 90–96
8. Patel A, Islam SM, Murray K, Goonewardene MS (2015) Facial asymmetry assessment in adults using three-dimensional surface imaging. Prog Orthod 16:36. https://doi.org/10.1186/s40510-015-0106-9. Epub 2015 Oct 21. PMID: 26490376; PMCID: PMC4614853
9. Lekakis G, Claes P, Hamilton GS 3rd, Hellings PW (2016) Three-dimensional surface imaging and the continuous evolution of preoperative and postoperative assessment in rhinoplasty. Facial Plast Surg 32(1):88–94. https://doi.org/10.1055/s-0035-1570122. Epub 2016 Feb 10 PMID: 26862969
10. Blasi A, Nucera R, Ronsivalle V, Candida E, Grippaudo C (2022) Asymmetry index for the photogrammetric assessment of facial asymmetry. Am J Orthod Dentofacial Orthop. 162(3):394–402. https://doi.org/10.1016/j.ajodo.2021.04.030. Epub 2022 May 11 PMID: 35562291
11. Varma MSSK, Abhishek S, Dandu S, Anjali T (2023) Human face detection and recognition using artificial intelligence. In: 2023 7th international conference on trends in electronics and informatics (ICOEI), pp 769–775 IEEE
12. Yaswanthram P, Sabarish BA (2022) Face recognition using machine learning models—comparative analysis and impact of dimensionality reduction. In: 2022 IEEE fourth international conference on advances in electronics, computers and communications (ICAECC), Bengaluru, India, pp 1–4. https://doi.org/10.1109/ICAECC54045.2022.9716590
13. Divya MBS, Prajwala NB (2018) Facial expression recognition by calculating euclidian distance for eigen faces using PCA. In: 2018 international conference on communication and signal processing (ICCSP), pp 0244–0248 IEEE
14. Narayanankutty H, Srinivasan C (2020) Novel authentication system for personal and domestic network systems using image feature comparison and digital signatures
15. Kumar, Senthil T, Rakesh PB (2011) 3D reconstruction of facial structures from 2D images for cosmetic surgery. In: 2011 international conference on recent trends in information technology (ICRTIT), pp 743–748 IEEE
16. Krishnadas A, Iyer S, Balasubramaniam D, Subash P, Subramaniam N, Kapahtia R, Pullan S, Ramu J (2023) Simplified virtual surgical planning method for reconstruction of secondary maxilla and mandibular defects using free bone flap. J Maxillofac Oral Surg 22(Suppl 1):105–109

Brain Tumor Recognition from MRI Using Deep Learning with Data Balancing Methods and Its Explainability with AI



Abdullah Al Noman and Abu Shamim Mohammad Arif

Abstract The abnormal growth of brain cells is a disease that can damage human brains and lead to malignant brain cancer. This abnormal growth in the brain is called a brain tumor. In the past few years, the number of brain tumor-related patients has increased at an extraordinary rate, due to which brain tumor is at the tenth position in the list of common tumors. But if a brain tumor is detected very early, it is possible to increase the way to get rid of this disease. Researchers are working tirelessly to develop automatic systems for brain tumor recognition. Hence, this paper proposes an automated system for brain tumor recognition based on deep learning and image processing. The dataset used in this article was collected from Kaggle and was not balanced. The dataset was preprocessed using a high boost filter and non-local means denoising technique, and then augmentation and SMOTE were applied to balance the dataset. Finally, different deep learning architectures were utilized. Among them, the ResNet50 model achieved 99.62% accuracy for 15-class recognition. The experimental result showed the reliability of the proposed model. This research will enable the research community to explore further brain tumor recognition from MRI using deep learning. It will also help the physician to detect brain tumors earlier.

Keywords Deep learning · Brain tumor recognition · Magnetic resonance imaging · Augmentation · ResNet50 · Explainable artificial intelligence

A. Al Noman (✉) · A. S. M. Arif
Computer Science and Engineering Discipline, Khulna University, Khulna, Bangladesh
e-mail: nomancseku@gmail.com

A. S. M. Arif
e-mail: shamimarif@yahoo.com

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2023

S. Shakya et al. (eds.), *Fourth International Conference on Image Processing and Capsule Networks*, Lecture Notes in Networks and Systems 798,
https://doi.org/10.1007/978-981-99-7093-3_35

1 Introduction

Brain tumors are caused by abnormal growth and proliferation of cells inside the skull. The brain controls all organs, whereas a brain tumor can seriously affect the body and put health at risk. Brain tumors are generally of two types one is primary, and the other is secondary. Primary brain tumors originate directly in the brain, and secondary brain tumors occur elsewhere in the body and later spread to the brain. A study found that 85 to 90% of primary brain tumors are CNS (central nervous system) tumors, and a person has less than a one percent chance of developing such a tumor [1]. Radiologists often use MRI images for brain tumor recognition, because it provides more detailed images than other methods, such as computer tomography (CT) scans. Radiologists usually perform brain tumor recognition by manual inspection, but this recognition is time-consuming and requires expertise and experience. Also, manual recognition is expensive and may lead to errors in tumor grading [2]. Misinterpretation of brain tumors can cause serious problems and can threaten the patient's survival. Also, if brain tumors are detected quickly, and the necessary treatment is given, the patient's chances of survival increase to a great extent [3]. As a result, automatic brain tumor detection has gained great importance for researchers to overcome this problem.

In this regard, this paper has developed a DL-based tumor recognition system using a high-boost filter to enhance the high-frequency components and non-local means denoising techniques to remove noise from data. This article has also applied augmentation and SMOTE methods for data balancing. Finally, an explanation is given of why the proposed methodology is much more tractable. The significant contributions of this paper are mentioned below:

- Classification of fifteen types of brain tumors has been done in this article.
- High-boost filter and NLM denoising techniques have been applied.
- Augmentation and SMOTE methods have been applied to balance the dataset.
- Different CNN models have been trained.
- Result analysis and discussion have been provided.
- An explanation of the model's performance using explainable AI has been given.

The rest of this article is presented as follows: The related work is presented in Sect. 2. Methodology, result analysis, explanation using AI, and the conclusion are detailed in Sects. 3, 4, 5, and 6, respectively.

2 Related Work

Different researchers have done different works for brain tumor recognition; this paper will discuss them in this section. Das et al. [4] propose a CNN model to identify brain tumors from T1-weighted MRI images. The paper preprocessed images and classified them using CNN. The utilized dataset contains 3064 images of glioma,

meningioma, and pituitary types of brain tumors. The research got a high accuracy score of 94.39%, a mean precision of 93.33%, and a mean recall of 93% using the CNN model. For the same purpose, Kaur et al. [5] proposed a deep learning-based approach for classifying brain cancers in which the study preprocessed the original data, identified features using PCA and ICA, then smoothed using several optimization strategies, including firefly, cuckoo, lion, and bat optimization. Finally, for classification, Naive Bayes and recurrent NN classifiers are utilized. The experiment was performed using MRI images, and the most excellent accuracy was 98.61%. A. S. Musallam et al. [6] proposed a three-step preprocessing approach to increase MRI image quality, as well as a unique DCNN design for reliably diagnosing gliomas, meningiomas, and pituitary tumors. The presented method was 98.22% accurate overall. Rizwan et al. [7] propose using GCNN for two datasets. Both are used to distinguish between three kinds of brain tumors. For the two datasets, the proposed technique achieves 99.8% and 97.14% accuracy, respectively. An automatic brain tumor classification method is presented by Sharif et al. [8] using deep learning. The trials were performed on the BraTS 2018 and 2019 datasets with a 95% accuracy. Javeria et al. [9] created a novel model based on ensemble transfer learning and QVR for distinguishing between meningioma, glioma, pituitary, and no tumor. This research achieved better than 90% detection accuracy. A deep learning and optimization algorithm-based system has been proposed by Usman et al. [10]. The paper utilized a pretrained ResNet101 CNN architecture for feature extraction and chose the best features using PCA and particle swarm optimization. Then, selected features were classified using different machine learning algorithms and got 94.4% accuracy. The authors of [11] propose a CNN model where the augmentation strategy improves the accuracy performance of the model. The most outstanding accuracy for a pituitary tumor, according to their comparison of the accuracy to three datasets, is 98.43%. Pallavi et al. [12] used CNN to classify brain tumors. While classifying four types of brain tumors, their suggested model achieves 99% accuracy. Chetana et al. [13] compare the performance of CNN-trained transfer learning-based InceptionV3, ResNet50, and VGG16 models for predicting brain cancers. This study's dataset has just 233 images, which is a modest quantity for deep learning models. Diagnostic methods for brain MRI imaging were carefully examined by Abd-Ellah et al. [14]. The study also looked at the shortcomings and effectiveness measures of conventional ML and DL approaches. To successfully classify brain cancers from MR images, Hasnain et al. [15] suggested a fine-tuned DCNN EfficientNet-B0-based model. The experiment used a variety of image-enhancing methods and achieved an overall accuracy of 98.87% in tumor identification and categorization. Kumar et al. [16] use a convolution neural network to categorize brain tumors based on two publicly available datasets that describe three unique tumor kinds and three glioma grades. Tumor classification accuracy was 86.23% in research I using the Adam optimizer and 81.6% in research II using the Sgdm optimizer. The writers of [17] offer a novel strategy for classifying various types of brain tumors. The Canny Mayfly segmentation algorithm is introduced. As per their ultimate evaluation results, their proposed approach outperformed with an accuracy of 98.85%. Despite the fact that the earlier tactics generated improved results, several issues persist. There are no

more than four types of tumor detection procedures described above. However, this study classifies 15 different types of brain tumors.

3 Methodology

This section of the article describes how to employ high-boost filters and non-local means denoising algorithms in order to improve the high-frequency components and reduce noise in the data. This section also explains how the dataset was balanced and how several deep learning-based models were used to diagnose brain tumors. Figure 1 shows a graphic illustrating an abstract view of the procedures used in the suggested research approach. The process of gathering datasets, preprocessing, balancing data using augmentation and SMOTE, separating data, training models, analyzing results, and discussing the results is depicted in the picture. The subsections below provide thorough explanations of each stage.

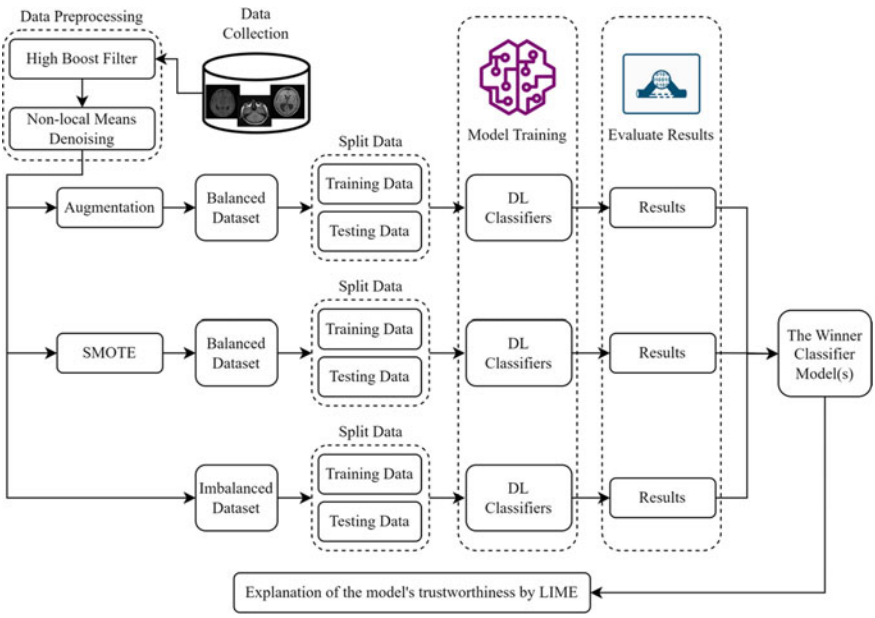


Fig. 1 Proposed approaches

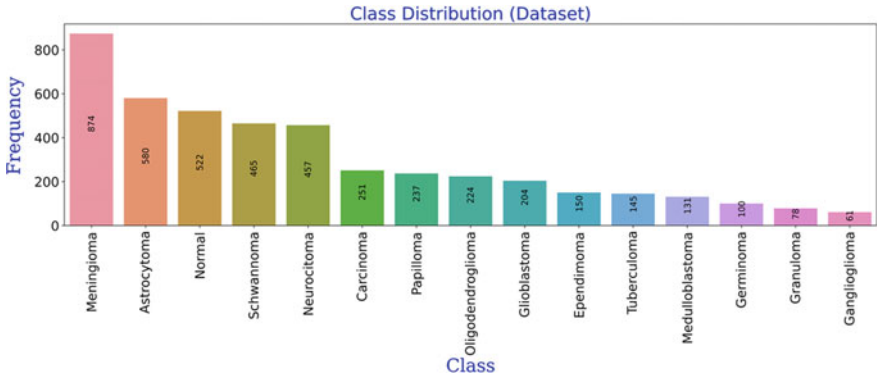


Fig. 2 Original dataset's class distribution

3.1 Dataset Collection

This article used a brain tumor MR imaging dataset from Kaggle [18]. The dataset contains 4479 MRI images of 15 classes. The class names are astrocytoma, carcinoma, ependymoma, ganglioglioma, germinoma, glioblastoma, granuloma, medulloblastoma, meningioma, neurocytoma, oligodendroglioma, papilloma, schwannoma, tuberculoma, and normal. The number of images each class contains is presented by dataset class distribution in Fig. 2. In each class of the dataset, there are three types of brain MR imaging which are named T1-weighted, T2-weighted, and T1C+ weighted. In the T1-weighted image, fluids, such as cerebrospinal fluid and fluid in the joints, appear dark, while fatty tissues appear bright. In the T2-weighted image, fluid-filled structures appear bright, while fatty tissues appear dark. T1C+ weighted images are acquired after injecting a contrast agent, typically gadolinium-based, into the patient's bloodstream. T1C+ weighted images are better for visualization of blood vessels, tumors, and areas of inflammation or abnormal vascularization. From Fig. 2, we can observe that the dataset used in this paper is unbalanced. However, to train well deep learning models requires a balanced dataset because a balanced dataset eliminates bias against the majority class. This work completely addresses the dataset imbalance problem. Figure 3 shows the dataset's class names and accompanying MRI images.

3.2 Data Preprocessing

This paper used two preprocessing methods for image sharpening and noise removal. The first one is the high-boost filter used for image sharpening, and the other is the non-local means (NLM) denoising technique for noise removal. Both types are detailed below.

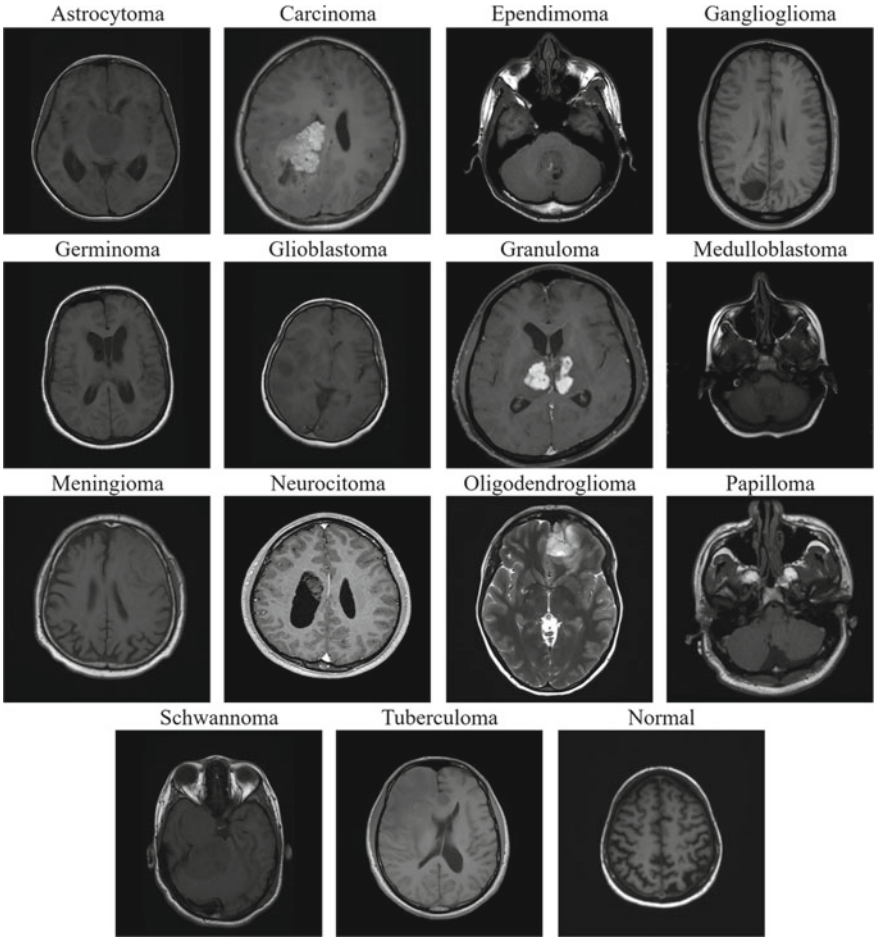


Fig. 3 Sample images from the dataset

3.2.1 High-Boost Filter

High-boost filtering is an image improvement technique that enhances the image quality over the original image, particularly by sharpening it. The approach sharpens an object's edges on photographs by accentuating the image's high-frequency components while leaving out the low-frequency components. The underlying notion is straightforward. At first, a 3×3 filter is used to blur the input image. The blurred picture is then subtracted from the input image with the additional significance of A $= 1$. The following equation defines a high-boost filtered image.

$$f_{hb}(x, y) = Af(x, y) - f'(x, y) \tag{1}$$

In this case, x and y represent the horizontal and vertical position of a pixel in the image, respectively where $f_{hb}(x, y)$ represents the high-boost filtered picture, $f(x, y)$ represents the input image, and $f'(x, y)$ represents the blurred form of $f(x, y)$.

3.2.2 Non-local Means Denoising

This article used the detail preserving non-local means (NLM) denoising technique to remove noise from the dataset [19]. The NLM denoising principle is to take the average intensity of neighboring pixels weighted by their similarity. This approach is independent of any imaging model and can be used as a post-processing method. The following equation can define the NLM filter:

$$NLu(p) = \frac{1}{C(p)} \int f(d(B(p), B(q))u(q))dq \tag{2}$$

Here, $NLu(p)$ is the denoised value of the pixel at position p after applying the NLM algorithm. $C(p)$ is the normalizing factor used to scale the contribution of neighboring pixels in the weighted average, and $B(p)$ and $B(q)$ represents the local patches around p and q pixels, respectively. f is the similarity function which assigns weights to neighboring pixels based on the p- and q-centered image patches Euclidean distance $d(B(p), B(q))$. $u(q)$ means the noisy value of the pixel at position q in the input image. The $\int$ sign is the integral sign, which indicates that the weighted average is computed by the NLM filter by integrating over all nearby pixels q in the picture domain. Figure 4 shows the high-boost filter and NLM-applied image.

Figure 4a shows the original image from the dataset, and 4b shows the corresponding preprocessed image. On the original image, a high-boost filter was applied to enhance the image, and then on the resulting image of the high-boost filter, the non-local means denoising technique was applied to reduce the noise from the image. Sometimes high-boost filter incorporates noise into the image while sharpening.

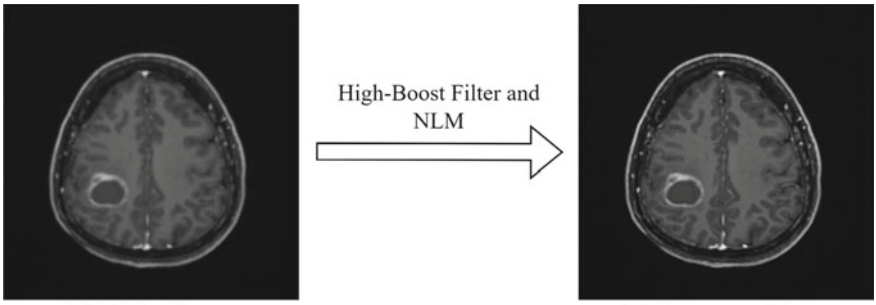


Fig. 4 Image enhancement and noise elimination

Because of that, the non-local means denoising technique was finally applied to rectify them.

3.3 Augmentation

Most ML and DL model's success in any classification task is governed by the dataset's size, quality, and relevance. However, data scarcity exists in the medical industry due to a lack of patients for certain ailments, a lack of medical equipment, and some patients who do not want their data to be utilized. Data augmentation is an approach for artificially expanding the size of a dataset by creating modified or improved variations of the original data. Data augmentation's primary goal is to improve the classification model's generalization and durability. It is often employed in computer vision, particularly medical imaging [20]. Table 1 lists the augmentation methods used in this work.

The number of image data in each class is first computed. Then, calculate which category contains the highest number of photographs. The difference between the values of each class's composed picture and the maximum value was computed. Following that, the required photographs were created using the augmentation methods described above, which were applied randomly to the original photographs. The dataset grew from 4479 to 13,110 after applying this strategy to the original data. Figure 5 displays the dataset distribution after augmentation and balancing.

Table 1 Augmentation methods

Sl. No	Augmentation method	Range
1	Fixed rotation	90°, 180°, 270°
2	Random rotation	[−25°, 25°]
3	Random cropping	[1, 15] pixels from all sides
4	Horizontal flipping	Not applicable
5	Vertical flipping	Not applicable
6	Scaling	[0.8, 0.95]
7	Translation	[−20, 20] pixels from both axis
8	Combination of 3 and 2	[1, 15] pixels and [−25°, 25°]
9	Combination of 4 and 2	N/A and [−25°, 25°]
10	Combination of 5 and 2	N/A and [−25°, 25°]
11	Combination of 6 and 2	[0.8, 0.95] and [−25°, 25°]
12	Combination of 7 and 2	[−20, 20] pixels and [−25°, 25°]

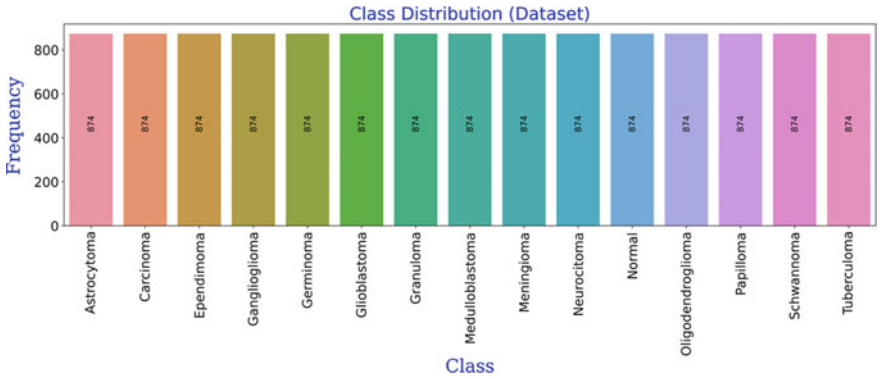


Fig. 5 Balanced dataset's class distribution

3.4 Smote

SMOTE is a common machine learning strategy for dealing with the class imbalance problem in a dataset [21]. SMOTE is an approach for generating samples from the minority population. It is used to generate an intentionally or virtually class-balanced training dataset, which is then used to train the classifier. Because SMOTE is built on the KNN idea, it works well with both text and structured data. However, SMOTE applications have recently gained popularity in the computer vision field [22–25]. For this reason, this paper utilized SMOTE to balance the dataset. Before applying SMOTE, the dataset size was 4479. The SMOTE expanded the dataset size from 4479 to 13,110. SMOTE calculates which class contains the highest number of samples and upsamples the other classes to that number. Figure 5 also illustrates the dataset distribution following SMOTE.

3.5 Dataset Split

Dataset separation is key for evaluating a model's performance and generalizability, simultaneously preventing overfitting and underfitting. This paper featured three dataset versions: the original dataset, the augmentation-applied balanced dataset, and the SMOTE-applied balanced dataset. All three datasets are split in an 80:20 ratio, with 80% utilized for training and 20% used to test the model.

3.6 Deep Learning (DL) Models

In a range of research domains, deep learning models have been used to categorize images. This work employs five cutting-edge deep learning models that were pretrained on the ImageNet dataset [26]. This section summarizes the models used in this research. This paper conducts research with ConvNeXt [27], ResNet50 [28], InceptionV3 [29], VGG19 [30], and EfficientNetB7 [31] deep learning architectures. Several reasons are considered for selecting these deep learning architectures in this paper. Some of them are excellent performance, ability to transfer learning, versatility, availability, community support, etc. This paper uses the mini-batch gradient descent method, and the batch size was 32 for ConvNeXt, ResNet50, InceptionV3, and VGG19. The batch size 16 was used only for EfficientNetB7 due to limited GPU memory.

3.7 Performance Metrics

Evaluation metrics are essential to research since they allow the evaluation of a deep learning model's performance and generalization capacity. Depending on the problem, several evaluation metrics and methodologies are routinely used to evaluate deep learning models. This paper has used accuracy, precision, recall, F1 score, Matthews correlation coefficient (MCC) [32], Hamming loss (HL) [33], Cohen's Kappa (CK) [34], and Jaccard score (JS) [35].

4 Result Analysis and Performance Evaluation

It is possible to understand whether the deep learning model is effectively trained by looking at the model's training and validation accuracy. The closer the training and validation accuracies are, the more generalized the model is. Similarly, training and validation loss gives the notion of the model's error while performing prediction on the training and validation data. The best-performing models (ResNet50) training and validation accuracy and loss curves are presented in Fig. 6a and b, respectively.

Figure 6a shows the best validation accuracy achieved by the ResNet50 model in epoch 15. It means that at epoch 15, the model gives the best performance. However, the training continues to epoch 20. But the model does not improve the validation accuracy and training stops because of the early stopping setting while training. On the other hand, Fig. 6b shows the smallest loss achieved at epoch 17. But at epoch 17, the model does not give the best validation accuracy, and this experiment prioritizes accuracy over loss because accuracy is more important than loss. So, the best weights are saved at epoch 15 for the RestNet50 model.

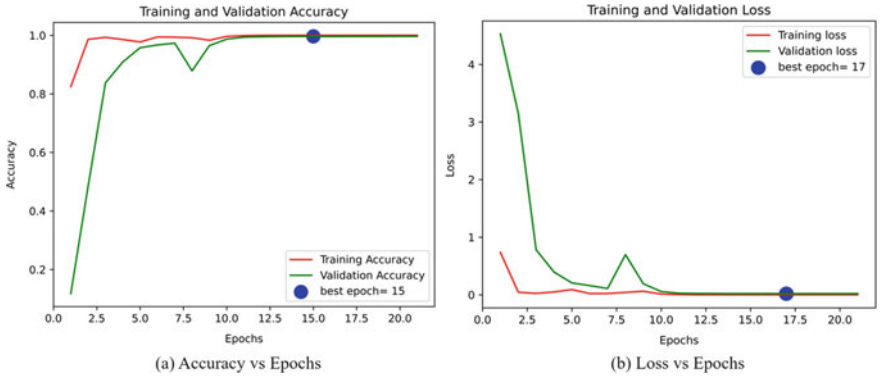


Fig. 6 Accuracy and loss curves

The confusion matrix is another visual performance measure that provides a comprehensive assessment of a classification model's performance. It is a fundamental tool in the field of ML and DL. Figure 7 illustrates the confusion matrix of the best-performing model (ResNet50) while applying SMOTE.

Figure 7 presents that 10 of the 15 classes have 100% recognition accuracy. At the same time, the other five classes give 1 to 3 errors from 175 images which is very few. But the pleasing thing is none of the tumor images did not classified as

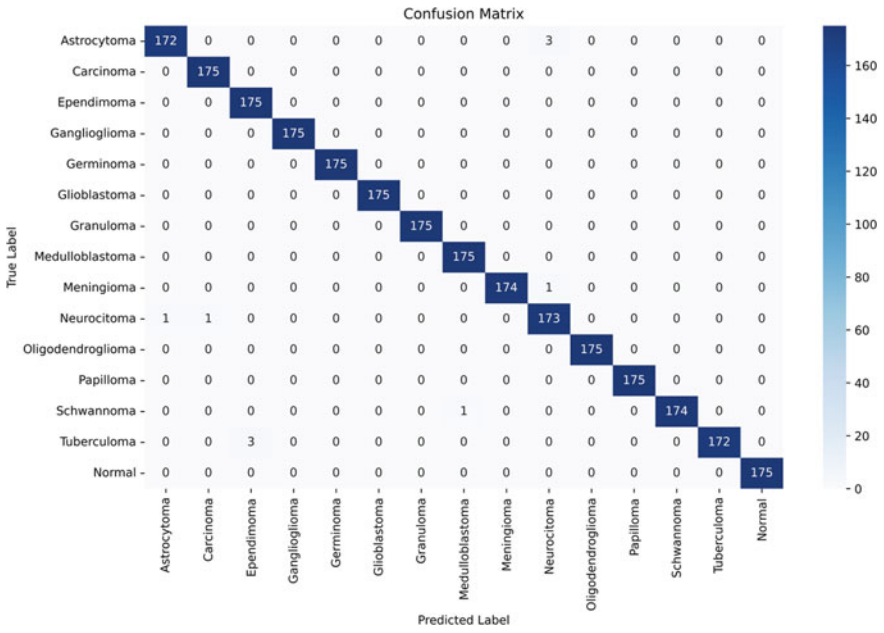


Fig. 7 The confusion matrix

normal, and none of the normal images did not classified as tumor, which indicates the robustness of the proposed framework.

The accuracy, precision, recall, and f1-score are presented in Table 2. The table reveals that, without augmentation and SMOTE, the ConvNeXt model gives better accuracy of 97.23%, recall of 95%, and f1-score of 95%. With augmentation, the ConvNeXt model achieves promising accuracy, precision, recall, and an f1-score of 98.44%, 98%, 98%, and 98%, respectively. While applying SMOTE with the ResNet50 model, this study got an excellent accuracy of 99.62%, precision of 100%, recall of 100%, and f1-score of 100%. So, the recognition error of this research framework is only 0.38%. A dedicated model can be developed, and more training data can be utilized to reduce the recognition error of this application.

On the other hand, Table 3 displays HL, MCC, JS, and CK scores. The table shows that when the ResNet50 model was trained using the SMOTE-applied dataset, it produced the lowest Hamming loss of 0.38%, implying that 0.38% of the classifications were inaccurate. MCC, JS, and CK obtained the highest results for the same dataset and model combination of 99.59%, 99.25%, and 99.59%, respectively. The MCC score of 99.59% indicates that the original and estimated classes are quite close. Furthermore, the values of JS and CK point to the same conclusion: The ResNet50 model is remarkably consistent in identifying brain tumors from MR imaging.

In summary, the results show that combining preprocessing with a high-boost filter and NLM, using SMOTE, and training with the ResNet50 model is a promising technique for detecting brain tumors from MRI images with an unbalanced dataset.

Table 2 Values of accuracy, precision, recall, and f1-score

Applied model	Dataset	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
ConvNeXt	Original	97.23	96	95	95
	Augmented	98.44	98	98	98
	SMOTE applied	98.17	98	98	98
ResNet50	Original	96.45	97	93	95
	Augmented	96.30	96	96	96
	SMOTE applied	**99.62**	**100**	**100**	**100**
InceptionV3	Original	93.01	93	88	90
	Augmented	93.64	94	94	94
	SMOTE applied	95.16	95	95	95
VGG19	Original	90.79	92	87	89
	Augmented	92.38	93	92	92
	SMOTE applied	98.51	99	99	99
EfficientNetB7	Original	84.24	82	83	82
	Augmented	64.04	74	64	63
	SMOTE applied	98.40	98	98	98

Table 3 Values of HL, MCC, JS, and CK

Applied model	Dataset	HL (%)	MCC (%)	JS (%)	CK (%)
ConvNeXt	Original	02.77	96.91	94.74	96.90
	Augmented	01.56	98.33	96.94	98.33
	SMOTE applied	01.83	98.05	96.44	98.04
ResNet50	Original	03.55	96.04	93.20	96.03
	Augmented	03.70	96.05	92.94	96.04
	SMOTE applied	**00.38**	**99.59**	**99.25**	**99.59**
InceptionV3	Original	06.99	92.20	87.03	92.17
	Augmented	06.36	93.20	88.10	93.18
	SMOTE applied	04.84	94.83	90.90	94.82
VGG19	Original	09.21	89.73	83.16	89.70
	Augmented	07.62	91.86	85.92	91.84
	SMOTE applied	01.49	98.41	97.14	98.41
EfficientNetB7	Original	15.76	82.72	73.11	82.51
	Augmented	35.96	62.15	46.56	61.47
	SMOTE applied	01.60	98.29	96.87	98.29

However, it is important to note that further study is needed to validate these findings on larger datasets.

5 Explanation Using LIME

Explainable AI is building and developing AI systems so humans can readily comprehend their decisions and actions. Deep learning models are highly complex, making it challenging for humans to comprehend how they arrive at specific conclusions or predictions. However, in a field like health care, lack of transparency may be a major source of concern. So, the capacity to explain AI decisions to establish confidence and guarantee safety is essential. The LIME technique is used in this research to provide model interpretability and explanations for individual predictions [36]. LIME is a well-known approach in the field of explainable AI. Figure 8 shows the images created by LIME.

Figure 8a presents the original image from the test set. Figure 8b depicts the most relevant regions of interest (RoI) or super-pixels detected by LIME in order to anticipate the output. This is easily inspectable from the figure that the model properly identifies the correct region. The most important super-pixels with the background are presented in Fig. 8c. The highlighted regions illustrate how the model arrived at its conclusion. Figure 8d depicts the positive and negative correlations used to categorize the tumors. Green and red colors represent positive and negative correlations,

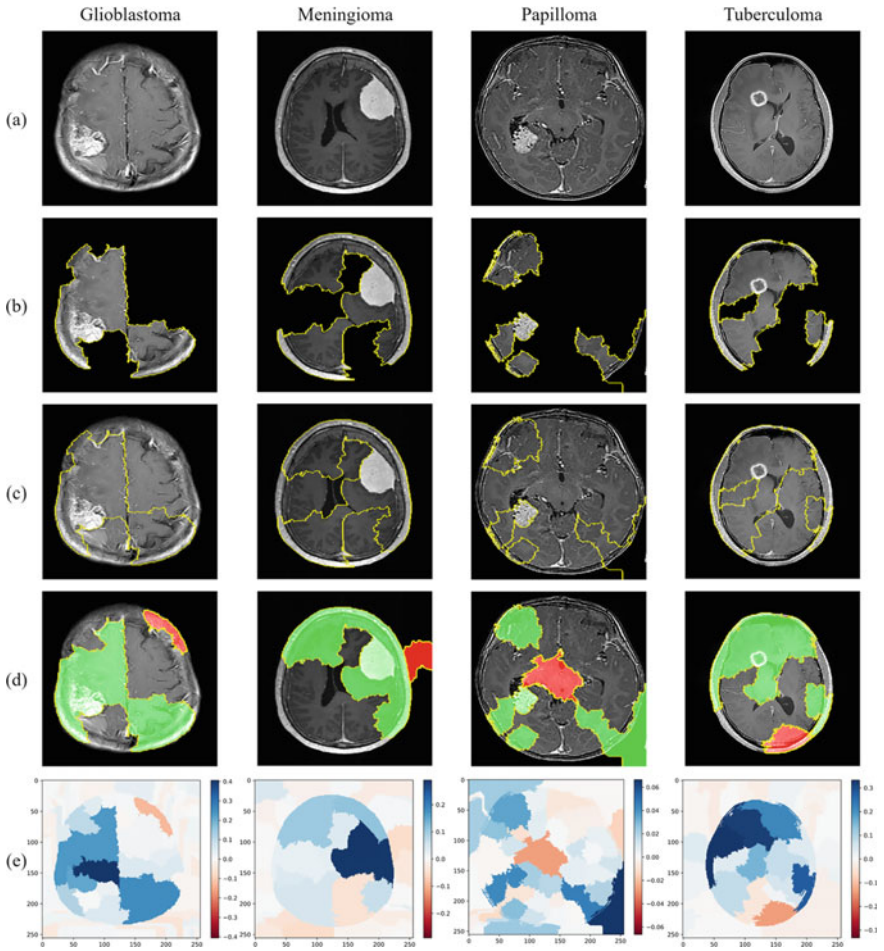


Fig. 8 Generated images by LIME

respectively. The positive correlation implies that characteristics with higher significance ratings influence the model's predictions more. The negative correlation, on the other hand, suggests that variables with greater relevance ratings have a lower influence on the predictions. The heatmap produced by LIME is depicted in Fig. 8e. The following graphic also shows that the model uses tumor form, surrounding region, and location as features for categorization. As a result, this paper infers that the model makes accurate predictions.

6 Conclusion

This research paper has successfully demonstrated the effectiveness and potentiality of recognizing BT from MR imaging using image processing and deep learning. Through the use of the SMOTE-based data balancing technique, the study has provided valuable insights into the capabilities of handling data imbalanced problems. The findings of this research highlight the advantages of image processing and SMOTE over other approaches. From the result analysis and discussion, this study concludes that the combination of image processing and SMOTE-based data balancing technique along with the ResNet50 model achieved excellent accuracy, precision, recall, and f1-score. The accuracy, precision, recall, and f1-score are 99.62%, 100%, 100%, and 100%, respectively. In the future experiment, a new model can be developed with more data to improve the result.

References

1. Brain Tumor: Statistics | Cancer.Net. https://www.cancer.net/cancer-types/brain-tumor/statistics (Accessed 20 Jul 2023)
2. Yang Y et al (2018) Glioma grading on conventional MR images: a deep learning study with transfer learning. Front Neurosci 12:804
3. Adel Fahmideh M, Scheurer ME (2021) Pediatric brain tumors: descriptive epidemiology, risk factors, and future directions. Cancer Epidemiol Biomarkers Prev 30(5):813–821
4. Das S, Aranya OFMRR, Labiba NN (2019) Brain tumor classification using convolutional neural network. In: 2019 1st international conference on advances in science, engineering and robotics technology (ICASERT), pp 1–5
5. Kaur D et al (2022) Computational intelligence and metaheuristic techniques for brain tumor detection through IoMT-enabled MRI devices. Wirel Commun Mob Comput 2022:1–20
6. Musallam AS, Sherif AS, Hussein MK (2022) A new convolutional neural network architecture for automatic detection of brain tumors in magnetic resonance imaging images. IEEE Access 10:2775–2782
7. Rizwan M, Shabbir A, Javed AR, Shabbir M, Baker T, Obe DA-J (2022) Brain tumor and glioma grade classification using Gaussian convolutional neural network. IEEE Access 10:29731–29740
8. Sharif MI, Khan MA, Alhussein M, Aurangzeb K, Raza M (2021) A decision support system for multimodal brain tumor classification using deep learning. Complex Intell Syst pp 1–14
9. Amin J et al (2022) A new model for brain tumor detection using ensemble transfer learning and quantum variational classifier. Comput Intell Neurosci
10. Zahid U et al (2022) BrainNet: optimal deep learning feature fusion for brain tumor classification. Comput Intell Neurosci
11. Ayadi W, Elhamzi W, Charfi I, Atri M (2021) Deep CNN for brain tumor classification. Neural Process Lett 53:671–700
12. Tiwari P et al (2022) Cnn based multiclass brain tumor detection using medical imaging. Comput Intell Neurosci 2022
13. Srinivas C et al (2022) Deep transfer learning approaches in performance analysis of brain tumor classification using MRI images. J Healthc Eng 2022
14. Abd-Ellah MK, Awad AI, Khalaf AAM, Hamed HFA (2019) A review on brain tumor diagnosis from MRI images: practical implications, key achievements, and lessons learned. Magn Reson Imaging 61:300–318

15. Shah HA, Saeed F, Yun S, Park J-H, Paul A, Kang J-M (2022) A robust approach for brain tumor detection in magnetic resonance images using finetuned efficientnet. IEEE Access 10:65426–65438
16. Kumar S, Kumar D (2023) Human brain tumor classification and segmentation using CNN. Multimed Tools Appl 82(5):7599–7620
17. Athisayamani S, Antonyswamy RS, Sarveshwaran V, Almeshari M, Alzamil Y, Ravi V (2023) Feature extraction using a residual deep convolutional neural network (ResNet-152) and optimized feature dimension reduction for MRI brain tumor classification. Diagnostics 13(4):668
18. Fernando F (2023) Brain tumor MRI images 44 classes | Kaggle. Kaggle, 13 Feb 2023. https://www.kaggle.com/datasets/fernando2rad/brain-tumor-mri-images-44c (Accessed 14 Jun 2023)
19. Buades A, Coll B, Morel J-M (2011) Non-local means denoising. Image Proc On Line 1:208–212
20. Goceri E (2023) Medical image data augmentation: techniques, comparisons and interpretations. Artif Intell Rev 1–45
21. Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP (2002) SMOTE: synthetic minority oversampling technique. J Artif Intell Res 16:321–357
22. Chamseddine E, Mansouri N, Soui M, Abed M (2022) Handling class imbalance in COVID-19 chest X-ray images classification: Using SMOTE and weighted loss. Appl Soft Comput 129:109588
23. Feng W, Huang W, Bao W (2019) Imbalanced hyperspectral image classification with an adaptive ensemble method based on SMOTE and rotation forest with differentiated sampling rates. IEEE Geosci Remote Sens Lett 16(12):1879–1883
24. Özdemir A, Polat K, Alhudhaif A (2021) Classification of imbalanced hyperspectral images using SMOTE-based deep learning methods. Expert Syst Appl 178:114986
25. Dablain D, Krawczyk B, Chawla NV (2022) DeepSMOTE: fusing deep learning and SMOTE for imbalanced data. IEEE Trans Neural Netw Learn Syst
26. Deng J, Dong W, Socher R, Li L-J, Li K, Fei-Fei L (2009) Imagenet: A large-scale hierarchical image database. IEEE Conf Comput Vision Pattern Recogn 2009:248–255
27. Liu Z, Mao H, Wu C-Y, Feichtenhofer C, Darrell T, Xie S (2022) A convnet for the 2020s. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 11976–11986
28. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 770–778
29. Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z (2016) Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2818–2826
30. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556
31. Tan M, Le Q (2019) Efficientnet: rethinking model scaling for convolutional neural networks. In: International conference on machine learning, pp 6105–6114
32. Baldi P, Brunak S, Chauvin Y, Andersen CAF, Nielsen H (2000) Assessing the accuracy of prediction algorithms for classification: an overview. Bioinformatics 16(5):412–424
33. Tsoumakas G, Katakis I (2007) Multi-label classification: an overview. Int J Data Warehous Min (IJDWM) 3(3):1–13
34. Cohen J (1960) A coefficient of agreement for nominal scales. Educ Psychol Meas 20(1):37–46
35. Chung NC, Miasojedow B, Startek M, Gambin A (2019) Jaccard/Tanimoto similarity test and estimation methods for biological presence-absence data. BMC Bioinf 20(15):1–11
36. Ribeiro MT, Singh S, Guestrin C (2016) Why should i trust you?' Explaining the predictions of any classifier. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, pp 1135–1144

Histopathology Breast Cancer Classification Using CNN



M. Venkateshwara Rao, Rajesh Saturi, D. Srinivas Goud, G. Srikanth Reddy, and N. Venkatesh

Abstract Breast cancer is the second leading cause of cancer mortality among women, early detection and treatment of breast cancer is very critical for the reduction of mortality rate**.** Cancer, known for its devastating impact, poses significant challenges in early detection within the medical field. In this study, breast cancer image samples were utilized for investigation purposes from BreakHis dataset. For accurate identification, current deep learning approaches relies on contour information. However, the cancer detection approach which is already existed retrieved only information which is semantic from the first layer and did not offer insights on the semantic levels which are not deep in breast cancer detection. Deep learning is a highly effective technique that possesses the capability to extract and structure significant insights from data, thus eliminating the necessity for domain experts to manually design feature extractors. Convolutional Neural Networks (CNNs), which belong to the category of deep, feed forward networks, have gained significant interest from researchers and industries. The recognition achieved by CNNs can be attributed to their integration in various domains such as transfer learning, speech recognition, natural language processing, signal processing, and object recognition. The proposed framework incorporates an effective and efficient strategy derived from a comprehensive review of commonly used approaches in each category. In the context of cancer screening, the process involves classifying image biopsies as either malignant or benign. Medical professionals like doctors and pathologists highly rely

M. Venkateshwara Rao (✉) · R. Saturi · D. Srinivas Goud · G. Srikanth Reddy
Department of CSE, Vignana Bharathi Institute of Technology, Ghatkesar, Hyderabad, India
e-mail: venkat2m2@gmail.com

R. Saturi
e-mail: rajesh.saturi@vbithyd.ac.in

G. Srikanth Reddy
e-mail: srikanth.gangidi@vbithyd.ac.in

N. Venkatesh
School of CS & AI, SR University, Ananthsagar, Hasanparthy (M), Warangal Urban, Telangana, India

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2023

S. Shakya et al. (eds.), *Fourth International Conference on Image Processing and Capsule Networks*, Lecture Notes in Networks and Systems 798,
https://doi.org/10.1007/978-981-99-7093-3_36

on the analysis of microscopic biopsy images to identify and categorize different abnormalities.

Keywords Histopathology images · Benign · Malignant machine learning · Deep learning · Convolutional neural network · Hematoxylin and Eosin (H&E)

1 Introduction

Cancer has become a significant global community health concern in recent times. A part of the World Health Organization (WHO), predicts nearly 10 million cancer-related deaths in 2020. Among the different forms of cancer, breast cancer (BC) is considered as the most commonly diagnosed type among women and results in a high mortality rate. Despite the advancements in understanding cancer biology and the identification of molecular markers, histopathological analysis remains as the effective diagnostic method.

> According to the report of IARC, breast cancer has a high mortality and morbidity rate among females. In India, breast cancer is considered as the second-largest chronic disease that affects around 300,000 people each year. Therefore, early-stage treatment and diagnosis can reduce the mortality rate, where the early diagnosis of breast cancer involves the detection of two major types of tumours: malignant and benign [1].

Breast cancer originates from breast tissue and is typically identified through the presence of a breast lump and certain deviations from normal conditions. Clinical screening for breast cancer involves techniques such as mammography, breast ultrasound, and other modalities. However, traditional manual diagnosis places a substantial burden on experts and is prone to diagnostic errors, impacting accuracy (Fig. 1).

> Breast cancer is the second most cause of cancer deaths in women after lung cancer. The death rate of women due to breast cancer is around 1 in 37 individuals that amount to 2.7% [1].

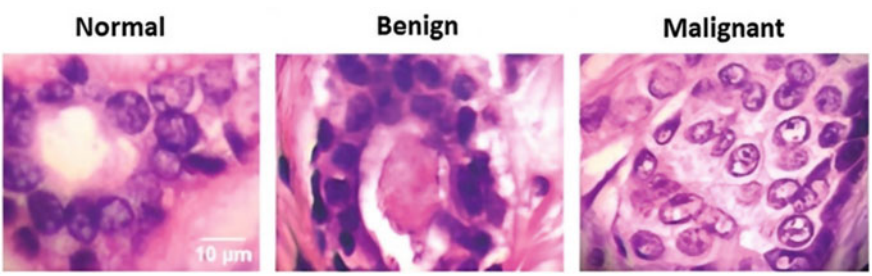


Fig. 1 Microscopic biopsy images

Deep learning, an evolving technology within machine learning, has attracted significant interest from researchers. Convolutional Neural Networks have demonstrated remarkable efficacy in large-scale image and video recognition tasks. As for example, the study conducted by Spanhol et al., the AlexNet architecture was utilized to successfully classify breast cancer pathology images, accurately differentiating between benign and malignant categories. Outperforming traditional machine learning algorithms with a 9% improvement in classification results. The methodology employed by the researchers resulted in an impressive average accuracy of 94% when classifying breast cancer cases images. Furthermore, Kausik et al. made a significant contribution by introducing an innovative framework known as Multiple Instance Learning (MIL) specifically designed for Convolution Neural Network (CNNs). This framework aims to improve the classification accuracy of pathological breast cancer images [2].

Nevertheless, the utilization of CNNs for the automatic categorization of pathological breast cancer images continues to pose significant challenges. One reason is the rapid growth in the number of CNN parameters as the model deepens, leading to a higher risk of overfitting. The acquisition of a substantial volume of labelled breast cancer images for training Convolutional Neural Networks (CNNs) is a costly endeavour, presenting a significant obstacle in the field [3].

2 Literature Survey

Rezk et al. [4] came up with a new data sampling approach for automatic histology breast cancer classification. At the start, the data sampling approach generates samples which is based on the data distribution of binary patterns. The developed sampling approach has helped to categorize the breast cancer histology image regions as cancerous or noncancerous. “In this literature, the performance of the developed sampling approach was compared with a few existing classical sampling approaches. From the obtained results, the developed model attained high performance compared to the existing approaches in terms of f-measure and accuracy.” “The major drawback of the data sampling approach is the exact matching between the images, sometimes it may lead to information loss” [1].

The outline information was used by the current deep learning techniques for breast cancer screening to deliver effective results. But because they didn’t make enough use of semantic details, the existing methods performed worse. Rajesh Saturi suggested a multi-objective feature selection strategy for the identifying of breast cancer in order to address of this problem. This method combines ACO with PSO. The BreakHis dataset is utilized in the suggested method along with the data augmentation was used to examine the random rotation and shearing transformation used to Combat the issue of over fitting [5].

Nahid et al. [6] came up with a Deep Neural Network (DNN) for diagnosis of breast cancer. In this study statistical and structural information extracted from the images collected from BreakHis dataset. “The performance of DNN was compared with a few

existing classifiers: CNN, SVM, and Long Short-Term Memory (LSTM). The simulation outcome showed that the DNN classifier attained better performance compared to other classifiers in the light of accuracy, precision, and f-measure. Generally, deep learning classifiers like DNN need an enormous amount of data for achieving better classification, which is a major concern in the applications like histopathology breast cancer detection" [6].

In this study, DNN uses the statistical and structural data produced from the gathered photos to categorize a collection of biological breast cancer images from the BreakHis database. In simpler terms, machine learning involves computers discovering valuable insights, while the DNN classifier outperformed alternative classifiers in terms of accuracy and precision and f-call measure. In order to get better classification, deep learning classifier like DNN typically require a huge amount of data [7].

A. *Machine Learning*

ML, categorized under the field of artificial intelligence (AI), enables software applications to enhance their predictive capabilities and make more precise forecasts without relying on explicit programming specifically designed for making those predictions. ML algorithms leverage historical and statistical data to predict new output values Fig. 2 [8]. While the discovery of new molecular markers is ongoing, histopathological analysis remains the most commonly utilized method for diagnosing cancer. Breast cancer, characterized by abnormal changes in breast tissue and the presence of a lump, is typically diagnosed using techniques such as mammography, breast ultrasound, and others. However, manual diagnosis places a significant burden on experts and is prone to diagnostic errors, affecting diagnostic accuracy [9]. Deep learning, a branch of machine learning, has attracted considerable attention from researchers. Convolutional Neural Networks (CNNs) have exhibited exceptional efficacy by recognizing the images and videos.

Spanhol et al. [10] conducted a study, where they successfully achieved the classification of breast cancer pathological images into benign and malignant categories

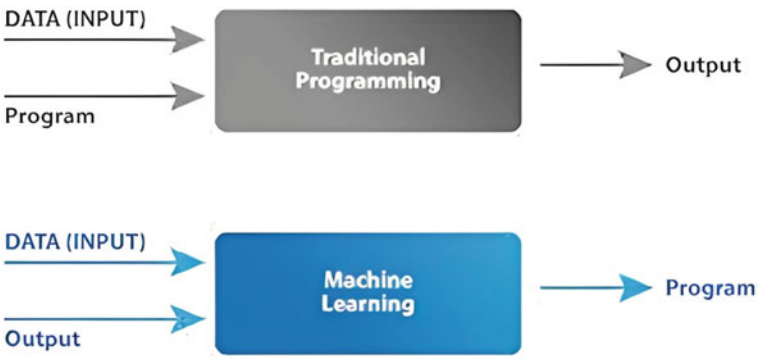


Fig. 2 Traditional machine learning structure

using AlexNet architecture. Machine learning algorithms learn from data in an iterative manner without explicit instructions on where to focus their search. As more data is provided, the algorithms gain a deeper understanding, leading to improved output quality. Machine learning increases the ability to autonomously and iteratively adapt to new data.

The process of machine learning is first initialized by providing training data to the selected algorithm, which employs pattern recognition to generate reliable and informed outcomes. The composition of the training data, whether it is already known or unknown holds significant importance in the development of a machine learning algorithm. Subsequently, new input data is introduced to assess the algorithm's performance and accuracy. By comparing the predictions with the actual results, adjustments are made to the algorithm through re-training by data scientists until the desired outcome is achieved. Through this iterative process, the machine learning algorithm gains the ability to learn autonomously and enhance its accuracy progressively as time progresses [11, 12].

B. ***Image Processing***

An image can be represented by a 2D function denoted as $f(x, y)$, where x and y correspond to the spatial coordinates. The intensity of the images at a given location is directly linked to the magnitude of F at the respective x and y values. The intensity of the image at a specific location is directly associated with the amplitude of F at the corresponding x and y values [13]. In the realm of digital images, the values of x, y, and amplitude are all finite. An image is essentially an array of pixels organized in rows and columns. Each pixel stores information about colour and intensity. Furthermore, by incorporating spatial coordinates x, y, and z, a three-dimensional representation of a photograph can be constructed. The process of pixelization employs a matrix-style layering technique, commonly known as an RGB image. It's worth noting that images come in various forms, such as RGB and Greyscale images [14, 15].

The field of image processing involves the implementation of diverse techniques on the image to improve its overall quality or extract valuable information. In this type of signal processing, image serves as the input, and either the image itself or its features are produced as the output. Image processing is a rapidly evolving technology that undergoes significant advancements. It holds great importance in the fields of engineering and computer science, making it a crucial topic of study [16].

3 Existing System

A traditional neural network, multilayer perceptron (MLP) is designed to mimic the human brain. MLPs have a number of shortcomings, particularly in terms of image processing. For each input (such as a pixel in a picture multiplied by 3 in the RGB case), MLPs use one perceptron. When dealing with large images, the sheer number of weights involved can become overwhelming. For instance, training a 224×224

pixel image with three colour channels require managing approximately 150,000 weights. As a result, training might be challenging, and overfitting can also occur [17].

4 Proposed System

In order to extend this, input includes the images from the dataset BreakHis. These are Hematoxylin and Eosin (H&E) stained images as they are gathered from different sources and put together to form this dataset. The dataset contains 400 images belonging to 2 different classes benign and malignant each of 100 images [18]. Input image will be pre-processed using different techniques like colour normalization and then the image is divided into patches. After that, the image will be passed through two trained CNN models. The output class of image is obtained by applying majority voting scheme to two CNN models. Output will be one of the two classes, benign and malignant.

5 Methodolgy

Step 1: Collecting data from BreakHis Dataset

The dataset utilized here consists of two types of microscopic biopsy images: one depicting breast cancer (referred to as Fig. 1) and the other representing noncancerous cells (referred to as Fig. 1). The dataset utilized in this research is comprised of 1693 microscopic images of breast tumour tissue obtained from the Breast Cancer Histopathological Database (BreakHis) refer Fig. 4. "These images were gathered from 82 patients and included in the database for further analysis. Among these images, 547 were classified as benign, representing noncancerous cells, while 1147 were classified as malignant, representing cancerous cells. The images in the dataset possess dimensions of 700 × 460 pixels and are stored in the PNG format. Additionally, they follow a 3-channel RGB configuration, with each channel having a 8-bit depth refer Table 1" [3].

Table 1 BreakHis dataset

Magnification	Benign	Malignant	Total
40X	652	1370	1995
100X	644	1437	2081
200X	623	1390	2013
400X	588	1232	1820
Total of Images	2480	5429	7909

The dataset is created with a collaborative effort from the Parana's P&D Laboratory—Pathological Anatomy and Cytopathology in Brazil. The dataset includes the samples obtained through the utilization of the surgical open biopsy (SOB) method, specifically a partial mastectomy or excisional biopsy technique [19]. "Unlike other types of needle biopsies, this procedure involves extracting a larger tissue sample and is performed in a hospital setting under general anaesthesia" [20].

A skin incision is done during an open biopsy to expose the tumour and enable for the cutting or scraping of a sample of tissue. Additional surgery could be done based on the results of the lab tests [21].

Step 2: Algorithm

Deep learning neural networks created specifically for the analysis of structured arrays of data-like representations. ConvNet does not use matrix multiplications, which is often how neural networks are thought of. It makes use of the convolutional approach refer Fig. 5 [22]. Convolution allows the combination of two functions to generate a third function that visually represents the transformation of one function's shape caused by the other refer Fig. 3. CNNs are efficient in identifying distinctive elements from input images, including lines, gradient rings, even eyes and face. Convolutional Neural Networks outperform in computer vision tasks due to its specific attribute function. CNNs have the capability to operate directly on lower-quality images without requiring pre-processing [23].

6 Implementation

In this paper the proposed model is implemented in anaconda platform, and python programming language is used to run this project.

7 Result

In accordance with the proposed experimental procedure, the BreakHis dataset has been partitioned into 70% training and 30% testing. To ensure the classifier's generalization to unobserved patients, the dataset was divided in such a way that the patients used to construct the training set were excluded from the creation of the testing set. The findings in this study reflect the average of five experiments. Each of the possible magnifications received its own unique application of this technique [24] .

Refer Table 2 shows the classification table for the proposed method in order to measure the two CNN models (0 and 1) based on the performance metrics like precision, recall, F1-score, and support.

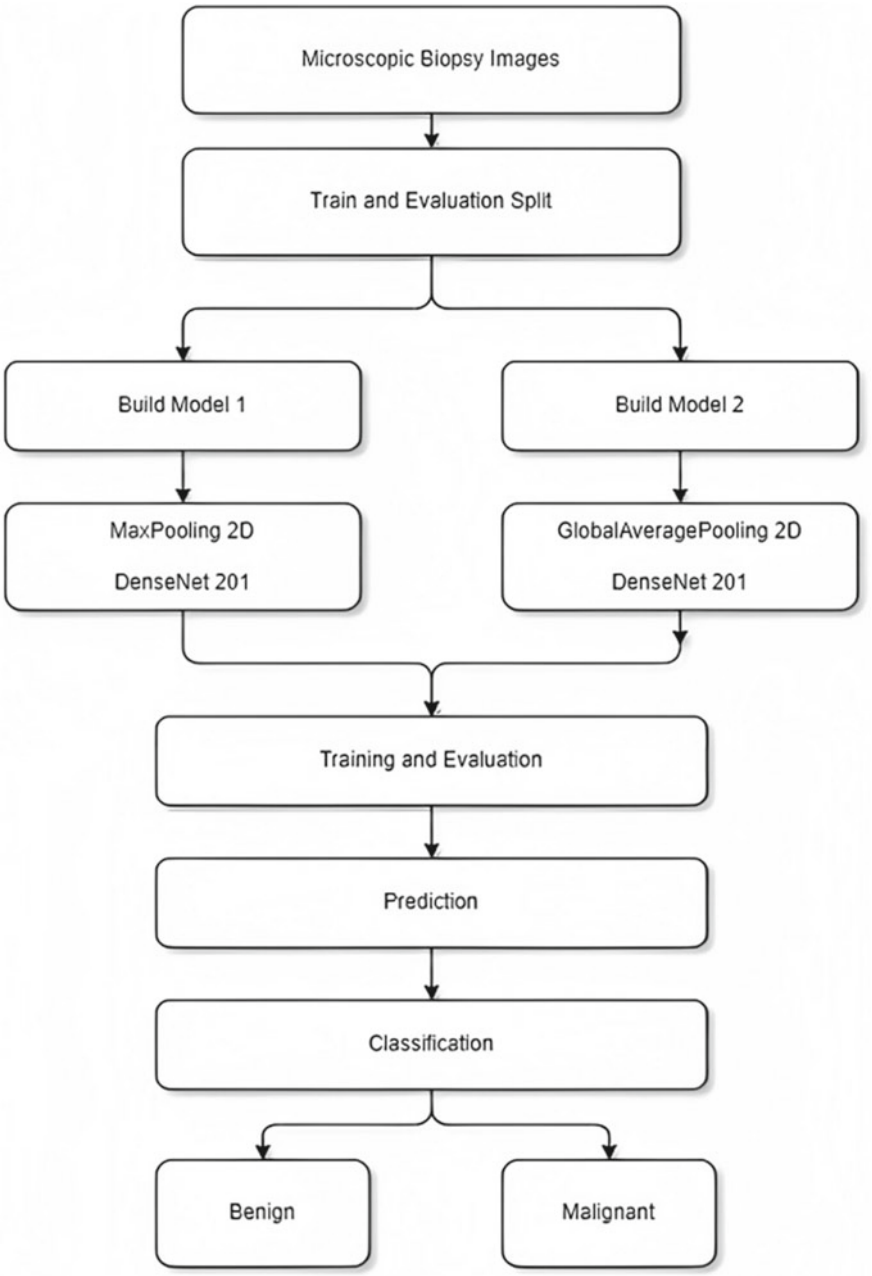


Fig. 3 Proposed method architecture

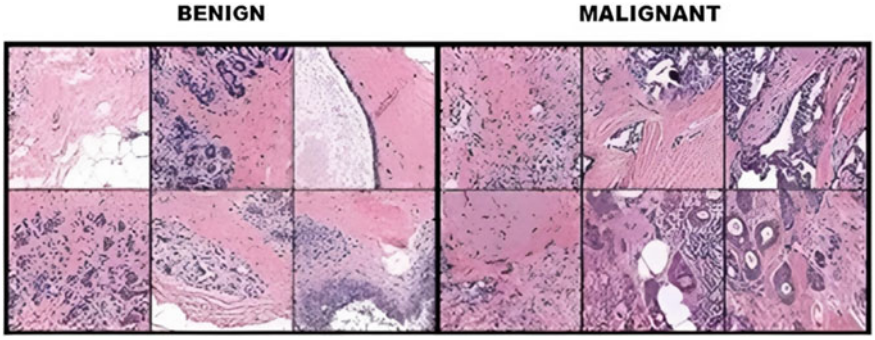


Fig. 4 Classification of cancerous tissue

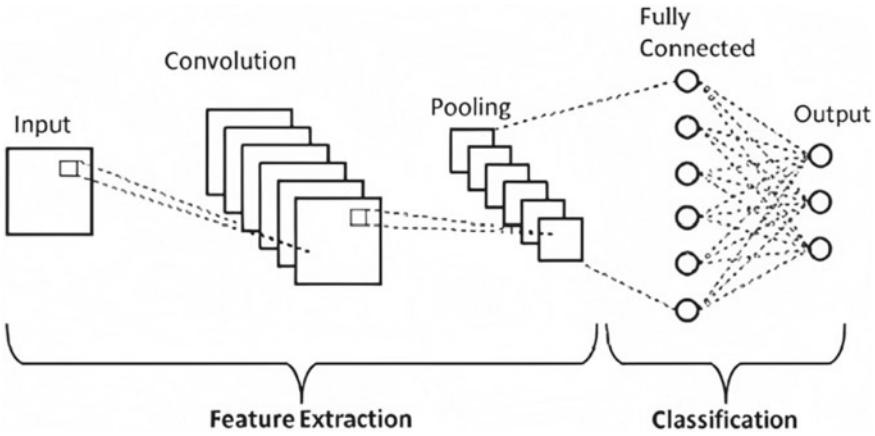


Fig. 5 CNN architecture

Table 2 Classification table

	Precision	Recall	F1-score	Support
0	0.97	0.73	0.83	176
1	0.89	0.99	0.93	369
Accuracy			0.91	545
Macro average	0.93	0.86	0.88	545
Weighted average	0.91	0.91	0.90	545

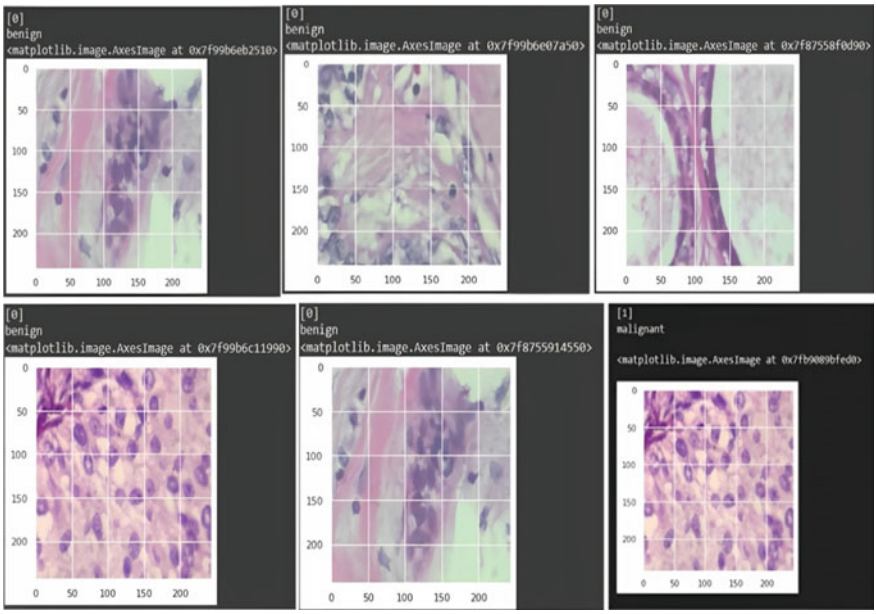


Fig. 6 Test model of the programme

8 Conclusion and Future Work

Developing automated breast cancer screening for improved patient care poses significant challenges. The main purpose of this learning was to evaluate and compare the effectiveness of two discrete Convolutional Neural Network (CNN) architectures in the detection of breast cancer. The dataset employed in this study includes 1693 microscopic images of breast tumour tissue, gathered from a cohort of 82 patients. Among these images, 547 were classified as benign, while 1147 were classified as malignant. The images were given in PNG format, with dimensions of 700 × 460 pixels and a 3-channel RGB configuration, each channel having an 8-bit depth. Employing CNN Model 2, the proposed system achieved an accuracy rate of 96%, outperforming CNN Model 1 with 93% accuracy refer Figs. 6 and 7. The primary goal of this research work is to provide support to healthcare professionals and oncologists in the precise diagnosis of breast cancer. Looking towards the future, the incorporation of artificial intelligence and machine learning into the diagnostic process has the potential to reduce human errors and enhance efficiency and facilitate early cancer detection, thereby empowering individuals in their Fig. ht against this disease, in future deep learning techniques can be applied to improve the accuracy of the proposed system with multi-objective functions.

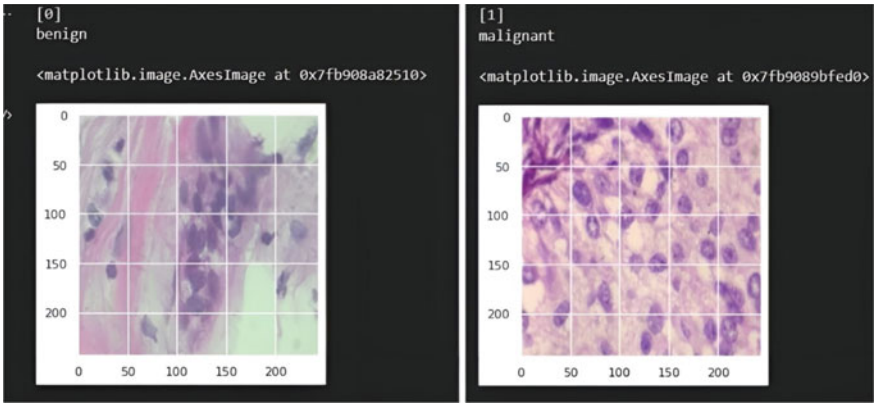


Fig. 7 Test model output

References

1. Saturi R, Parvataneni PC (2022) Histopathology breast cancer detection and classification using optimized superpixel clustering algorithm and support vector machine. Received: 22 Mar 2021. Accepted: 6 Apr 2022 The Institution of Engineers (India) 2022
2. Saturi R, Chand PP, A Novel variant optimized search algorithm for nuclei cell detection in histopathogy breast cancer images. Department of CSE, University College of Engineering, Osmania University, India
3. Saturi R, Implement an efficient segmentation method to overcome the issues of overlapping by segmenting the histopathological images of cancer by Research Scholar. Department of Computer Science & Engineering, University College of Engineering
4. Rezk E et al (2017) Conceptual data sampling for breast cancer histology image classification. Comput Biol Med 89:59–67
5. Saturi R, Premchand P, Multi-objective feature selection method by using ACO with PSO algorithm for breast cancer detection. Department of Computer Science and Engineering, University College of Engineering, Osmania University, India
6. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: 2016 IEEE, conference on computer vision and pattern recognition (CVPR). Las Vegas, NV, pp 770–778
7. Huang G, Liu Z, van der Maaten L, Weinberger KQ (2017) Densely connected convolutional networks. In: 2017 IEEE conference on computer vision and pattern recognition (CVPR). IEEE. pp 2261–2269. 10. 1109/CVPR.2017.243
8. Fawcett T (2006) An introduction to ROC analysis. Pattern Recogn Lett 27(8):861–874. https://doi.org/10.1016/j.patrec.2005.10.010
9. Golatkar A, Anand D, Sethi A, Classification of breast cancer histology using deep learning
10. Spanhol FA, Oliveira LS, Petitjean C, Heutte L (2016) Breastcancer histopathological image classification using convolutional neural networks. In: International joint conference on neural networks (IJCNN). IEEE, pp 2560–2567
11. Roy K, Banik D, Bhattacharjee D, Nasipuri M (2019) Patch- based system for classification of breast histology images using deep learning. Comput Med Imaging Graph 71:90–103. ISSN 0895-6111
12. Kowal M, Filipczuk P, Obuchowicz A, Korbicz J, Monczak R, Computer-aided diagnosis of breast cancer based on fine needle biopsy microscopic images
13. Ordi J, Castillo P, Saco A, del Pino M, Ordi O, Rodr´ıguez-Carunchio L, Ram´ırez J, Validation of whole slide imaging in the primary

14. Prasoon A, Petersen K, Igel C, Lauze F, Dam E, Nielsen M (2013) Deep feature learning for knee cartilage segmentation using a triplanar convolutional neural network. In: Medical image computing and computer assisted intervention—MICCAI 2013 ser, vol. 8150. Springer, Berlin, pp 246–253
15. George YM, Zayed HH, Roushdy MI, Elbagoury BM (2014) Remote computer-aided breast cancer detection and diagno
16. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems, pp 1097–1105
17. Schnitt, Elmore JG (2014) Digitized whole slides for breast pathology interpretation: Current practices and perceptions. J Digit Imaging 27(5), 642–648, Oct 2014
18. Janowczyk A, Madabhushi A (2016) Deep learning for digital pathology image analysis: A comprehensive tutorial with selected use cases. J. Pathol. Inf. 7
19. Araujo T, Aresta G, Castro E, Rouco J, Aguiar P, Eloy C, Pol´onia A, ´Campilho A (2017) Classification of breast cancer histology images using convolutional neural networks. PloS one 12(6):e0177544
20. Strigl D, Kofler K, Podlipnig S (2010) Performance and scalability of GPU-based convolutional neural networks. In: Parallel, distributed and network-based processing (PDP), 2010 18th Euromicro International Conference IEEE, 2010, pp 317–324. [15] LeCun Y, Bottou L, Bengio Y, Haffner P (1998) Gradientbased learning applied to document recognition. Proc IEEE 86(11):2278–2324
21. Duffner S (2008) Face image analysis with convolutional neural networks, Ph.D. dissertation. University of Freiburg Freiburg, Germany
22. LeCun Y, Kavukcuoglu K, Farabet C (2010) Convolutional networks and applications in vision. In: Proceeding of the 2010 IEEE international symposium circuits and systems (ISCAS), pp 253
23. He K, Zhang X, Ren S, Sun J (2015) Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: Proceedings of the IEEE international conference on computer vision, pp 1026–1034
24. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O et al (2011) Scikit-learn: machine learning in python. J Mach Learn Res 12:2825–2830

Automated Wave Height Measurement and Analysis Through Image Processing: A Computer Vision Approach for Oceanography



Medha Wyawahare, M. Selva Balan, Anvay Deshpande, Priyanka Acharya, Ankita Kumari, and Ashfan Khan

Abstract Wave height is an important parameter in ocean and coastal engineering. It is used to design coastal structures, such as breakwaters and seawalls and to predict the impact of waves on coastal areas, such as erosion and flooding. Traditional methods for measuring wave height, such as wave gauges are expensive and time-consuming to install and maintain. Image processing techniques offer a more cost-effective and efficient way to measure wave height. In this paper, a wave height detection algorithm based on image processing is presented. The algorithm uses a combination of edge detection and thresholding techniques to identify the wave crests in a sequence of images. The wave height is then calculated as the distance between the wave crests. The algorithm was tested on a dataset of images of ocean waves and was shown to be accurate and reliable.

Keywords Ocean wave · Wave height · Buoys · Centroid displacement · Image processing

Supported by CWPRS, Pune

M. Wyawahare (✉) · A. Deshpande (✉) · P. Acharya · A. Kumari · A. Khan
Vishwakarma Institute of Technology, Pune, Maharashtra, India
e-mail: medha.wyawahare@vit.edu

A. Deshpande
e-mail: anvay.deshpande20@vit.edu

P. Acharya
e-mail: priyanka.acharya20@vit.edu

A. Kumari
e-mail: ankita.kumari20@vit.edu

A. Khan
e-mail: ashfan.khan20@vit.edu

M. S. Balan
Central Water and Power Research Station, Khadakwasla, Pune, Maharashtra, India

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2023

S. Shakya et al. (eds.), *Fourth International Conference on Image Processing and Capsule Networks*, Lecture Notes in Networks and Systems 798,
https://doi.org/10.1007/978-981-99-7093-3_37

1 Introduction

Wave height detection refers to the process of determining the measurement of wave height in a body of water. This measurement holds significant importance in various fields, including oceanography, marine engineering, and coastal management. The height of waves provides valuable insights into the energy and momentum of ocean currents, aids in modeling coastal erosion and flooding, and facilitates the design of offshore structures. Traditionally, wave height has been assessed through the utilization of sensors such as wave buoys, which employ pressure sensors or accelerometers to gauge the pressure fluctuations caused by waves. Another approach involves using radar sensors to measure the distance between the water surface and a fixed point, and subsequently calculating the height of waves based on the observed changes in distance. However, the present techniques for measuring wave height have certain limitations in terms of accuracy, precision, cost, and maintenance.

In recent years, a new technology has been developed that offers improved accuracy and efficiency. This technology involves utilizing a single high-speed camera to capture continuous images of the water surface. By employing advanced image processing techniques, the wave height information can be extracted from these images.

Camera-based wave height detection has several advantages when compared with traditional methods. It is more cost-effective and low-maintenance than sensor-based approaches, while also offering higher spatial resolution and greater measurement accuracy. Furthermore, camera-based methods provide additional insights into wave shapes and dynamics, which prove valuable for comprehending and simulating ocean currents.

Overall, camera-based wave height detection represents an exciting new approach to measure wave height, with the potential to improve the understanding of ocean dynamics and inform coastal engineering projects.

2 Related Work

A study aims to validate using video for wave height estimation. A fully developed image processing algorithm is presented and tested through field studies. The system, called WavePack, uses a camera, acquisition unit, and processing unit to detect and estimate near shore wave height from digital video sequences. Results from testing at Narrow Neck Beach, Australia show that the system accurately estimates wave heights with a Pearson correlation coefficient of 0.86 [1].

One of the research uses Sentinel 1 Synthetic Aperture Radar (SAR) images and National Data Buoy Center (NDBC) buoy measurements. SAR images are processed through calibration, subset selection, and filtering to estimate effective dominant wavelength and significant wave height. NDBC buoy measurements are used to

validate the estimated values. The estimation result is better for the swell sea state than the wind wave-dominated sea surface [2].

The long-term automated monitoring of Nearshore Wave Height from Digital Video uses an algorithm that detects brightness variations in video frames to automatically monitor and measure wave heights in the nearshore area of the ocean. The brightness variations are caused by the movement of waves and are used to calculate wave height and period. This non-intrusive method can be used in any weather condition and calibrated to account for factors such as camera tilt, atmospheric distortion, and camera position relative to the ocean surface [3].

The author has used a convolutional neural network (CNN)-based classification model to estimate wave height levels from a single ocean image and a regression model for sequential images. ResNet was found to be the best convolutional neural network model for ocean image processing due to its high accuracy, short training time, and balanced correction rate. The paper proposed a bidirectional ConvLSTM (long short-term memory)-based regression model to estimate real-valued significant wave heights from sequential ocean images. The model extracted spatio-temporal features using VGGNet and ConvLSTM, and achieved low error rates in terms of mean absolute error and mean absolute percentage error [4].

The researchers first used convolutional neural network (CNN) to extract features from the images. The extracted features show the change in wave heights which was useful for long short-term memory (LSTM). LSTM is used for wave height prediction and analysis has been done on the proposed convolutional neural network-long short-term memory (CNN-LSTM)-based approach with other algorithms such as LSTM, linear regression and support vector machine (SVM) [5].

The authors have measured the dynamic wave height of tuned liquid column dampers (TLCDs) using a digital camcorder for recording the video of a laterally shaking TLCD with blue dye applied to highlight the liquid. The vision-based sensing method's image processing algorithm speeds up the assessment of wave height by simply counting white pixels in a binary image. Edge detection and moving object tracking are two sequential image processing techniques that have been applied [6].

The authors have presented a framework for predicting wave conditions using machine learning models as a substitute for the industry-standard Simulating Waves Nearshore (SWAN) tool. The authors utilized a multi-layer perceptron (MLP) regression to replicate significant wave height generated by SWAN, and a support vector machine (SVM) to classify wave periods. The models are over 4000 times faster than SWAN, while still producing similarly accurate results within the targeted domain [7].

The authors have presented a backpropagation (BP) neural network model optimized by mind evolutionary algorithm-backpropagation (MEA-BP) for predicting ocean wave heights. Wind speed value (UV) and mean sea level pressure (MSL) are selected as input features to predict ocean wave height. The mind evolutionary algorithm is employed to optimize the BP neural network model. The results show that the (MEA-BP) model outperforms existing methods such as genetic algorithm-backpropagation (GA-BP) and standard backpropagation (SBP) in predicting ocean wave heights, achieving high accuracy and correlation coefficients [8].

The authors have trained a model which was able to predict the wave parameters of the next one hour with the help of long short-term memory (LSTM). The learning algorithm, stochastic gradient descent (SGD) is employed when there is a need to calculate the local minima of a function whereas adaptive moment estimation (ADAM) is an extensive version of stochastic gradient descent (SGD). The accuracy of the stochastic gradient descent (SGD) algorithm was better when compared with the adaptive moment estimation (ADAM) algorithm and was able to predict with an error of less than 20 [9].

The past data has been used in order to train a model using various sine functions or better known as multiple sine functions decomposition (MSFD) The authors have calculated the coefficients of the resultant function from training data and substituted their values into sine functions. The proposed model has a very high learning ability and most of the errors were concentrated in the range of −0.4 to 0.2 m [10].

This study aims to predict the significant wave height through machine learning (ML), a soft computing method. The feedforward neural network (FNN) and long short-term memory (LSTM) models are adopted to predict significant wave height. Input parameters for the input layer were selected by Pearson correlation coefficients. To obtain the optimized hyperparameter, a sensitivity study is conducted on the window size, node, layer, and activation function. The results showed that the most suitable model for predicting the significant wave height was feedforward neural network (FNN-W48) owing to its accuracy and calculation time [11].

One of the study proposes the use of structural acceleration data for improving the significant wave height prediction accuracy of multiclass classification models. The authors have used artificial neural network and decision tree models. It is also demonstrated that redundancy of training parameters could inject unwanted overfitting, reducing model generalization. Precautions were also taken like train-test splits and K-fold validation for both training artificial neural networks and decision tree ensembles [12].

3 Proposed Work

The ball detection algorithm works by first converting the video frame to the hue, saturation, value (HSV) color space. This is because the HSV color space is more intuitive for humans to understand and work with than the BGR color space. In HSV, each pixel is represented by three values: hue (H), saturation (S), and value (V). Hue refers to the color of the pixel, saturation refers to the purity of the color, and value refers to the brightness of the pixel. The cvtColor function in OpenCV takes the video frame as input and converts it from BGR to HSV color space. This transformation allows the ball detection algorithm to focus solely on the color characteristics of the ball, while ignoring brightness variations in the video frame (see Fig. 1).

Once the frame has been converted to HSV, a mask is created based on the lower and upper boundaries of the ball color range. The mask is a binary image, where white pixels represent pixels that are within the ball color range and black pixels

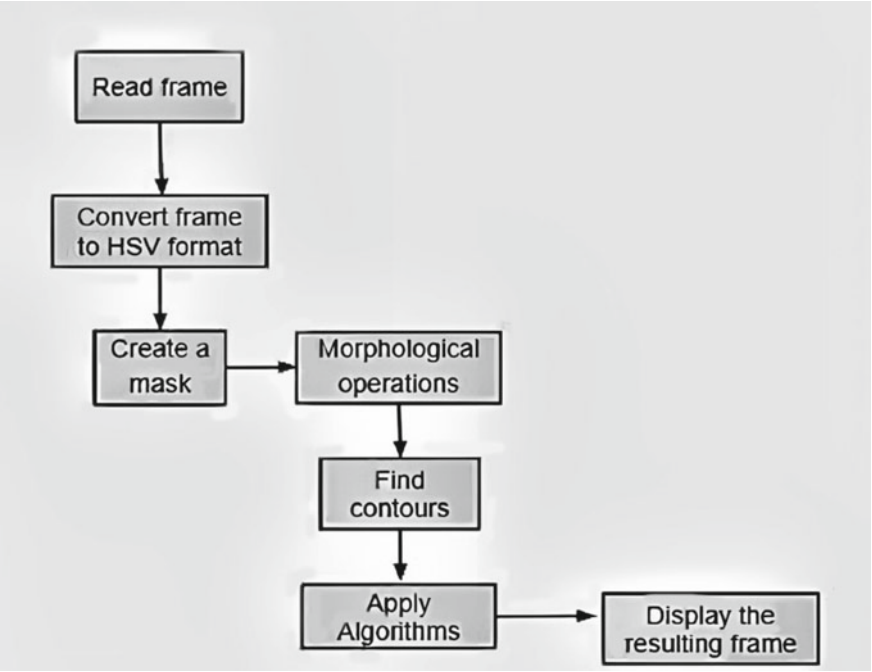


Fig. 1 Showing proposed methodology

represent pixels that are outside the ball color range. The mask is then applied to the frame to remove any pixels that are not within the ball color range. This leaves a binary image that only contains the ball.

Next, morphological operations are applied to the mask to remove any noise. Morphological operations are a set of image processing techniques that can be used to erode, dilate, open, close, and smooth images. In this case, we use morphological operations to remove any small, insignificant objects from the mask. Finally, contours are found in the mask. Contours are the boundaries of connected components in the image. In this case, contours represent the outline of the ball. The process of finding contours involves several steps.

1. The input image is typically converted to grayscale to simplify the analysis and reduce computational complexity. Grayscale images only contain intensity values, which allows for easier thresholding and segmentation.
2. Thresholding is applied to convert the grayscale image into a binary image. Thresholding is the process of dividing the image into foreground and background by selecting an appropriate intensity threshold.
3. Once the image is thresholded, the findContours function is used to identify and extract the contours. The findContours function takes the binary image as input and returns a list of contours found in the image. The contours are represented as

a sequence of points that form the boundary of connected regions in the binary image.

By finding contours, the location and shape of the ball within the frame can be identified. Once the contours have been found, the wave height can be calculated. Several methods exist for calculating wave height, but the most common one is to use the distance between the highest and lowest points of the ball contour to determine the height with precision.

By subtracting the y-coordinate of the previous center (prev center [1]) from the y-coordinate of the current center (center [1]), the vertical movement (upward displacement) of the yellow ball in the image is obtained.

The expression for calculating the upward displacement (vertical movement) of the yellow ball is given by:

$$y\,\text{diff} = \text{center}[1] - \text{prev center}[1].$$

The result '*y* diff' will be positive if the ball moves upward, negative if it moves downward, and zero if there is no vertical movement.

4 Algorithm Implementation

The selection of a specific algorithm for wave height detection depends on various factors, including the type of data available, the complexity of the wave patterns, and the desired level of accuracy. For video-based wave height detection, image processing algorithms like contour detection, thresholding, and morphological operations are commonly used. These algorithms are effective in identifying the boundaries of the waves in the video frames and calculating the wave height based on the displacement of the wave contours. For sensor-based wave height detection, signal processing techniques like Fourier analysis and spectral analysis are often employed. These algorithms can extract wave height information from the sensor data and provide accurate measurements. The choice of algorithm also depends on computational efficiency and real-time processing requirements. Overall, selecting the appropriate algorithm is crucial to ensure accurate and reliable wave height detection in various environmental conditions.

Algorithm 1: Thresholding

Loop through each contour:

1. Calculate the current position of the ball.
2. Having a previous position, calculate the upward displacement.
3. If the displacement is positive, the ball is moving upward.
4. Update the previous position to the current position.

4.1 Algorithm 2: Center Displacement

1. Calculate the yellow ball center by taking the midpoint of the bounding rectangle.
2. Compute the vertical movement difference of the yellow ball by comparing the current center with the previous center.
3. Update the previous center and previous y-coordinate values for each iteration.
4. Draw a circle at the centroid of the yellow ball and a bounding rectangle on the original frame.

4.2 Algorithm 3: Y-coordinate and Radius Difference

1. If circles are detected (circles are not none), iterate over each detected circle using a for loop.
2. Calculate the height of the wave based on the circle's position by subtracting the radius from the y-coordinate and multiplying it by the PIXEL-TO-MM-RATIO.
3. Create text to display the wave height, formatted to two decimal places, using f-string notation.
4. Add the wave height text above the circle.

4.3 Algorithm 4: Centroid Method

1. Set the PIXEL-TO-MM-RATIO to the desired value, representing the conversion factor from pixels to millimeters.
2. Set the NUM-FRAMES to the desired number of frames to use for calculating the wave height.
3. Calculate the centroid of each valid contour. If the moment m00 is nonzero (to avoid division by zero), calculate the x and y coordinates of the centroid.
4. Draw a circle at the centroid position on the frame using cv2.circle.
5. Add the centroid (x, y) coordinates to the centroids list.
6. Keep only the last NUM-FRAMES centroids in the list to maintain a sliding window of centroids.

4.4 Algorithm 5: Optimal Reference Point

1. The largest greenish yellow contour is estimated by assuming that the maximum greenish yellow area is occupied by the buoy in one frame. This largest contour will eliminate the external noises of greenish yellow color (e.g., Trees and other small reflections) but it will still contain the reflection of the buoy in the water.

2. The buoy is detected along with its reflection. The top most pixel of the ball is selected as the reference point which will eventually eliminate the reflection of the ball.
3. The maximum and minimum y-coordinates reached by the buoy are estimated by tracking the vertical motion of the reference point.
4. The wave height is estimated to be the difference between maximum and minimum vertical displacement of the reference point.

5 Results

Based on the obtained data from Table 1, Algorithm 1 (see Fig. 2a) demonstrates the widest height range, supporting measurements from 0.11338 to 2.4 mm. This algorithm is well-suited for applications that require a broader range of height measurements. On the other hand, Algorithm 4 (see Fig. 2d) offers the narrowest range, limited to measurements between 0.1 and 0.14 mm. This algorithm can be more applicable in scenarios where precision within a specific range is essential.

Algorithm 2 (see Fig. 2b) provides a moderate height range, with measurements ranging from 0.1 to 1.2 mm. It strikes a balance between the wider range of Algorithm 1 and the narrower range of Algorithm 4. Algorithm 3 (see Fig. 2c) offers a relatively compact height range from 1.36 to 1.62 mm, suitable for applications requiring precise measurements within this specific interval.

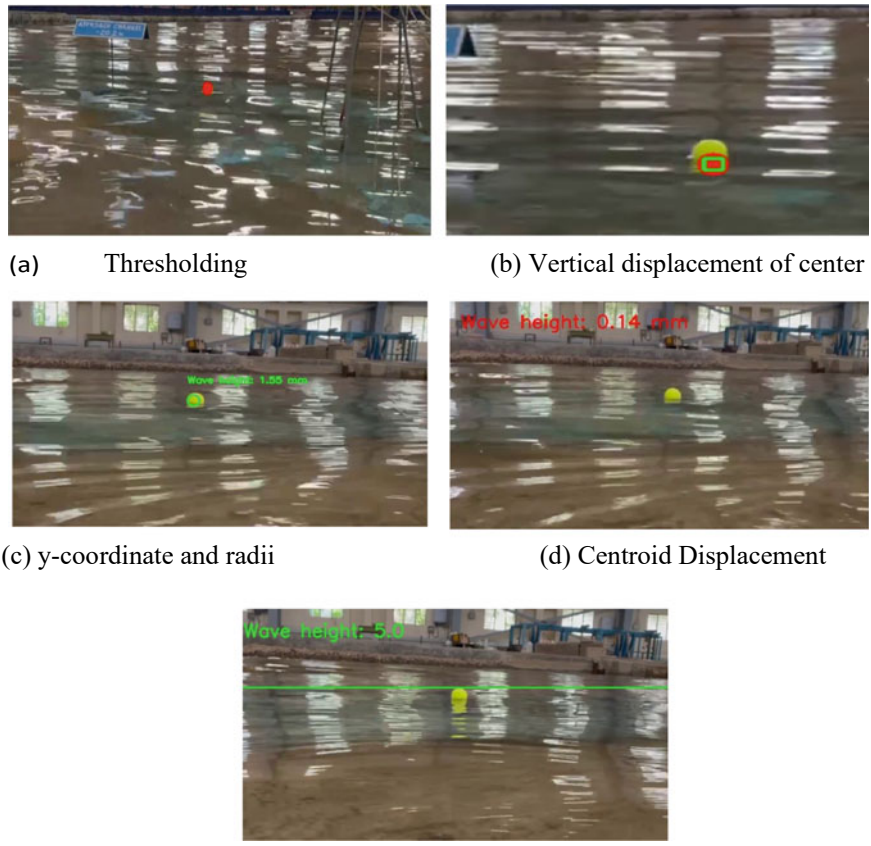
Algorithm 5 (see Fig. 2e) provides the smallest range among the algorithms analyzed, supporting heights between 0.02 and 0.18 mm. This algorithm is ideal for applications that necessitate fine-grained height measurements within a limited range.

The specifications for the video capturing, processing, and data source:

1. **Camera Model**: The specific camera model used for video capturing is the ZED-4 camera. The ZED-4 camera is known for its high-resolution imaging capabilities and depth sensing features, making it suitable for computer vision and 3D reconstruction applications.
2. **Image Processing Libraries**: For postprocessing and analysis of the captured videos, image processing libraries and tools like OpenCV may have been utilized.

Table 1 Showing calculated wave heights and execution time of the algorithms

Algorithm	Min. height (in mm)	Max. height (in mm)	Execution time (in sec)
Thresholding	0.113388	2.4	5.927
Center displacement	0.1	1.2	5.753
y-coordinate and radius diff	1.36	1.62	4.558
Centroid method	0.1	0.14	5.350
Optimal reference point	0.02	0.18	5.642



(a) Thresholding (b) Vertical displacement of center

(c) y-coordinate and radii (d) Centroid Displacement

(e) Line passing through reference point

Fig. 2 Showing the obtained results with proposed algorithms

3. **Data Source**: For automatic wave height prediction, the data has been collected from the Central Water and Power Research Station (CWPRS, 2023). The videos and images were captured from the active model at CWPRS through ZED-4 camera.

These software specifications and tools are vital in conducting video capturing, image processing, and subsequent analysis for the research study using the ZED-4 stereo camera and still camera. The cameras ensure accurate data acquisition and facilitate effective evaluation and interpretation of the results obtained from the captured video footage.

6 Limitations and Discussion

The discussed studies and algorithms provide a diverse range of approaches to wave height estimation using various techniques such as image processing, machine learning, and sensor data analysis. Each algorithm has its own strengths and limitations, which can impact the accuracy and applicability of wave height measurements in different scenarios.

For instance, Algorithm 1 utilizes a simple thresholding method for ball detection, which may not be accurate for small or fast-moving objects like waves. However, the algorithm used in WavePack [1] demonstrates promising results in accurately estimating wave heights from digital video sequences. The use of a camera, acquisition unit, and processing unit in the WavePack system enables accurate wave height estimation, as evidenced by the high Pearson correlation coefficient of 0.86.

Similarly, Algorithm 2 faces challenges with complex backgrounds, occlusions, and changes in ball appearance. These limitations can affect the algorithm's ability to accurately track wave crests and estimate wave heights. On the other hand, the utilization of Sentinel 1 Synthetic Aperture Radar images and NDBC buoy measurements in [2] provides a complementary approach to wave height estimation. By processing Synthetic Aperture Radar images and validating the results with buoy measurements, the algorithm achieves better estimation accuracy for certain sea states.

Algorithm 3's reliance on a single reference point assumption for wave height calculation can be a limitation in capturing complex wave motion with multiple peaks or different reference points. However, the approach presented in [3] that uses brightness variations in video frames to automatically monitor and measure wave heights in the nearshore area of the ocean offers a non-intrusive method that can be used in any weather condition.

The limitation of Algorithm 4 is that it assumes a consistent and smooth trajectory of the ball over the last NUM-FRAMES. If there are sudden changes or fluctuations in the ball's position or trajectory within those frames, it can lead to inaccurate wave height calculations, whereas in method 5 the buoy is detected by extracting the largest greenish yellow area; this algorithm is based on the assumption that the largest greenish yellow colored area on the frame is of the buoy itself.

The most significant limitations in algorithms used for wave height detection are:

1. **Sensitivity to noise**: The algorithms are sensitive to noise in the image. This can lead to errors in the wave height measurements.
2. **Difficulty in identifying the wave crest**: The algorithms can have difficulty in identifying the wave crest. This can lead to errors in the wave height measurements.
3. **Computational complexity**: The algorithms can be computationally complex. This can make them difficult to implement in real-time applications.

Despite these limitations, the algorithms can be used to obtain useful information about wave conditions. The algorithms can be used to monitor wave height, which can be used to assess the risk of coastal flooding and erosion. The algorithms can

also be used to study the behavior of waves, which can be used to improve the design of ships and offshore structures.

7 Conclusion and Future Scope

In this study, the performance of five different algorithms for wave height detection was investigated. The algorithms were tested on a dataset of synthetic and real-world wave data. The results showed that all five algorithms were able to detect wave heights with good accuracy. However, there were some differences in the performance of the algorithms under different conditions. The results of the study showed that there is no single algorithm that is best for all wave height detection applications. The choice of algorithm will depend on the specific application and the conditions under which the algorithm will be used.

Future research can focus on expanding the method to work with multiple buoys that could enable the monitoring of wave patterns across larger areas of the ocean. Implementing a network of buoys with synchronized cameras would provide a comprehensive understanding of wave behavior over vast regions, benefiting coastal management and marine engineering. Integration with remote sensing technologies, such as radar could provide a comprehensive and multimodal approach to wave measurement.

References

1. Eason G, Noble B, Sneddon IN (1955) On certain integrals of Lipschitz-Hankel type involving products of Bessel functions. Phil Trans Roy Soc London A247:529–551
2. Pramudya FS, Pan J, Devlin AT (2019) Estimation of significant wave height of near-range traveling ocean waves using Sentinel-1 SAR images. IEEE J Sel Top Appl Earth Obs Remote Sens 12(4):1067–1075
3. Gal Y, Browne M, Lane C (2013) Long-term automated monitoring of nearshore wave height from digital video. IEEE Trans Geosci Remote Sens 52(6):3412–3420
4. Choi H, Park M, Son G, Jeong J, Park J, Mo K, Kang P (2020) Realtime significant wave height estimation from raw ocean images based on 2D and 3D deep neural networks. Ocean Eng 201:107129
5. Guan X (2020) Wave height prediction based on CNN-LSTM. In: 2020 2nd international conference on machine learning, big data and business intelligence (MLBDBI). IEEE, pp 10–17
6. Kim J, Park C-S, Min K-W (2016) Fast vision-based wave height measurement for dynamic characterization of tuned liquid column dampers. Measurement 89:189–196
7. James SC, Zhang Y, O'Donncha F (2018) A machine learning framework to forecast wave conditions. Coast Eng 137:1–10
8. Wang W, Tang R, Li C, Liu P, Luo L (2018) A BP neural network model optimized by mind evolutionary algorithm for predicting the ocean wave heights. Ocean Eng 162:98–107
9. Osawa K, Yamaguchi H, Umair M, Hashmani MA, Horio K (2020) Wave height and peak wave period prediction using recurrent neural networks. In: 2020 international conference on computational intelligence (ICCI). IEEE, pp 1–4

10. Wang H, Fu D, Liao S, Wang G, Xiao X (2019) Significant wave height prediction based on MSFD neural network. In: 2019 tenth international conference on intelligent control and information processing (ICICIP). IEEE, pp 1–5
11. Park SB, Shin SY, Jung KH, Lee BG (2021) Prediction of significant wave height in Korea strait using machine learning. J Ocean Eng Technol 35(5):336–346
12. Demetriou D, Michailides C, Papanastasiou G, Onoufriou T (2021) Coastal zone significant wave height prediction by supervised machine learning classification algorithms. Ocean Eng 221:108592

Enhanced Feature Fusion from Dual Attention Paths Using Feature Gating Mechanism for Scene Categorization of Aerial Images



G. Akila and R. Gayathri

Abstract The widespread use of remote sensing (RS) aerial images in scene categorization has drawn the attention of numerous researchers. As the consequence of this advancement, the approaches for RS image scene categorization that use convolutional neural networks (CNNs), have progressed enormously. Majority of the layouts in the works already published only took the account of scene's global information. However, a lot of replicated areas in the global information reduce the effectiveness of classification and disregard local contextual details, which reflect finer spatial characteristics of local objects. Additionally, the majority of CNN-based approaches give the identical weights on each feature vector, which prevents the mode from differentiating the important features. To overcome the above limitations, we propose a model called enhanced feature fusion from dual attention paths using feature gating mechanism (EFFDA-FG). This model employs Resnext50 baseline model, with channel spatial attention for deriving global and local features along with feature fusion using a gating mechanism. Our suggested model was implemented on three well-known datasets and it proves to give accuracy equal to or higher than that of many state-of-art CNN architectures while using comparatively lesser parameters.

Keywords Aerial images · Convolutional neural networks (CNN) · Channel attention · Spatial attention · Feature-gating

G. Akila (✉)
Department of Electronics and Communication Engineering, Meenakshi Sundararajan Engineering College, Chennai 600024, India
e-mail: aki47g@gmail.com

R. Gayathri
Department of Electronics and Communication Engineering, Sri Venkateswara College Of Engineering, Sriperumbudur 602117, India
e-mail: rgayathri@svce.ac.in

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2023

S. Shakya et al. (eds.), *Fourth International Conference on Image Processing and Capsule Networks*, Lecture Notes in Networks and Systems 798,
https://doi.org/10.1007/978-981-99-7093-3_38

1 Introduction

Image classification has undergone significant research in recent years because of the vast amounts of remote sensing (RS) data that have been produced [1]. Research on RS-based aerial scene classification, which aims to categorize the acquired image semantic level data with the proper category labels, has also evolved. There are two key reasons why things are difficult in this situation: (1) intra-class diversity, where scenes from the same class of images vary greatly in attributes like color and shape, and (2) inter-class similarity, where scenes from several image classes have textures that are similar to one another. Therefore, learning how to extract useful features from complex images is crucial to increasing classification accuracy. The three main categories of feature extraction techniques are convolutional neural networks (CNNs), vision transformers (VIT), and manual feature-based techniques. Manual feature-based techniques might design a single feature or combine several features to improve the results of categorization [2]. However, when complex content is used as features, the capacity to describe scenes is limited. Global characteristics of pictures are typically learned for classification by VIT-based algorithms [3–5]. Local feature improvement, however, has not made much progress.

CNN was recently introduced to the RSSC industry, where it has achieved success in terms of categorization [6, 7]. By learning important local regions in the scene, Bi et al. [8] built a local semantic-augmented convolution network to build discriminative local semantic representation. In order to successfully fuse features, Wang et al. [9] created a multilevel feature fusion network and an adaptive technique for channel downscaling of high-dimensional features. Traditional feature extraction techniques like principal component analysis, sparse coding, and histograms of directed gradients have a limited capacity for feature representation. Convolutional neural networks (CNNs), in particular, have made significant advancements in the field of RS scene classification in recent years. To enhance the performance of the model, Li et al. [10] developed a joint pooling method and integrated it into a bilinear CNN. For scene categorization, Xu et al. [11] suggested a multilayer feature fusion network (MF2Net) to efficiently combine multilayer convolutional features. In order to understand the underlying hierarchical structures of scene images and lessen the visual-semantic disparity, Wang et al. [12] devised a multi-granularity canonical appearance pooling (MG-CAP). CNN-based methods have demonstrated more potential for extracting robust features in RS scene photos when compared with conventional feature extraction techniques. However, as seen in Fig. 1, the feature representation capability of models is still insufficient because RS scene photos frequently have complicated backgrounds and dispersed little items.

A number of attention techniques have been put out in recent years to learn distinguishing aspects for RS scene classification. To extract potent and multilayer feature representations, Wang et al. [13] investigate a relation-attention network (RANet). Zhao et al. [14] created an enhanced attention module (EAM) based on the convolutional block attention module (CBAM) to infer both channel and spatial attention maps. Convolutional local attention module (CLAM) was developed by

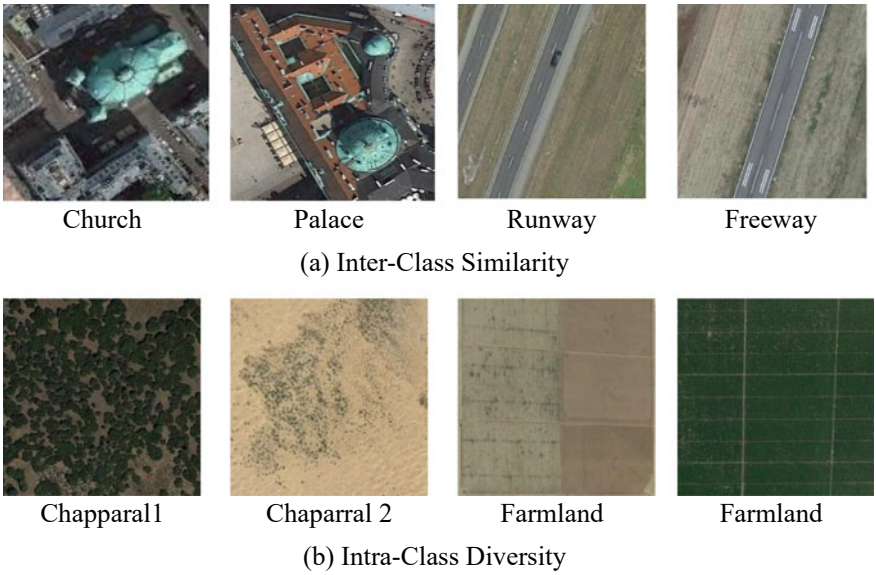


Fig. 1 Examples scene images for inter-class similarity and intra-class diversity from NWPU-RESISC45 dataset

Chen et al. [15] and included into the ResNet50 [16]. These techniques, however, have two issues. First, the model's ability to concentrate on more important regions is hampered by an incomplete consideration of the long-range reliance and local information. Second, it's not always ideal to integrate attention maps or modules.

A unique enhanced feature fusion from dual attention paths using feature gating mechanism (EFFDA-FG) is suggested to address the aforementioned issues. The following is a summary of these notable contributions:

- An end-to-end network called EFFDA-FG is suggested as a ResNext50-based approach to learn discriminative features for aerial scene categorization.
- In this case, the first as well as final sections of the fifth convolutional module of ResNext50 are used to extract two sets of deep features. On the first feature, channel spatial attention is performed, and on the last feature, channel attention is performed.
- The obtained features are combined using a enhanced feature fusion that involves feature gating and element-wise multiplication to extract the final intricate features for robust and efficient classification.

2 Methodology

2.1 *Proposed Architecture*

The overall architecture of EFFDA-FG can be viewed from Fig. 2.

As can be seen, features from ResNext50's final stage are sent through an attention network, a feature gating mechanism and a multiplication layer to extract more significant details from the features. The following is a description of the architecture.

The input images are fed to the baseline architecture ResNext50 which consists of five convolutional sections as shown in Fig. 2a. The last section consists of high semantic level features that helps in classification. The last section has three convolutional blocks. Output features are taken from the first and last blocks of this section.

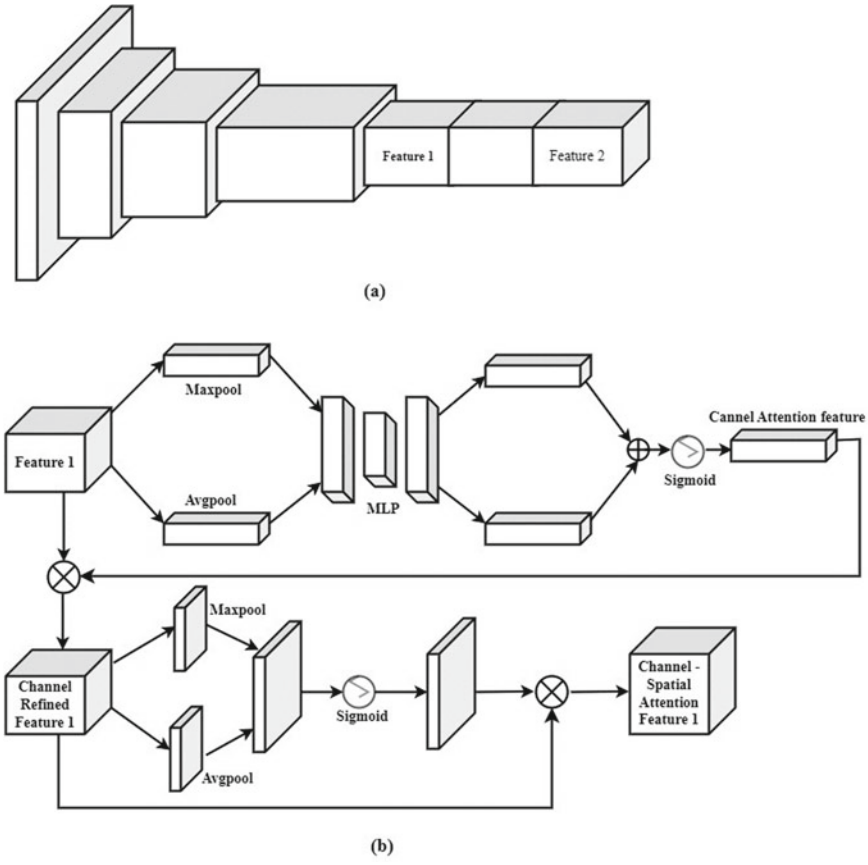


Fig. 2 Overall architecture of EFFDA-FG, **a** Feature 1 and 2 extractions from Resnext50 backline architecture, **b** Channel-spatial attention on feature 1, **c** Channel attention on feature 2, and **d** Feature gating and feature fusion for final classification

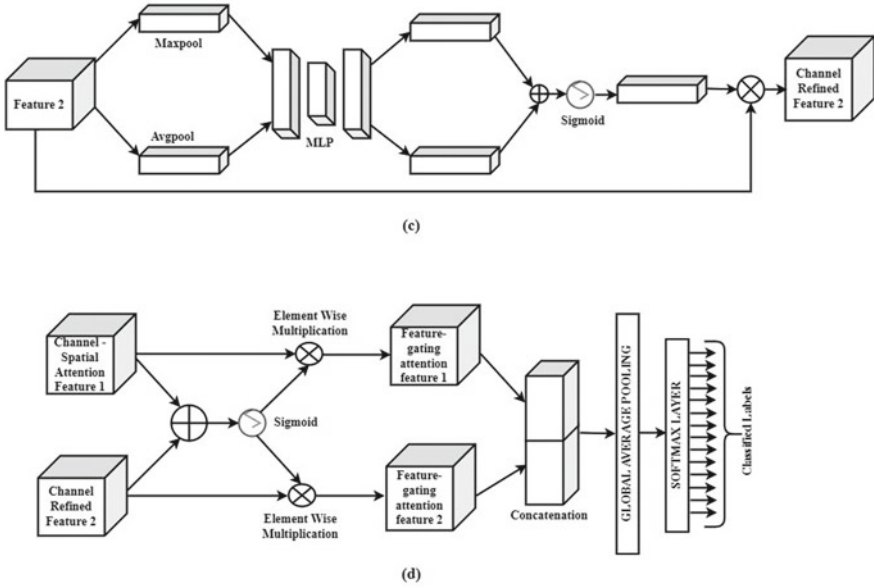


Fig. 2 (continued)

The output from first convolutional block is fed to channel-spatial attention network as it is an intermediate level that may also contain spatial features as shown in Fig. 2b. The output from the last convolutional block is fed to channel attention network as depicted in Fig. 2c.

The output from these attention networks is added and then passed through the sigmoid layer and then multiplied with their respective input features as shown in Fig. 2d. This forms the feature gating mechanism that extracts the final intricate features for robust and efficient classification. Each phase of the model is explained in detail.

ResNeXt50. The ResNeXt50 has become widely favored and influential in computer vision due to a multitude of compelling factors driving its adoption. One prominent reason for selecting ResNeXt50 is its improved capacity compared with the original ResNet50 architecture. By incorporating a cardinality parameter, it enhances expressiveness and model capacity, enabling it to capture intricate patterns and relationships effectively. Scalability is another key advantage of ResNeXt50. With its modular and flexible design, it can easily adapt to different depths and widths, making it suitable for various tasks and datasets. This adaptability empowers researchers and practitioners to customize it to meet specific requirements, allowing for efficient modeling of complex tasks. Efficiency is a critical consideration, and this network excels in this aspect. Despite its increased capacity, it maintains computational efficiency by utilizing grouped convolutions, reducing computational costs while preserving strong representational power. Consequently, it offers an efficient solution

for resource-constrained environments. It leverages transfer learning by pretraining on large-scale datasets like ImageNet. This pretraining provides a solid foundation of learned features, facilitating improved generalization and faster convergence when fine-tuning on specific tasks or smaller datasets. This capability is particularly valuable when limited training data is available. ResNeXt50 strikes a balance between expressiveness, computational efficiency, scalability, and exceptional performance, making it a compelling option for a broad range of computer vision applications.

Convolutional Block Attention Module (CBAM). The CBAM [17] module consists of two components: the channel attention module (CAM) and the spatial attention module (SAM).

Channel Attention Module (CAM): The channel attention module aims to capture channel-wise dependencies and emphasize informative channels. It operates as follows: Given an input feature map X of shape (N, H, W, C), where N represents the batch size, H and W represent the spatial dimensions, and C represents the number of channels:

1. Global Average Pooling (GAP): Apply global average pooling across the spatial dimensions of X to obtain channel-wise statistics. This is achieved by calculating the mean value for each channel:

$$S_C = \frac{1}{H * W} \sum X_{ijc}; \text{for } i \text{ in range}(H),\ j \text{ in range}(W). \tag{1}$$

2. Channel Max Pooling (CMP): Apply max pooling across the channel dimension of X to capture the most prominent features in each channel. This is done by calculating the maximum value for each channel:

$$M_C = \text{Max}(X_C); \text{for } c \text{ in range}(C). \tag{2}$$

3. Shared Fully Connected Layers (FC): Pass the outputs of GAP and CMP through a shared set of fully connected layers to model the channel-wise relationships and generate attention weights. The FC layers can be represented as W_1 and W_2, with ReLU activations. The nonlinear Relu activation function maps all negatives to zero, while positives remain unchanged. Its range is expressed as $F(x) = \text{Maximum}(0, x)$

$$A_C = \text{ReLU}\ (W_2(\text{ReLu}\ (W_1([S_C,\ M_c]))). \tag{3}$$

4. Channel Attention: Generate a channel attention mask by transforming the attention weights through a sigmoid activation function. The sigmoid is a linear activation function given by $F(x) = 0; x < 0$ and $F(x) = x; x \geq 0$

$$M = \text{Sigmoid}(A_C). \tag{4}$$

5. Rescaling: Multiply the input feature map X by the channel attention mask M to emphasize important channels:

$$Y = XM \tag{5}$$

Spatial Attention Module (SAM): The spatial attention module captures spatial dependencies to focus on informative regions. It operates as follows:

Given the output feature map Y from the channel attention module:

1. Channel-wise Max Pooling (CMP): Apply max pooling across the channel dimension of Y to capture the most important features within each channel:

$$M_{hwc} = \text{Max}\,(); \text{ for } h \text{ in range}(H), w \text{ in range}(W), c \text{ in range}(C). \tag{6}$$

2. Channel-wise Average Pooling (CAP): Apply average pooling across the channel dimension of Y to capture the overall distribution of features:

$$S_{hwc} = \frac{1}{C} * \Sigma Y_{hwc}); \text{ for } h \text{ in range}(H), w \text{ in range(W)}, c \text{ in range}(C). \tag{7}$$

3. Spatial Attention: Generate a spatial attention mask by concatenating the results of CMP and CAP, followed by a convolutional operation and a sigmoid activation:

$$S = \text{Sigmoid (Conv(ReLU ([}S_{\text{hwc}}, M_{\text{hwc}}\text{]}))) \tag{8}$$

4. Rescaling: Multiply the output feature map Y by the spatial attention mask S to emphasize informative regions:

$$Z = YS \tag{9}$$

The final output feature map Z represents the refined and modulated features obtained from both the channel attention module (CAM) and the spatial attention module (SAM). It encapsulates the improved representation learning, capturing discriminative information within the input feature map.

Feature Gating Mechanism and Feature Fusion. The output from two attention networks are added, passed through sigmoid activation, and multiplied by their respective input features. This forms the feature gating mechanism. Feature gating is an advantageous technique in image classification tasks, offering several benefits for the network. Here are some ways in which feature gating can enhance image classification:

Selective Feature Fusion: Feature gating allows the network to selectively combine or merge features from different layers or branches. This selective fusion emphasizes the most informative features while suppressing less relevant ones, leading to improved classification accuracy.

Adaptive Attention: By employing feature gating, the network can dynamically allocate attention to different regions of an image. The network assigns higher gating values to important regions and lower values to less informative areas, enabling focused attention on the most discriminative parts of the image. This adaptive attention mechanism results in improved classification performance.

Noise Reduction: Images often contain noise or irrelevant details that can impact classification accuracy. Feature gating acts as a filtering mechanism by assigning lower gating values to noisy or less informative features. This allows the network to reduce the influence of such features, resulting in noise reduction and improved robustness.

Class-specific Feature Amplification: Certain features may play a crucial role in distinguishing specific classes. Through feature gating, the network can amplify the importance of class-specific features. By doing so, the network focuses on discriminative aspects that are particularly relevant to different classes, potentially leading to improved classification accuracy for those classes. Feature gating in image classification enables the network to selectively combine features, allocate attention adaptively, reduce noise, and amplify class-specific information. This approach enhances the network's ability to capture and utilize the most relevant aspects of images, ultimately leading to improved performance, accuracy, and robustness in image classification tasks.

3 Implementations

Three datasets were used to verify the authentication of our model. They are UCMerced dataset [18]—21 classes with 2100 images, AID dataset [19]—30 classes with 10,000 images, and NWPU-RESISC45[20]—45 classes with 35,000 images dataset. The images are preprocessed, augmented, and rescaled to 256X256X3. For augmentation the images are rotated from 0 to 60°, vertical and horizontal flipped. Number of epochs was set to 300 and intial learning rate is 0.01. We implemented this network in Google-Colab-Pro + environment. We have used the standard training test ratio that was used by many of the state-of-art-models for better comparison. In UCMerced datset the training/test ratio is 50/50 and 80/20. For AID dataset, the training test ratio used is 20/80 and 50/50. And for NWPU-RESISC45 dataset, 10/20 and 20/80 training test ratio were used.

Table 1 Classification accuracy of our model compared with different state-of-art models with different train test ratio

Model	UCMeced		AID		NWPU	
	50/50	80/20	20/80	50/50	10/90	20/80
MBFANet	–	99.66	93.98	96.93	91.61	94.01
LCNN-BFF	–	99.29	91.66	94.62	86.53	91.54
ACNet	–	**99.76**	93.33	95.38	91.09	92.42
ACR-MLFF	97.99	99.37	92.73	95.06	90.01	92.45
SEMSDNet	98.40	99.41	94.23	**97.64**	91.68	93.89
EFFDA-FG (Ours)	**98.96**	**99.76**	**94.04**	96.89	**93.01**	**94.50**

4 Results and Discussions

4.1 Classification Accuracy

Comparison of accuracy between different state-of-art models is given in the Table 1, from which we could see that our suggested EFFDA-FG model gives the best accuracy except for AID 50/50 train/test ratio.

4.2 Classification Report

The obtained classification report shows that our model performs in a best way with precision, recall and f1-score as given in Tables 2 and 3 for AID and NWPU dataset. For AID dataset, 28 classes out of 30 classes show the precision, recall, and f1-scores to be more than 90%. These three factors lies in the range between 84 and 90% for other classes like center, church and square. For NWPU dataset, these three factors lie in the range of 90–100% for nearly 38 classes. 38/45 classes lie in the range of 90–100%. All the other classes church, palace, commercial-area, dense-residential, medium-residential, railway, rectangular-farmland have these values to be in range of 82–90%.

4.3 Confusion Matrix

The confusion matrix for aid dataset with 50/50 training test ratio is shown in Fig. 3, from which we can see that all the classes are classified with good accuracy. Nearly, 13 classes have 100% accuracy. Almost 27 classes have accuracy greater than 90%. The remaining three classes center, school, and stadium have accuracy of 88%, 82%,

Table 2 Classification report [precision/recall/f1-score] for different categories of AID dataset with 50/50 training test ratio

	Precision	Recall	F1-score
Airport	0.986301	1.000000	0.993103
Bare land	0.983871	0.983871	0.983871
Baseball field	0.977778	1.000000	0.988764
Beach	1.000000	1.000000	1.000000
Bridge	0.986301	1.000000	0.993103
Center	0.884615	0.884615	0.884615
Church	0.839286	0.979167	0.903846
Commercial	0.931507	0.971429	0.951049
Dense residential	0.987805	0.987805	0.987805
Desert	0.983607	1.000000	0.991736
Farmland	0.973333	0.986486	0.979866
Forest	1.000000	0.980000	0.989899
Industrial	0.974026	0.961538	0.967742
Meadow	0.965517	1.000000	0.982456
Medium residential	1.000000	0.965517	0.982456
Mountain	1.000000	1.000000	1.000000
Park	0.969231	0.900000	0.933333
Parking	1.000000	1.000000	1.000000
Playground	0.972603	0.959459	0.965986
Pond	0.976744	1.000000	0.988235
Port	1.000000	1.000000	1.000000
Railway station	0.98000	0.942308	0.960784
Resort	0.896552	0.896552	0.896552
River	1.000000	1.000000	1.000000
School	0.907407	0.816667	0.859649
Sparse residential	0.967742	1.000000	0.983607
Square	0.888889	0.848485	0.868217
Stadium	0.964912	0.948276	0.956522
Storage tanks	1.000000	0.986111	0.993007
Viaduct	0.98823	1.000000	0.994083
Accuracy	0.969000	0.969000	0.969000
Macro avg	0.966209	0.966610	0.966010
Weighted avg	0.969302	0.969000	0.968814

Table 3 Classification report [precision/recall/f1-score] for different categories of NWPU dataset with 20/80 training test ratio

	Precision	Recall	F1-score
Airplane	0.982659	0.971429	0.977011
Airport	0.960606	0.905714	0.932353
Baseball_Diamond	0.976812	0.962857	0.969784
Basketball_Court	0.971671	0.98	0.975818
Beach	0.955679	0.985714	0.970464
Bridge	0.954023	0.948571	0.951289
Chaparral	0.983099	0.997143	0.990071
Church	0.84	0.84	0.84
Circular_Farmland	0.977208	0.98	0.978602
Cloud	0.963585	0.982857	0.973126
Commercial_Area	0.888563	0.865714	0.87699
Dense_Residential	0.871314	0.928571	0.899032
Desert	0.959538	0.948571	0.954023
Forest	0.935933	0.96	0.947814
Freeway	0.971698	0.882857	0.92515
Golf_Course	0.982857	0.982857	0.982857
Ground_Track_Field	0.97191	0.988571	0.98017
Harbor	0.966574	0.991429	0.978843
Industrial_Area	0.951952	0.905714	0.928258
Intersection	0.976945	0.968571	0.97274
Island	0.985423	0.965714	0.975469
Lake	0.924581	0.945714	0.935028
Meadow	0.960843	0.911429	0.935484
Medium_Residential	0.850136	0.891429	0.870293
Mobile_Home_Park	0.962857	0.962857	0.962857
Mountain	0.943182	0.948571	0.945869
Overpass	0.918699	0.968571	0.942976
Palace	0.823353	0.785714	0.804094
Parking_Lot	0.928	0.994286	0.96
Railway	0.886243	0.957143	0.92033
Railway_Station	0.930303	0.877143	0.902941
Rectangular_Farmland	0.884718	0.942857	0.912863
River	0.945087	0.934286	0.939655
Roundabout	0.985294	0.957143	0.971014
Runway	0.920548	0.96	0.93986
Sea_Ice	0.997151	1	0.998573
Ship	0.960227	0.965714	0.962963
Snowberg	0.969529	1	0.984529

(continued)

Table 3 (continued)

	Precision	Recall	F1-score
Sparse_Residential	0.948127	0.94	0.944046
Stadium	0.98209	0.94	0.960584
Storage_Tank	0.969188	0.988571	0.978784
Tennis_Court	0.949438	0.965714	0.957507
Terrace	0.939306	0.928571	0.933908
Thermal_Power_Station	0.961194	0.92	0.940146
Wetland	0.953125	0.871429	0.910448
Accuracy	0.944444	0.944444	0.944444
Macro Avg	0.944917	0.944444	0.944325
Weighted Avg	0.944917	0.944444	0.944325

and 84%. Mostly they are interpreted among themselves as most of them look the same from top angle.

The confusion matrix of NWPU dataset with 20/80 training test ratio is shown in Fig. 4. Here we could see that 41 classes have accuracy in the range of 90–100%. Chapparrel, sea_ice and snowberg has 100% accuracy. Church, commercial_area, freeway, railway station, and wet land has accuracy in range of 80–90%. Palace has the lowest accuracy of 79%.

4.4 GradCAM

From the gradient-weighted-class-activation-mapping-plot (GradCAM) [21], Fig. 5, we can see the original sparse residential image and the heatmap generated at different outputs of our suggested model. Images b and c are the features obtained from resnext conv5_block1 and conv5_block3. Then *d* is the channel attention output that concentrates more on the center area, showing the residential part alone, but *e* is the channel spatial attention output that covers all area that might be used in deriving that it is sparse residential, i.e., it covers the center, pool area, residential part, and also the house behind the big trees and the road to this sparse area. Adding d and e gives *f*. Then feature *f* is passed through sigmoid layer, which supresses the unwanted features and concentrates on more required features. Sigmoid attention multiplied by its attention input is given by *h* and *i*. From *h* we can see that, more concentration is given to the center residential part while it omits the house on the right side, but it is captured well by *i*. By concatenating *h* and *i* features for classification purpose, our model shows improved performance.

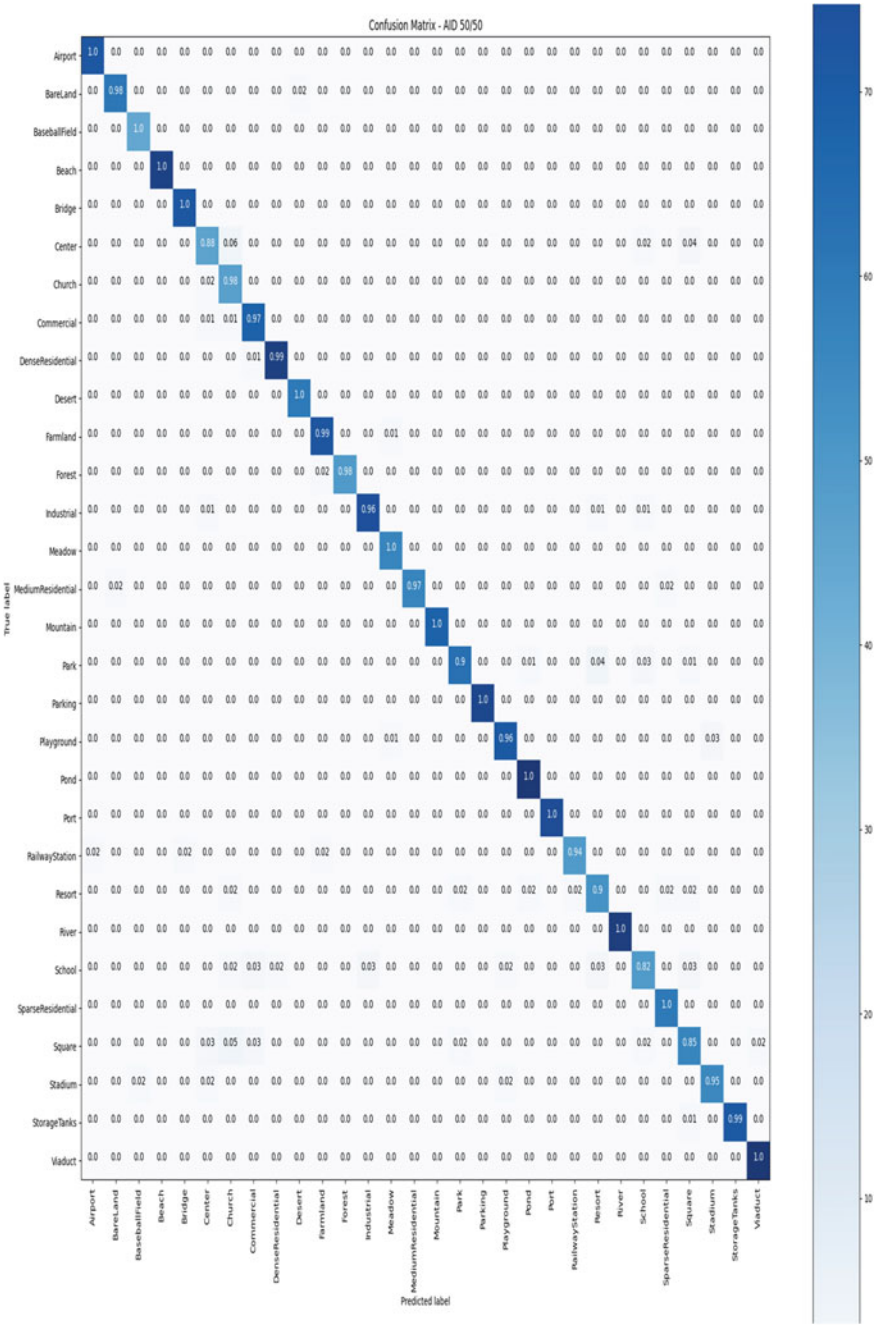


Fig. 3 Confusion matrix (CM) for AID dataset with 50:50 training test ratio

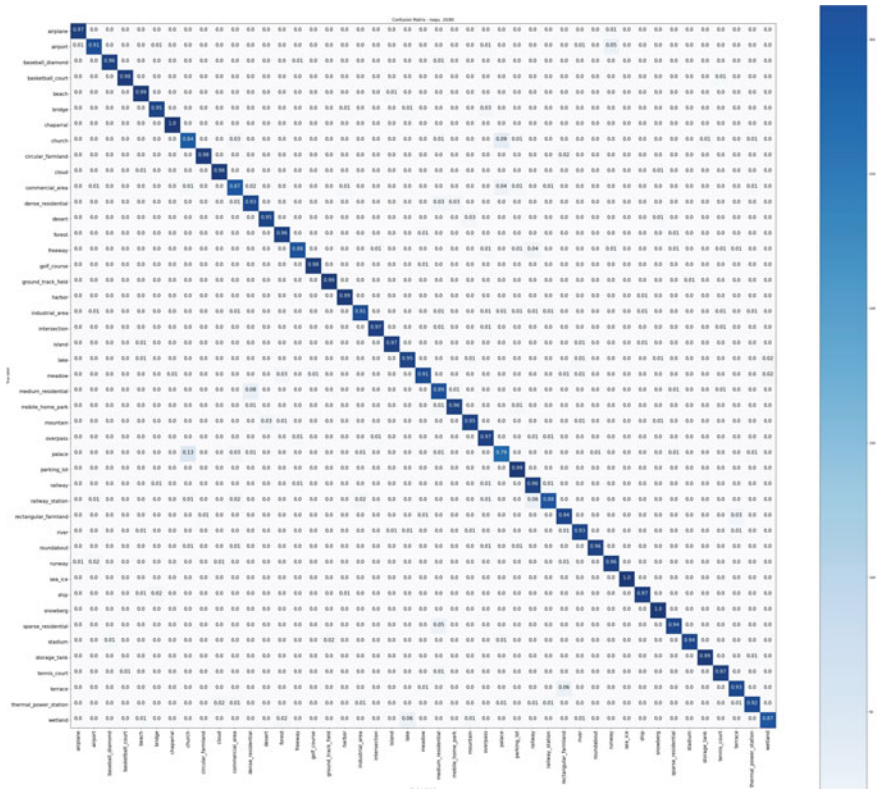


Fig. 4 Confusion matrix (CM) for NWPU dataset—20:80 training test ratio

4.5 T-SNE Plot

The t-distributed-stochastic neighbor embedding (T-SNE) [22], plot of our proposed model is shown in Fig. 6, for AID and NWPU dataset. Here, from the pictures, we can see that our model is able to obtain discriminate features as the separable clusters that are falling apart from each other.

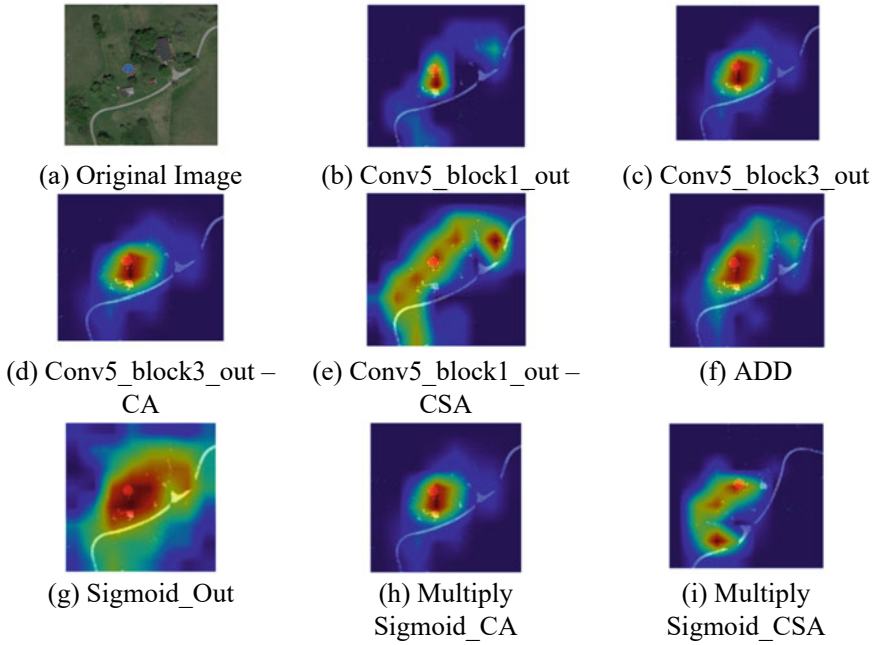


Fig. 5 GradCAM plot showing heatmaps generated at different levels of the model

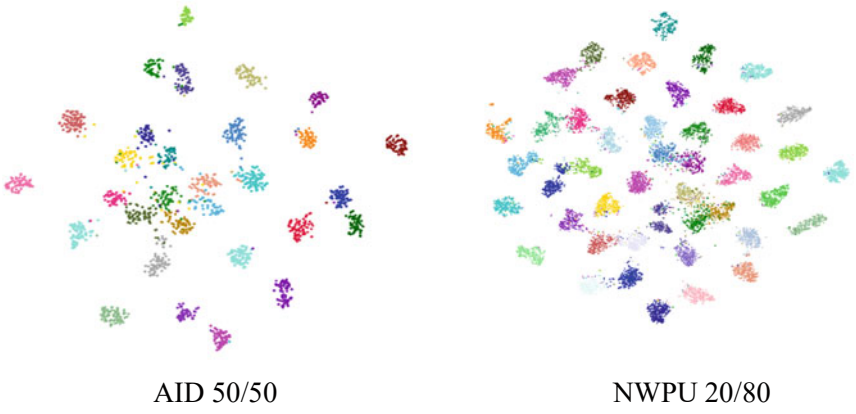


Fig. 6 T-SNE plot for AID and NWPU dataset for our model

5 Conclusion

A robust architecture with ResNeXt50 baseline model, named enhanced feature fusion from dual attention paths using feature gating mechanism (EFFDA-FG) has been suggested. Here we derived two sets of features from last convolutional stage of

ResNeXt50. Channel attention and channel spatial attention are performed on these features, and they are concatenated after passing through a sigmoid-based gating mechanism, which improves the performance of the model. The overall accuracies obtained by our model are 98.96, 99.76, 95.04, 96.89, 93.01, and 94.50 for different training testing ratios of UCM (50/50), UCM (80/20), AID (20/80), AID (50/50), NWPU (10/90), and NWPU (20/80), respectively. The precision, f1-score and recall also shows better performance. The confusion matrix was also analyzed, where we see that all categories are classified with a great accuracy. The GradCAM + plot and T-SNE plot proves that our model performs well.

References

1. Chen S, Ma W, Sun C, Chen S (2021) A comprehensive survey of remote sensing image scene classification: perspectives, datasets, and future directions. Remote Sens 13(12):2316
2. Zhou Z, Wang Y, Fu P, Liu L (2021) Feature learning and representation for remote sensing image scene classification: a comprehensive review. IEEE Trans Geosci Remote Sens 59(2):1011–1030
3. Dosovitskiy A, Beyer L, Kolesnikov A, Weissenborn D, Zhai X, Unterthiner T, Brox T (2021) An image is worth 16 × 16 words: Transformers for image recognition at scale. In: International conference on learning representations (ICLR)
4. Bazi Y, Bashmal L, Al Rahhal MM, Al Dayil R, Al AN (2021) Vision transformers for remote sensing image classification. Remote Sens 13(3):516
5. Roy SK, Deria A, Hong D, Rasti B, Plaza A, Chanussot J (2023) Multimodal fusion transformer for remote sensing ımage classification. IEEE Trans Geosci Remote Sens 61(1–20)
6. Castelluccio M, Poggi G, Sansone C, Verdoliva L (2015) Land use classification in remote sensing images by convolutional neural networks. arXiv:1508.00092
7. Li E, Xia J, Du P, Lin C, Samat A (2017) Integrating multilayer features of convolutional neural networks for remote sensing scene classification. IEEE Trans Geosci Remote Sens 55(10):5653–5665
8. Bi Q, Qin K, Zhang H, Xia G-S (2021) Local semantic enhanced ConvNet for aerial scene recognition. IEEE Trans Image Process 30:6498–6511. https://doi.org/10.1109/TIP.2021.3092816
9. Wang X, Duan L, Shi A, Zhou H (2022) Multilevel feature fusion networks with adaptive channel dimensionality reduction for remote sensing scene classification. IEEE Geosci Remote Sens Lett 19:1–5. Art no. 8010205. https://doi.org/10.1109/LGRS.2021.3070016
10. Li E, Samat A, Du P, Liu W, Hu J (2022) Improved bilinear CNN model for remote sensing scene classification. IEEE Geosci Remote Sens Lett 19:1–5
11. Xu K, Huang H, Li Y, Shi G (2020) Multilayer feature fusion network for scene classification in remote sensing. IEEE Geosci Remote Sens Lett 17(11):1894–1898
12. Wang S, Guan Y, Shao L (2020) Multi-granularity canonical appearance pooling for remote sensing scene classification. IEEE Trans Image Process 29:5396–5407
13. Wang X, Duan L, Ning C, Zhou H (2022) Relation-attention networks for remote sensing scene classification. IEEE J Sel Top Appl Earth Observ Remote Sens 15:422–439
14. Zhao Z, Li J, Luo Z, Li J, Chen C (2021) Remote sensing image scene classification based on an enhanced attention module. IEEE Geosci Remote Sens Lett 18(11):1926–1930
15. Chen S-B, Wei Q-S, Wang W-Z, Tang J, Luo B, Wang Z-Y (2022) Remote sensing scene classification via multi-branch local attention network. IEEE Trans Image Process 31:99–109
16. Yu D, Guo H, Xu Q, Lu J, Zhao C, Lin Y (2020) Hierarchical attention and bilinear fusion for remote sensing image scene classification. IEEE J Sel Top Appl Earth Observations Remote Sens 13:6372–6383. https://doi.org/10.1109/JSTARS.2020.3030257

17. Woo S, Park J, Lee J-Y, Kweon IS (2018) CBAM: convolutional block attention module. In: Proceedings of the European conference on computer vision. Springer, Munich, pp 3–19
18. Yang X, Zhou W, He Z, Gao J (2010) Urban land cover classification using integrated remote sensing and GIS techniques: a case study in the Phoenix metropolitan area. Remote Sens Environ 114(8):1833–1848
19. Xia GS, Hu F, Hu H, Shi X, Bai X, Zhong Y, Zhang L (2017) AID: a benchmark data set for performance evaluation of aerial scene classification. IEEE Trans Geosci Remote Sens 55(7):3965–3981
20. Cheng G, Han J, Lu X, Zhang C, Sun J (2017) Remote sensing image scene classification: benchmark and state of the art. Proc IEEE 105(10):1865–1883. https://doi.org/10.1109/JPROC.2017.2712042
21. Selvaraju RR, Cogswell M, Das A, Vedantam R, Parikh D, Batra D (2019) Grad-CAM: Visual explanations from deep networks via gradient-based localization. In: Proceedings of the IEEE ınternational conference on computer vision, pp 618–626
22. Van der Maaten L, Hinton G (2008) Visualizing data using t-SNE. J Mach Learn Res 9(Nov):2579–2605

Thermal Image-Based Fault Detection Using Machine Learning and Deep Learning in Industrial Machines: Issues-Challenges and Emerging Trends



Vaishnavi V. Kulkarni, Vishwanath R. Hulipalled, Mayuri Kundu, Jay B. Simha, and Shinu Abhi

Abstract Thermal imaging has long been utilized across industries to maintain electrical equipment and detect faults in machines, ensuring their reliable operation. Infrared thermography, or thermal imaging, has emerged as a powerful, non-contact, non-destructive monitoring tool. Unlike RGB imaging, it offers a non-intrusive approach that respects privacy and does not rely on ambient lighting. As heat signatures vary based on defects or machine conditions, infrared imaging is essential for effective condition monitoring. This survey study introduces the principles and applications of thermal imaging in fault detection and a comprehensive analysis of recent developments and approaches in the field. The survey encompasses various machine types, including electrical systems, mechanical components, and industrial equipment. Multiple techniques, algorithms, and deep learning models for image processing, feature extraction, and fault classification are extensively discussed. The paper also includes an in-depth literature review on fault detection in distinct machines, such as induction motors, rotating machines, and transformers. The survey aims to identify open challenges and issues in fault detection for different machines, such as induction motors and transformers.

Keywords Thermal imaging · Fault detection · Induction motor · Rotating machine · Transformer · Infrared thermography

V. V. Kulkarni (✉) · M. Kundu
School of CSE, REVA University, Bangalore, India
e-mail: R22PVCS01@reva.edu.in

M. Kundu
e-mail: mayuri.kundu@reva.edu.in

V. R. Hulipalled
School of C&IT, REVA University, Bangalore, India
e-mail: vishwanath.rh@reva.edu.in

J. B. Simha · S. Abhi
REVA Academy for Corporate Excellence (RACE) REVA University, Bangalore, India
e-mail: shinuabhi@reva.edu.in

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2023

S. Shakya et al. (eds.), *Fourth International Conference on Image Processing and Capsule Networks*, Lecture Notes in Networks and Systems 798,
https://doi.org/10.1007/978-981-99-7093-3_39

1 Introduction

Due to increased specialization and investment, the costs and complexity of industrial machinery have risen significantly. Ensuring the health of machines through fault diagnostics and monitoring is crucial to prevent productivity loss and system failures. Infrared thermography, a technique for acquiring and visualizing thermal pictures, has become popular. Unlike RGB imaging, thermal imaging relies solely on the emitted infrared radiation from objects, making it faster and more efficient in processing information. Thermal imaging sensors are now commonly used in various applications, such as fever scanners, insulation detectors, and electrical hotspot detectors.

As thermal patterns change depending on defects or machine conditions, thermography is essential for monitoring machine health non-intrusively. Predictive maintenance, utilizing sensing and inspection data, estimates the future deterioration and remaining usable life of equipment, aiding in maintenance planning and other engineering tasks. Thermal imaging cameras convert thermal energy into visible light to study specific objects or scenes, and the resulting images, known as thermograms, are analyzed using thermography methods.

1.1 Applications of Thermal Imaging

Figure 1 illustrates the diverse applications of thermal imaging. These include breast cancer detection and diabetic foot examination, where body temperature variations are identified. Infrared imaging is effective for monitoring mechanical and electrical installation conditions. Additionally, thermal visualization aids in detecting cold and hot spots, allowing for the identification of machine faults.

Gas leaks, including methane, propane, and natural gas can be detected using thermal imaging as these gases emit thermal radiation that is invisible to the naked eye. Thermal cameras can identify temperature variations caused by gas leaks, making them valuable tools for safety inspections.

In building inspections, thermal cameras detect places with heat loss or moisture intrusion, such as inadequately insulated walls, roofs, and windows [1]. The technology works by detecting the infrared radiation emitted by objects to create images showing temperature variations.

Thermal imaging cameras have various applications beyond gas leak detection. The military and defense industries were among the first to adopt thermal cameras. They are used in maritime applications for navigation, search, and rescue missions, agriculture for plant disease identification and crop monitoring, and border security for tracking and detecting vehicles and people.

In surveillance, thermal imaging can be utilized to distinguish faces, especially in low light or poor visibility conditions. Thermal cameras can recognize people based on the heat signatures their faces emit.

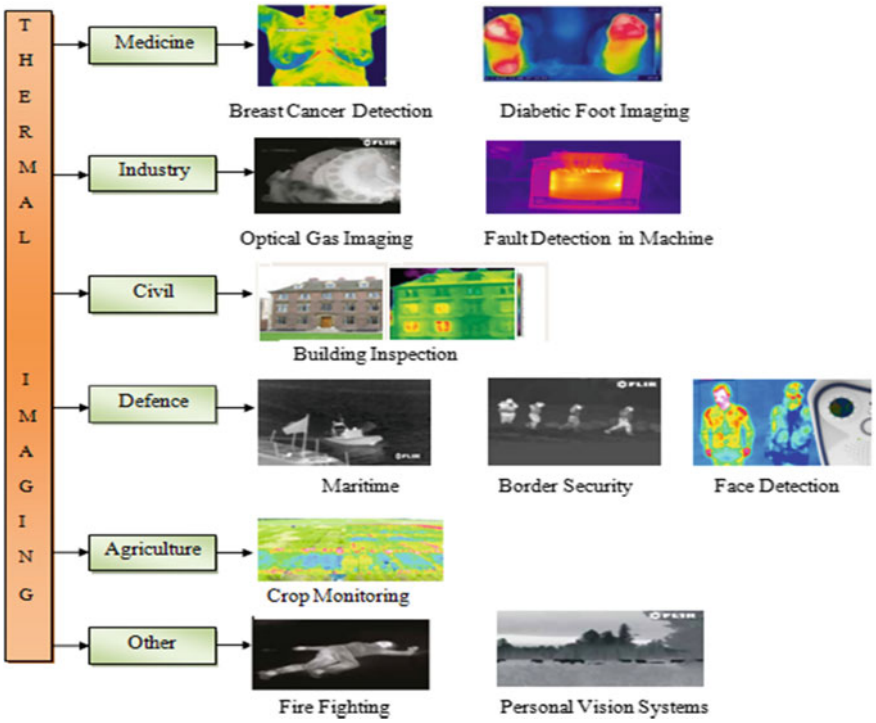


Fig. 1 Different applications of thermal imaging

Additionally, infrared cameras are used to monitor fire behavior, detect hidden fires, and for personal imaging systems to observe wildlife at night. They are also helpful in outdoor activities like camping and hiking to identify hidden dangers.

Thermal Imaging in Breast Cancer Detection. Breast cancer detection through thermal imaging, also known as "thermography" or "thermal breast imaging," involves capturing heat patterns emitted by breast tissue using specialized thermal cameras that detect infrared radiation. These cameras create thermal maps of the breast's surface, highlighting areas with abnormal heat patterns. Cancer cells often exhibit higher metabolic rates, leading to increased blood flow and elevated temperatures in affected regions. By comparing these thermal maps with established patterns and normal thermograms, potential abnormalities can be identified, aiding in early breast cancer detection.

However, thermal imaging is not a standalone diagnostic method, and conventional techniques like mammography and ultrasound are still necessary for definitive diagnosis and further evaluation. Proper training, standardization, and interpretation are crucial to achieving accurate results and maximizing the potential of thermal imaging in breast cancer detection.

Thermal Imaging in Face Recognition. Thermal imaging can be utilized to distinguish faces based on the differences in thermal patterns emitted by different parts of the face. It's called facial heat signature: Each person's face emits a unique heat signature due to variations in blood flow, skin temperature, and surface features. For instance, the eyes, nose, and mouth exhibit different thermal patterns, making it possible to identify facial landmarks and distinguish one face from another. Thermal imaging captures these heat patterns, creating distinct thermal representations for individuals. Unlike visible light-based facial recognition, thermal imaging works effectively in low light or complete darkness, making it valuable for surveillance and security applications.

2 Literature Survey

Defect detection in machines can be achieved through two approaches: thermal imaging analysis and conventional signal analysis techniques. The latter involves analyzing signals like vibration, current, and sound to identify defects in rotating machinery. The vibration-based technique is widely used due to its ability to represent the dynamic properties of machines accurately.

Recently, the combination of thermal imaging and machine learning has become popular. Thermal images provide temperature details that help detect anomalies in systems or devices. This non-contact method allows quick identification and examination of flaws without human involvement, thanks to advanced machine learning techniques. Integrating thermal imaging with machine learning enables efficient analysis of large datasets.

Researchers have explored various approaches in this field. For instance, Glowacz [2] analyzed the infrared images of induction and commutator motors, using features extracted through arithmetic mean along with the Otsu approach, the mean of a histogram, and the mean along with the 20 highest peaks. Furthermore, it was classified using long short-term memory (LSTM) and nearest neighbor (NN) classifier methods, achieving a recognition rate of 95.33–100%.

Sakallı and Koyuncu [3] applied DenseNet201, MobileNetV2, ResNet50, ShuffleNet, and Xception deep learning models on a Mendeley database that contains thermal images of induction motors and transformers and achieved a recognition rate of 100%.

Shao et al. [4] developed generative adversarial network (GAN) to extract global infrared correlation features from infrared images and classified them by convolutional neural network (CNN) that gained above 90% accuracy.

Javed et al. [5] collected infrared images in various states, like normal and overloaded to prevent motor faults. Features were extracted using local Octa pattern (LOP) along with genetic algorithm (GA) and were classified using a support vector machine (SVM), and this approach exhibited above 90% accuracy.

Ibrahim et al. [6] developed a hybrid model for fault detection in motors, tested at three different running speeds: 1480, 1450, and 1380 rpm, encompassing bearing

problems, broken rotor bar faults, and stator faults. They used a FLIR C2 camera to capture infrared images of motors with six faulty states. Feature extraction was performed using gray level co-occurrence matrices (GLCM), and the invasive weed optimization (IWO) method was utilized for feature selection. The study employed random forest (RF) and extreme gradient boosting (XGBoost) algorithms to classify healthy and malfunctioning motor states.

Bhagat and Chauhan [7] extracted features using GLCM and IWO-based algorithms. Furthermore, authors categorized the various states of a motor using XGB and random forest (RF) classifier that resulted in 93% and 85% recognition rates evaluated on no load, 50% load, and 100% load conditions.

BV and Ananthan [8] extracted texture features and applied the sliding window method to detect the area of interest. Furthermore, the images are classified as decision tree, AdaBoost, random forest, and gradient-boosted trees and obtained an accuracy of 93.9%, 93.8%, 97.2%, and 92.8%, respectively.

Choudhary et al. [9] overcome the issue of insufficient information and string noise in the thermal images that hinder the performance of motors. They preprocessed the images using 2D discrete wavelet transform and extracted the pertinent features using principal component analysis (PCA). The chosen attributes were sorted according to importance using Mahalanobis distance and classified using complex decision tree (CDT), linear discriminant analysis (LDA), and SVM.

Mahami et al. [10] proposed an ensemble learning method that uses a bag-of-visual word (BoVW) for feature extraction and classified using an extremely randomized tree (ERT), SVM, RF, KNN, the self-organizing fuzzy logic classifier (SOF), deep rule-based (DRB), LSSVM applied these methods on a three-phase system with the following specifications: 11 kW of output power, 5 A of current, and 220/230 V of voltage that achieved 100% accuracy.

Glowacz [11] calculated the mean of infrared images of brushless dc motors and extracted the features. Next, the authors analyzed the states of motors running at different speeds with blocked ventilation and healthy and faulty clippers that achieved 100% classification accuracy.

Choudhary et al. [12] considered one healthy and five abnormal states of rolling element bearings and used an artificial neural network (ANN) and LeNet-5 convolutional neural network (CNN). The faulty states considered were outer and inner race bearing states, cage defect, ball defect, and lack of lubrication. CNN gained an average accuracy of 99.80% in all circumstances.

Mehta et al. [13] performed feature extraction using 2D discrete wavelet transform (DWT) and reduced the dimension of the extracted features using PCA and KNN, SVM, and LDA methods for classification. The tests were carried out on a bearing test rig with a single-phase 2 HP, 2 poles, and 220 V induction motor, and all the classifiers gained above 95% accuracy.

Fang et al. [14] employed various techniques to extract details based on temperature, texture, and shape from infrared images. They used Hu moment characteristics, GLCM features, and regional temperature features. After combining this data, they used a GAN model to input it. According to the findings of their studies, the model

had an average classification success rate of roughly. 84.2% and accurately classified the categories of equipment failures and overheating with 82.2% and 86.2% accuracy, respectively.

Fanchiang et al. [15] introduced an innovative IRT-based technique for detecting overheating issues in cast-resin transformers through their full-time online fault monitoring system. They trained the Wasserstein autoencoder reconstruction (WAR) model and the differential image classification (DIC) model. The differential image is obtained by calculating the absolute difference between actual and regenerated images at the pixel level. During testing, a defect diagnosis module is created by combining the well-trained WAR and DIC models. The experimental results demonstrate that the proposed model surpasses existing deep learning approaches in terms of being lightweight, having a small storage size, offering quick inference time, and providing precise diagnostic performance.

Early-stage diagnosing malfunctions in equipment and machines by looking for temperature changes in thermal pictures can help prevent equipment failures, flame risks, and other potential hazards. Although many fault scenarios generate high heat, certain defects cause equipment to produce less heat than normal. This might be difficult when trying to differentiate between functioning and defective equipment.

Najafi et al. [16] address the problem by dividing the data into hot and cold states. The authors find the region of interest in a cold state by random forest and hot state using AdaBoost and perform sliding window-based segmentation. The suggested methods were evaluated on the Mendeley datasets.

Khanjani and Ezoji [17] identified the area of interest by matching SIFT key points. Next, they are represented as feature vectors using CNN and classified into cold and hot clusters using K means and SVM algorithms that gain 100% accuracy.

Lozanov et al. [18] developed a model employing infrared pictures of a motor in different conditions, such as a healthy motor, motor bearing failure, and cooling fan. The framework uses three textural features, i.e., GLCM-based, component-based, and histogram-based to extract the essential data and Otsu thresholding to segment the motor structure. Support vector machines (SVM) are used to do the classification task. The classification accuracy of the framework, which was assessed with a 62.5–37.5% training-test split, was 83.3% for the three motor conditions.

Al-musawi et al. [19] utilized thermal images and converted them into the hue, saturation, and value (HSV) color model. They employed various edge detection methods, such as Otsu, Roberts, Prewitt, Sobel, and Canny to identify the hot regions based on a threshold. To extract essential defect information from the Hue image, multiple image matrices were applied. Additionally, the researchers used image metrics, including mean, mean square error, peak signal-to-noise ratio, variance, standard deviation, skewness, and kurtosis to detect abnormalities in ball bearings, inner races, and outer races of three-phase induction motor bearings.

Li et al. [20] introduced nine defective states to the rotating machine and processed the infrared images using CNN, DBN, DNN, and SAE models that gained classification accuracy of 98.64%, 83.60%, 75.53%, and 60.84%.

Various researchers have contributed to machine fault detection, focusing on induction motors, transformers, rotating machines, and electrical equipment. Their

valuable insights and methodologies are documented in the form of research papers, which are summarized in Table 1.

3 Types of Faults in Machines

Machines can experience internal and external faults, further categorized as mechanical, electrical, and environmental issues. In induction motors and rotating machines, common faults include stator winding defects (stator faults), faults in the rotor parts (rotor faults), and lubrication issues. In ball bearings, distinct faults like inner and outer race damages are observed, often resulting from running at high speeds. Researchers like López-Pérez and Antonino-Daviu [21] have discussed various faults, including ruptured rotor bars, harmonic interference, shaft misalignment, operational overload, lubricant contamination, and voltage imbalance. Figure 2 [13] illustrates the faults present in bearings, featuring different sizes of holes.

Next, the transformers with no and heavy loads are shown [22] in Fig. 3.

4 Issues/Challenges

Challenges and difficulties encountered in collecting infrared images and identifying machine faults include:

- Temperature Fluctuations: Variations in temperature due to external conditions or heat transfer properties of machine components can make it challenging to interpret thermal imaging and distinguish actual flaws from typical temperature swings.
- Sensor Complexity: Different sensors used for vibration data collection can complicate sensor installation and potentially harm rotating equipment. Vibration data can also produce significant noise during machine operation, making fault identification difficult, which can be mitigated through infrared thermography for fault detection.
- Insufficient Data: Acquiring sufficient thermal imaging data for training and evaluating AI/ML models may be challenging and time-consuming, especially under various faulty conditions.
- Environmental Factors: Environmental elements like humidity and temperature can impact thermal imaging, affecting the identification of fault conditions and requiring consideration during data gathering and analysis.
- Complexity and Cost: Implementing effective defect detection systems can be expensive and resource-intensive, making long-term maintenance challenging.

Table 1 Fault identification in machines using infrared images

Study	Algorithms	Equipment	Fault examined	Accuracy
Glowacz, Adam [2]	Differences of arithmetic mean with Otsu's method NN, LSTM	Commutator, induction motor	"Healthy commutator motor (HCM), broken rotor coil of the commutator motor (BRCoCM), shorted stator coils of the commutator motor (SSCoCM), healthy single-phase induction motor (HSIM), single-phase induction motor with shorted coils of auxiliary winding (SIMwSCoAW)"	ACC: 95.33–100%
Gönül Sakallı and Hasan Koyuncu [3]	DenseNet201, MobileNetV2, ResNet50, ShuffleNet, Xception	Transformer, induction motor	"Cooling fan failure, rotor fault, short-circuit faults transformer, no load motor, and no load transformer"	ACC: 100%
Shao et al. [4]	Dual-threshold attention-guided GAN (DTAGAN), CNN	Rotating machine	Ball fault of bearing, compound fault, inner and outer race faults of the normal and unbalanced rotor	Above 90%
Javed et al. [5]	LOP, GA, SVM	Induction motor	Overloaded, fault, and normal operating conditions	Above 90%
Ibrahim et al. [6]	GLCM, IWO Random Forest, XGBoost	Induction motor	Overloaded, fault, and normal operating conditions	XGBoost:93% Random Forest:85%
Bhagat and Chauhan [7]	GLCM, IWO Random forest XGBoost	Induction motor	No load, 50% load, and 100% load conditions	RF: 85%, XGBoost: 93%
BV and Ananthan [8]	Sliding window method, GLCM	Induction motor	Rotor fault, 1,2,3 phases stator fault, cooling fan failure	Gradient boost: 87.8% Ada boost: 87.8% Decision tree: 87.9% RF: 90%
Choudhary et al. [9]	2D-DWT, CDT, PCA, LDA, and SVM	Induction motor	Inner race, and outer race faults, lubrication defect	LDA:100%, CDT and SVM above 95%
Mahami et al. [10]	BoVW, SURF, SIFT, ERT, KNN, SOF, RF, DRB, LSSVM	Induction motor	Stator faults under different states and conditions	ACC: 100%

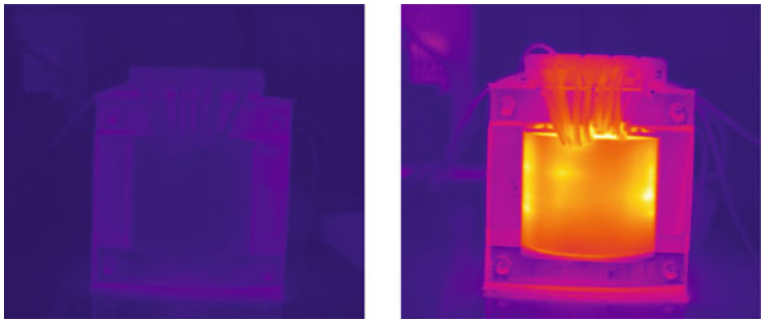
(continued)

Table 1 (continued)

Study	Algorithms	Equipment	Fault examined	Accuracy
Glowacz in [11]	CPoAMoTI	Brushless DC motors	Blocked ventilation, healthy and faulty clippers	ACC: 100%
Choudhary et al. [12]	ANN, LeNet-5, CNN	Rotating machine	Inner and outer race faults, lubrication, ball, and cage defects	CNN: 99.80%
Mehta et al. [13]	2D-DWT, PCA, LDA, KNN, SVM	Rotating machine	"Healthy, inner race, and outer race bearing states"	Above 95%
Fang et al. [14]	GLCM, GAN	Electric transformer	Defective and overheated faults	Above 85%
Fanchiang et al. [15]	WAR, DIC	Cast-resin transformer	Wire and connection overheating, normal and inter-turn short circuit	ACC: 99.95%
Najafi et al. in [16]	Sliding window-based method, Ada boost, RF	The three-phase induction motor and transformer	11 classes of induction motor 9 classes of transformer states	ACC: 93.8%, 95.6% for 11 classes 9 classes
Khanjani et al. [17]	SIFT, CNN, K means SVM	Three-phase induction motor	No load and loaded states	ACC: 100%
Musawi et al. [19]	HSV color model Otsu, Roberts, Prewitt, Sobel, and Canny edge detection methods	Three-phase induction motor	Inner race and outer race bearing defects	Mean in healthy motor = 0.5305 with 50% of load based on the Otsu method
Li et al. [20]	CNN LeNet-5 model, DBNs, DNN, SAE	Rotating machine	Inner race, outer race, balanced and unbalanced faults	CNN:98.64% DBN:83.60% DNN:75.53% SAE: 60.84%



Fig. 2 Various states of bearings



(a) (b)

Fig. 3 Transformer with **a** No load **b** Load

- Interpretability of AI/ML Models: The lack of transparency in AI/ML models' decision-making hampers their interpretability, despite high predictive accuracy. Understanding the variables driving the model's predictions and providing justifications for maintenance recommendations in predictive maintenance is essential, and future research should focus on developing methods to clarify and comprehend AI/ML models' choices when analyzing thermal images.

Open Chanllenges. Here are some of the key open challenges in thermal imaging:

- Resolution and Image Quality: Improve thermal camera clarity and sharpness.
- Sensitivity and Temperature Range: Enhance detection of subtle temperature differences.
- Cost: Reduce the expense of thermal imaging systems.
- Real-Time Processing: Develop fast data processing for immediate applications.
- Environmental Interference: Minimize the impact of external factors on thermal data.

5 Methodology

To manage and prevent fault diagnosis in machines, early identification is crucial. The general flowchart of the proposed method for early fault detection and classification is represented in Fig. 4.

The steps in the general framework are as follows:

Step 1. Data Collection: Thermal images are acquired using thermal cameras such as FLIR, Fluke, and Dali tech cameras with suitable resolutions for different machine types. Infrared images are collected in healthy and abnormal states under different defective conditions and machines running at different speeds.

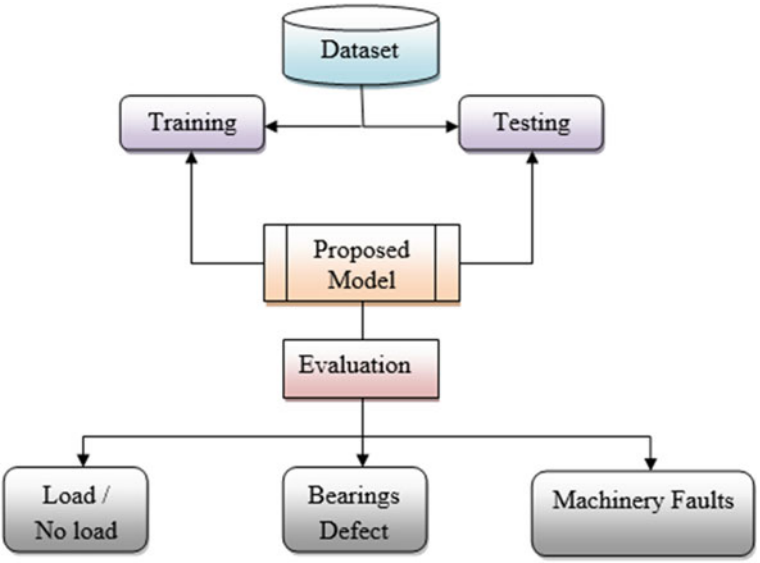


Fig. 4 General flowchart of the proposed method

Step 2. Data Division: The collected dataset is divided into training, validation, and testing. A common data split ratio is used, such as 70% for training, 10% for validation, and 20% for testing.
Step 3. Data Augmentation: Data augmentation techniques are applied to increase the training dataset's diversity and improve the model's generalization. Augmentation techniques include rotation, flipping, zooming, and adding noise to thermal images.
Step 4. Preprocessing: Preprocessing techniques are applied to enhance image quality, eliminate noise, and identify specific regions of interest. Filters remove noise and segment regions based on temperature variations to identify hot and cold spots. Sample infrared images of transformer are shown in Fig. 5.
Step 5. Feature Extraction: Features are extracted using various image processing algorithms such as color thresholding, wavelet transforms, GLCM, and deep learning models. Transfer learning models can also automatically extract features,

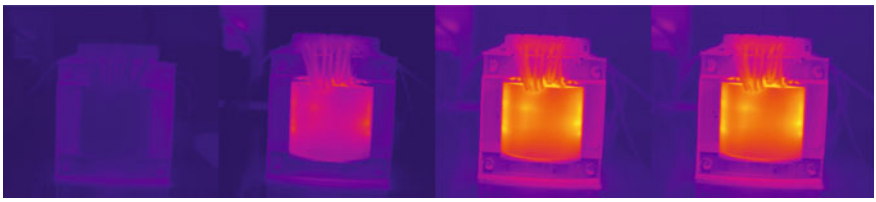


Fig. 5 Sample infrared images of the transformer [22]

eliminating the need for manual feature extraction. Both approaches can be utilized based on specific requirements.
Step 6. Training Module: In this module, the training dataset, which contains 70% of the collected data, is used to train machine learning algorithms, artificial neural networks (ANN), or convolutional neural networks (CNN). The model learns to recognize patterns in thermal images corresponding to different machine faults.
Step 7. Validation Module: During the training process, the model's performance is evaluated on the validation dataset, which contains 10% of the data to tune hyperparameters, prevent overfitting, and monitor training progress.
Step 8. Testing Module: After training, the trained model is tested using the separate testing dataset, which contains 20% of the data, containing thermal images of machines with known faults. The model's performance is evaluated based on accuracy, precision, recall, and f1 score to assess its ability to classify different machine faults correctly.
Step 9. Analysis: The methods are evaluated based on multiple performance measurements, and the results are analyzed to determine the effectiveness and accuracy of the proposed fault detection and classification system. The data division and augmentation strategies are vital to the model's overall performance and generalization capability.

5.1 Discussion

Selection of Algorithms for Thermal Imaging. The selection of algorithms for thermal imaging should be justified based on the specific task and requirements of the application. Different algorithms excel in different aspects of thermal image processing, and the choice depends on factors such as accuracy, speed, computational complexity, and the nature of the problem being addressed.

- Object Detection: For detecting objects in thermal images, deep learning-based object detection algorithms like you only look once (YOLO) or faster region-based convolutional neural networks (R-CNN) are commonly used due to their high accuracy and ability to handle complex scenes.
- Image Segmentation: For segmenting regions of interest in thermal images, semantic segmentation models like U-Net or Mask R-CNN are often employed, as they can provide pixel-level accuracy.
- Image Fusion: For combining thermal and visible light images, image fusion algorithms like wavelet transform or Laplacian pyramid can be used to enhance the interpretability of the combined images.
- Anomaly Detection: Anomaly detection algorithms, such as one-class support vector machine (SVM) or autoencoders are suitable for identifying abnormal thermal patterns that may indicate faults or irregularities.
- Real-Time Processing: In scenarios where real-time processing is critical, lightweight deep learning models, such as MobileNet or EfficientNet are preferred due to their reduced computational demands.

Need of Preprocessing and its Impact on Performance. Preprocessing is a critical step in thermal imaging analysis that enhances the quality of images and prepares them for defect detection. It involves noise reduction, region segmentation, image normalization, feature extraction, and outlier removal. The impact of preprocessing on model performance is significant. It improves the accuracy of defect detection by reducing noise, focusing on relevant regions of interest, bringing images to a consistent scale, extracting essential features, and enhancing the model's robustness. Effective preprocessing is essential for optimizing thermal images for machine learning algorithms and plays a crucial role in achieving accurate and reliable results in thermal imaging applications.

Improving Thermal Imaging Performance

To improve thermal imaging performance, upgrade to higher-resolution sensors for better image quality and detailed thermal information. Implement advanced signal processing algorithms to accurately interpret data and identify patterns. Increase sensitivity and temperature range for detecting subtle differences in various environments. Integrate with other imaging modalities for comprehensive analysis and object recognition.

Use intelligent algorithms for automated pattern recognition. Consider environmental factors to reduce interference and encourage cross-disciplinary collaboration for innovative advancements in thermal imaging technology across industries.

6 Results and Evaluation

Thermal Imaging Dataset. Several widely used thermal imaging datasets are available for research and development purposes in the field of thermal imaging and machine learning. These datasets play a crucial role in training and evaluating algorithms for various applications. Some of the popular thermal imaging datasets are as follows:

- FLIR Thermal dataset
- VTT Thermography dataset
- IR-ImageNet
- KAIST Multispectral Pedestrian dataset
- Infrared Pedestrian dataset (IRPD)
- FLIR ADAS dataset

Performance Evaluation. In diagnostic tests, a sample can be categorized as true negative (actual negative), true positive (actual positive), false negative (wrongly anticipated as negative), or false positive (wrongly predicted as positive). False negative indicates that the test incorrectly anticipated a negative result for a positive sample, while true positive indicates that the test correctly predicted a positive sample. False positive means that the test mistook a negative sample for a positive one, while true negative denotes a correctly predicted negative sample.

Various metrics are frequently used to assess diagnostic models, including specificity, accuracy, sensitivity (also known as recall), f1 score, precision, and the receiver operating characteristic (ROC) curve. These evaluation measures can be quantitatively expressed using Eqs. 1 through 7, which provide measures for f1 score, accuracy, specificity, sensitivity, and precision. The ROC curve visually represents the performance of a classification model at different levels.

$$\text{Accuracy} = \frac{\text{Total number actual predictions}}{\text{Total number of predictions}} \tag{1}$$

$$\text{Specificity} = \frac{\text{True negatives}}{\text{True negatives} + \text{False Positives}} \tag{2}$$

$$\text{Sensitivity} = \frac{\text{True positives}}{\text{True positives} + \text{False positives}} \tag{3}$$

$$\text{Precision} = \frac{\text{True positives}}{\text{True positives} + \text{False positives}} \tag{4}$$

$$F1\text{Score} = 2 * \left(\frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \right) \tag{5}$$

$$\text{TPR} = \frac{\text{True positives}}{\text{True positives} + \text{False negatives}} \tag{6}$$

$$\text{FPR} = \frac{\text{False positives}}{\text{False positives} + \text{True negatives}} \tag{7}$$

7 Conclusion

The literature review has established thermal imaging as a promising method for detecting machine defects. Analyzing thermal images to understand temperature distribution and patterns in machine components can aid in the early identification of potential flaws before they escalate. Numerous studies have demonstrated the effectiveness of thermal imaging in detecting various issues like bearing, gear, and motor faults. Early detection of failures leads to reduced downtime, increased equipment lifespan, enhanced safety, and reduced manufacturing costs. However, several factors need to be considered when choosing a specific defect detection technique. The survey also highlights some challenges that need resolution to effectively use thermal imaging in machine fault detection, such as data availability and interpretability of machine learning models. Overall, the findings suggest that thermal imaging holds significant potential as an affordable, non-destructive, and non-intrusive tool for detecting machine faults.

Acknowledgements The author would like to acknowledge the Visvesvaraya Ph.D. scheme, Meity, Govt. of India, and REVA University for providing the required infrastructure and other support.

References

1. Wilson AN, Gupta KA, Koduru BH, Kumar A, Jha A, Cenkeramaddi LR (2023) Recent advances in thermal ımaging and its applications using machine learning: a review. IEEE Sens J 23(4):3395–3407, 15 Feb 2023
2. Glowacz AT (2023) Thermographic, Fault diagnosis of electrical faults of commutator and induction motors
3. Sakallı G, Koyuncu H (2023) Identification of asynchronous motor and transformer situations in thermal images by utilizing transfer learning-based deep learning architectures. Measurement 207
4. Shao H, Li W, Cai B, Wan J, Xiao Y, Yan S (2023) Dual-threshold attention-guided gan and limited ınfrared thermal ımages for rotating machinery fault diagnosis under speed fluctuation. IEEE Trans Ind Inf
5. Javed M, Shabbir Z, Asghar F, Amjad W, Mahmood F, Khan O, Virk US, Waleed A, Haider Z (2022) An efficient fault detection method for ınduction motors using thermal ımaging and machine vision. Sustainability
6. Ibrahim A, Anayi F, Packianather M (2022) New machine learning model-based fault diagnosis of ınduction motors using thermal images. In: 2nd ınternational conference on advance computing and ınnovative technologies in engineering (ICACITE). India, pp 58–62
7. Bhagat AK, Chauhan A (2022) Thermal ımage-based fault analysis of ınduction motors using a novel machine learning model. In: 11th ınternational conference on system modeling & advancement in research trends (SMART). India, pp 1429–1433
8. BV C, Ananthan T (2022) Machine learning based fault detection in ınduction motor using thermal ımaging. In: 2022 3rd ınternational conference on electronics and sustainable communication systems (ICESC). India, pp 929–936
9. Choudhary A, Goyal D, Letha SS (2021) Infrared thermography-based fault diagnosis of ınduction motor bearings using machine learning. IEEE Sens J 21(2):1727–1734, 15 Jan 2021
10. Mahami A, Rahmoune C, Bettahar T, Benazzouz D (2021) Induction motor condition monitoring using infrared thermography imaging and ensemble learning techniques. Adv Mech Eng
11. Glowacz A (2021) Thermographic fault diagnosis of ventilation in BLDC motors. Sensors 21(21):7245
12. Choudhary A, Mian T, Fatima S (2021) Convolutional neural network based bearing fault diagnosis of rotating machine using thermal ımages. Measurement
13. Mehta A, Goyal D, Choudhary A, Pabla B, Belghith S (2021) Machine learning-based fault diagnosis of self-aligning bearings for rotating machinery using ınfrared thermography. Math Probl Eng
14. Fang J, Yang F, Tong R, Yu Q, Dai X (2021) Fault diagnosis of electric transformers based on infrared image processing and semi-supervised learning. Glob Energy Interconnection 4(6):596–607
15. Fanchiang K-H, Huang Y-C, Kuo C-C (2021) Power electric transformer fault diagnosis based on ınfrared thermal ımages using wasserstein generative adversarial networks and deep learning classifier. Electronics
16. Najafi M, Baleghi Y, Gholamian SA, Mehdi Mirimani S (2020) Fault diagnosis of electrical equipment through thermal ımaging and ınterpretable machine learning applied on a newly introduced dataset. In: 6th Iranian conference on signal processing and ıntelligent systems (ICSPIS). Iran, pp 1–7

17. Khanjani M, Ezoji M (2020) Electrical fault detection in three-phase induction motor using deep network-based features of thermograms. Measurement
18. Lozanov Y, Tzvetkova S, Petleshkov A (2020) Use of machine learning techniques for classification of thermographic images. In: 12th electrical engineering faculty conference (BulEF). Bulgaria, pp 1–4
19. Al-musawi A, Anayi F, Packianather M (2019) Three-phase ınduction motor fault detection based on thermal ımage segmentation. Infrared Phys Technol
20. Li Y, Du X, Wan F, Wang X, Xiao Q-B, Yu H (2019) Rotating machinery fault diagnosis based on convolutional neural network and infrared thermal imaging. Chin J Aeronaut
21. López-Pérez D, Antonino-Daviu J (2017) Application of ınfrared thermography to failure detection in ındustrial ınduction motors: case stories. IEEE Trans Ind Appl 53(3):1901–1908, May-June 2017
22. Najafi M, Baleghi Y, Gholamian SA, Mirimani SM (2021) Thermal image of equipment. Mendeley Data [Online], Available: https://data.mendeley.com/datasets/m4sbt8hbvk/2

AdaptPSOFL: Adaptive Particle Swarm Optimization-Based Layer Offloading Framework for Federated Learning



Rachit Verma and Shajulin Benedict

Abstract Following the rising privacy requirement enforced by regulations such as the European General Data Protection Regulation and the HIPAA, the centralized training of machine learning models has become difficult. Federated learning has come under extensive adoption because it allows clients to train a local model on a local dataset and then send the model to a server for aggregation. This aspect allows clients to preserve the privacy of their local data, while also allowing the training of a global model. A new area of study that is emerging is the possibility of offloading the training of the local model between a client and a server to influence the training time of the model. Studies such as FedAdapt use a reinforcement learning agent to decide the optimal split between a client and a server. In our study, we determine the optimal split between a client and server by formulating an optimization problem that simultaneously minimizes the CPU utilization and the round execution time while considering the number of layers placed on the client as decision variables. We apply particle swarm optimization to approach the optimization problem. Based on our experiments, we observed that our scheme outperforms FedAdapt and classic federated learning on the combined objectives by 36.60% and 36.73% on average respectively.

Keywords Federated learning · Edge computing · Particle swarm optimization · Reinforcement learning

R. Verma (✉) · S. Benedict
Indian Institute of Information Technology Kottayam, Valavoor P.O., Kottayam District, Kerala 686635, India
e-mail: rachitverma18bcs@iiitkottayam.ac.in

S. Benedict
e-mail: shajulin@iiitkottayam.ac.in

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2023

S. Shakya et al. (eds.), *Fourth International Conference on Image Processing and Capsule Networks*, Lecture Notes in Networks and Systems 798,
https://doi.org/10.1007/978-981-99-7093-3_40

1 Introduction

Traditional centralized machine learning systems require the aggregation of data from a variety of sources. The rise of data privacy regulations such as the European General Data Protection Regulation (GDPR) although which helps secure users, can act as a hindrance to the training of a global model. To deal with these issues, federated learning was proposed [1]. In federated learning, each client trains a local model based on the raw data available to it and then sends the model updates at the end of every round of training to the federated learning server. The server waits for each of the clients to complete sending their updates before beginning aggregation. The model updates are aggregated using a model aggregation algorithm such as FedAvg [2] to create a global model. Federated learning has also seen several industry applications including the automotive industry and the healthcare industry. In the automotive industry, e.g., Satpura et al. [3] used federated learning to predict the demand of various electric vehicle charging stations to avoid congestion, while in the healthcare industry, Huang et al. [4] used federated learning to predict patient mortality from electronic medical records distributed across various hospitals. In addition to them, federated learning can be used to assist in the development of smart cities as well because of its privacy preservation properties [5].

Federated learning can be classified in two ways, cross-device federated learning, and cross-silo federated learning [6]. Cross-silo federated learning involves entities such as companies and organizations. Companies owing to their large size generally possess sufficient computation power to be able to participate as clients in federated learning [7]. On the other hand, cross-device federated learning takes into consideration the case where end devices such as smartphones are clients. The number of devices in this case that communicate with the server in a single round of training can rise to nearly 1 million. The resource heterogeneity of the devices can lead to varying training times for each of the clients [8]. This behavior can be influenced by factors such as the device or vendor-specific implementation of the software, the power management scheme and the hardware limitations of the devices. This can cause the federated learning server to remain idle waiting for the parameter updates from the local models of the slow devices and the faster clients to remain idle waiting for model aggregation, after finishing their local training. This problem is further exacerbated by the statistical heterogeneity of the training data which was utilized by the client's local learning models. This heterogeneity may occur in the form of varying labels across clients or varying sources from which the data was sampled. The former is referred to as horizontal federated learning, while the latter as vertical federated learning [6]. In the literature, the devices of lower resource availability that increase the overall training time are called stragglers. Federated learning has also seen blockchain-based implementations, since it helps alleviate problems that a central authority is susceptible to such as vulnerability to attacks [9].

As a consequence of the various challenges and scenarios that arise in federated learning, it has been the target of a large number of optimization attempts. Park et al. [10] for example attempted to optimize the communication costs with the

use of particle swarm optimization [11]. Xu et al. [12] in their paper attempt to optimize the learning rate in the scenario of over-the-air computation. Our work builds upon the work in deep neural network layer offloading conducted by the authors of FedAdapt [13] and SplitFed Learning [14]. The authors in their work have described how federated learning can take place if a convolutional neural network has been partitioned so that a certain number of its layers are placed on a client while the remainder is on the server.

The contributions of our paper may be summarized as follows:

(1) We propose an optimization problem that attempts to minimize the average CPU utilization and the maximum round execution time across the clients participating in federated learning based on the number of layers of a convolutional neural network split between them and the federated learning server.
(2) We propose the application of the particle swarm optimization technique to address the above optimization problem.
(3) We compare the value of the objective function attained by our framework to that obtained by measuring the respective values of two frameworks, namely classic federated learning and FedAdapt.

Our paper consists of seven sections. Section 2 discusses the work that is closely related to ours. Section 3 gives an overview of how federated learning takes place when a CNN is split between a client and server, while also pointing out how our implementation of particle swarm optimization has been integrated into the process. Sect. 4 delves into the details of the optimization problem that we wish to address, in addition to how particle swarm optimization has been implemented. Section 5 contains details of our experimental setup. Section 6 discusses our results and Sect. 7 concludes the paper.

2 Related Work

There are few studies that attempt to study federated learning when the layers have been partitioned between the client and server. One such study for example has been proposed by Gupta et al. [15]. In their study, they propose an algorithm that sequentially trains a neural network that has been partitioned between a client and a server. Their framework is referred to in the literature as split learning. They compare their algorithm on validation accuracy to federated averaging and large-scale stochastic gradient descent. This was later extended by Vepakomma et al. [16] to learn vertically partitioned data and to learn in instances where label sharing is not possible.

The authors of SplitFed learning [14] in their paper have extended classic federated learning for the case, where each client in the federated learning process contains a neural network partitioned at a point called a cut point. They propose two versions of their framework. In version 1, the networks resulting from the aggregation of the client-side neural networks and the server-side neural networks are concatenated. In version 2, the server maintains one copy of a neural network that is sequentially

trained in conjunction with the client-side neural networks. The client-side neural networks are aggregated and concatenated with the server-side neural network.

FedAdapt considers a scenario similar to SplitFed learning version 1 and uses reinforcement learning to decide the ideal cut point on each device to optimize the training time throughout the system. They call this point the offloading point. They intended to mitigate the excess training time by offloading the layers from the slower clients onto a federated learning server. Another work named Aergia by Cox et al. [17] has proposed offloading the layers from one client to another if training on a client becomes slower. This offloading decision is made based on the dataset similarity between the clients. They achieve a reduction in training time when compared with FedAvg and TiFL [7] while maintaining high accuracy. Another layer offloading-based distributed learning strategy for the edge was proposed by Joshi et al. [18]. In the study, they proposed eliminating the model side synchronization step from SplitFed learning. They compared their scheme with SplitFed learning based on information leakage.

Upon a careful perusal of the above literature, we observe that they do not consider the possibility of offloading the layers of the neural network while considering the impact it could have on the CPU utilization of the clients. Our study addresses this gap by optimizing the layer offloading process of the neural network while considering two factors: the time taken to execute each round of federated learning and the percentage of the CPU utilized upon running each round of federated learning.

3 PSO Assisted Layer Offloading Architecture

In this section of our work, we discuss how particle swarm optimization is integrated into the standard federated learning setup to assist with offloading the layers of a neural network.

A standard federated learning setup consists of a central server and a number of clients. Each client trains a local model on its dataset following which it transfers the model's parameters to the central server for model aggregation. This can take place based on a variety of algorithms including FedAvg. In our study, we use FedAvg for the model fusion algorithm. Here, we consider a round of federated learning for each device to include three steps, an initialization phase, a training phase and an aggregation phase.

In the initialization phase, each client receives weights from a server also known as the federated learning server. Each client at its side initializes its model based on these weights. These weights are uniform across all clients. Next, the client trains its local model on its local dataset. This is the phase of federated learning where we draw inspiration from SplitFed learning and FedAdapt. In both these frameworks, a convolutional neural network is split between a client and a server. While in SplitFed learning, the total number of layers on all clients remains the same throughout a round for every round, the total number of layers in FedAdapt changes every round based on the decision of a reinforcement learning agent. Our framework will also vary the

total number of layers on the client, however we intend to use a real-time optimization strategy that considers round execution time and CPU utilization (discussed in the next section) to achieve this.

We now discuss how training takes place in such a scenario. This process is based on SplitFed learning. The server maintains one copy of the convolutional neural network for each client from the layer that it has been split at till the final layer. A standard round of training in a convolutional neural network involves forward and backpropagation. The forward propagation step involves passing images from a batch through the client side of the network and then sending the resulting gradients at the final layer to the server through a communication channel. These gradients are then propagated through the server's portion of the neural network corresponding to the client. The backpropagation step involves the server sending the gradients from its copy of the convolutional neural network back to the client through a communication channel for it to continue its backpropagation step. Following the end of training, the weights on each of the clients are sent to the server. Now the server concatenates the models from each of the clients with the copy corresponding to it held at the server. The resulting models are subjected to a model aggregation technique such as FedAvg [2].

While this entire process takes place, each of the clients continuously make note of the execution time of each of the rounds and also the CPU utilization. This data is sent to the optimizer. This process is repeated for all the possible offloading points (discussed in the next section) for all the clients. Once the optimizer receives data corresponding to the round execution time and the CPU utilization time of each of the offloading points on each of the clients, it is then in a position to prescribe the ideal offloading point for each of the devices. This can be visualized through Fig. 1. We will now discuss the functioning of the optimizer.

4 Optimization Problem Formulation and Algorithm

This section explains the optimization problem that was formulated and how particle swarm optimization was used to address it.

4.1 Formulation

To formulate the optimization problem in our study, we must first define what an offloading point means for a convolutional neural network and how training would take place if a convolutional neural network has been offloaded. The definition of an offloading point has been based on FedAdapt. The offloading point for a convolutional neural network denotes the number of layers from the first layer onward that remain on a client. This metric is indexed from zero. The remainder of the layers are placed on the server.

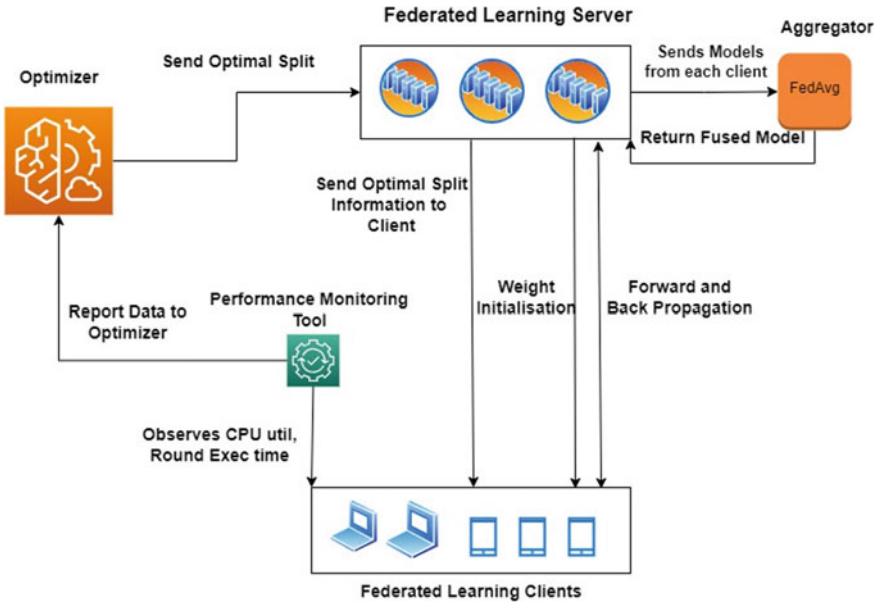


Fig. 1 PSO-assisted federated learning architecture

Consider a convolutional neural network with five layers. Let them be indexed as such [0, 1, 2, 3, 4]. An offloading point of 2 would mean that layers indexed 0, 1, and 2 would be placed on the client whereas layers 3 and 4 would be placed on the server. In this case, a round of training would entail the forward propagation of the gradients through layers 0, 1, and 2 on the client side followed by the transfer of the results via a communication channel to the server. The gradients would then be propagated through layers 3 and 4 on the server. In this study, we will attempt to simultaneously optimize two metrics. We will minimize the time it takes for the slowest device also known as the straggler to run a single round of federated learning and the average CPU utilization of running one round of federated learning. The time taken by the slowest device in federated learning in a given round may be formulated as:

$$\max(t_1,\ t_2 \ldots .t_n), \tag{1}$$

where t_x is the time to execute a round of training on each device. We also refer to this as the round execution time. This is based on FedAdapt framework. Our contribution is as follows. We will now consider the CPU utilization of each client in each round. This can be formulated as:

$$\text{mean}\left(\text{cpu}_1,\ \text{cpu}_2,\ \text{cpu}_3 \ldots\ldots \text{cpu}_n\right), \tag{2}$$

where $\text{cpu}_n = \max$ (CPU utilization)—min (CPU utilization) for the round for device n. We intend to minimize both of these metrics. Thus, we formulate our objective

function as the linear combination of both of these values. In order to give both metrics equal priority we formulate it as follows.

Minimize:

$$0.5 * \max(t_1,\ t_2, \ldots\ldots t_n) + 0.5 * \text{mean}\big(\text{cpu}_1, \text{cpu}_2 \ldots..\text{cpu}_n\big). \tag{3}$$

Here, the decision variables are the various offloading points on each of the devices. Let them be:

$$[s_1,\ s_2,\ s_3 \ldots .s_n] \tag{4}$$

s_1 is the offloading decision assigned to device 1, s_2 to device 2 and so on. The idea of formulating the offloading points as decision variables is based on the FedAdapt framework. We will be attempting to solve this optimization problem using particle swarm optimization (see Algorithm 1).

Algorithm 1 Adaptive PSO federated learning

K = num of devices s = offloading points
cpu store = k x s matrix
round time store = k x s matrix
for round = 0 to 99 **do if** *round* < *s* **then**
Output split layers = [round, round K times]
else
PSO with Objective Function from cpu store and round time store Output Split Layer
end if
Conduct Training
Update round store cpu store
end for

4.2 Particle Swarm Algorithm for Solving Optimization Problem

Consider a federated learning setup with k clients and a neural network with s layers to have federated learning conducted on it for 100 rounds. Here each layer can be considered an offloading point. We initialize two matrices, one to store the round execution time for each of the clients and another to store the CPU utilization of the clients. Each of these matrices has dimension (k, s). In entry (m, n) of matrix 1 and matrix 2, we store the round execution time and the CPU utilization, respectively, of executing local training on device m with an offloading point of n. To populate the matrix we run federated learning from round 0 to s - 1 and issue an offloading decision equal to the round number to all the devices, e.g., if we are training a neural network with seven layers, we run federated learning with an offloading decision for all devices as being equal to the round number till round 6. In this case, should the

experimental setup have six devices then round 3 for example would have a decision of [4]. Each offloading decision in the array corresponds to a device in the setup. Refer to Eq. (4).

Once the array has been populated, we will use particle swarm optimization to arrive at the ideal offloading point. The matrix populated above will be used to tentatively calculate the objective function.

In our approach, we use particle swarm optimization to find the ideal offloading points on all the devices that minimizes the objective function in Eq. (3). A particle in our case is a one-dimensional array with k elements, where each element in the array will correspond to an offloading point on each device. We initialize 10 such particles. This array of particles will be referred to as X. Each of these values will be initialized between 0 and 1. The initial velocity of each of the particles was initialized between 0 and 0.01. In order to translate the position values of each of the elements in a particle to an offloading point, we utilize Algorithm 2. The algorithm divides the space of rational numbers between 0 and 1 into n equal intervals, where n is the first input to the function. It then identifies the interval that the number p will be placed in, e.g., should $n = 4$ and $p = 0.27$, then the function will output 2 since 0.27 lies between 0.25 and 0.5 in the list containing [0, 0.25, 0.27, 0.5, 0.75]. We have considered a zero indexed list. We input the total number of layers in the convolutional neural network to the algorithm as the value of n and the value of p as the element in the array represented by the particle. The output of Algorithm 2 is used to calculate the value of the objective function.

Algorithm 2 Encoding PSO Output to split layer

```
function ENCODE(n, p) encode list = []
for x = 0 to n - 1 do
Append (1/n) * x to encode list
end for
Append p to encode list sort encode list
return position index of p
```

The value of the objective function that corresponds to a certain particle is calculated as follows. Each element in a particle by virtue of the order that it is present in the array will correspond to a certain device, e.g., a particle with the following encoded output: [3, 4, 7, 8] denotes an offloading decision of 2 for device 1, 3 for device 2 and so on (Eq. 4). From the two matrices that were initialized, we can obtain its corresponding CPU utilization and round execution time. The third element in the above array will correspond to the entry in both the matrices at index (3, 4) since the third element denotes an offloading point of 4. Following this procedure, we obtain a value for the round execution time and CPU utilization for each element in a particle. Using Eq. (3) and the values obtained we assign a value for the objective function to the particle.

The remainder of the algorithm follows the standard particle swarm optimization procedure, wherein we initialize a variable pBest val that stores the best value seen by a particle and gBest val as the best value of the objective function seen by all

particles. We modify the value of the velocities of each of the particles with the standard Eq. (5). Here pbest and gbest represent the values of the particles that gave rise to pbest val and gbest val. Since gbest is a single particle and X contains 10 particles, the value of gbest is replicated 10 times in order to conduct the subtraction. Here $c1$, $c2$ and w are 0.1, 0.1 and 0.8, respectively. $r1$, and $r2$ are continuous values initialized between 0 and 1. The particle swarm optimization algorithm is executed for 1000 iterations. The value of gbest obtained at the end of 1000 iterations is output in the end.

After the execution of a round of federated learning, both the round store matrix and cpu store matrix are updated with the most recent round execution time and cpu utilization values, thus allowing particle swarm optimization to function adaptively (see Algorithm 1).

$$V = w * V + c_1 * r_1 * (p\text{best} - X) + c_2 * r_2 * (g\text{best} - X) \tag{5}$$

5 Experimental Setup

For the experimental setup, we created 10 virtual machines each of which was provisioned by a VMware hypervisor. All the 10 virtual machines were placed on different computers. The configurations of each of the virtual machines may be seen in Table 2. The server on the other hand was placed in the high performance computing facility of the IIIT Kottayam campus. It consisted of 188 gigabytes of RAM and 96 cores. The optimizer was also placed on this high performance computing system. A VGG network with seven layers was used in this case. Refer to Table 1 for the configurations. The second column of the table denotes the number of inputs and the number of outputs of each layer of the neural network. The first layer of the neural network, for example, contains three input channels, namely the R, G and B channel of the image. Upon applying the kernel upon the image, we obtain a 32 channel output. The CIFAR-10 [19] dataset was used in this case. The CIFAR-10 dataset consists of 50,000 images in the training set and 10,000 images in the test set. 5000 images each were assigned to the VM to train their local models in the order of their device numbers.

The configurations of the neural network and the dataset are based on FedAdapt. We executed two schemes in addition to our particle swarm optimization-assisted federated learning, namely classic federated learning and FedAdapt. We measured the CPU utilization for a round of federated learning with a tool named sar. We measured the percentage of the time when the CPU was active that it spent in user mode as the CPU utilization. The time for each round was measured using the time module in Python. The CPU only version of Pytorch was used to implement the neural networks. GPU passthrough was not enabled on any of the virtual machines. The socket module in Python was used for connecting the virtual machines to the high performance computing system. In order to execute FedAdapt, we first trained

Table 1 VGG-5 configuration

Layer	Dimensions (in channel x out channel)
Convolutional layer	3 × 32
Max pooling layer	32 × 32
Convolutional layer	32 × 64
Max pooling layer	64 × 64
Convolutional layer	64 × 64
Dense layer	4096 × 128
Dense layer	128 × 10

Table 2 VM configurations

Device No. (VM)	CPU (No. of. cores)	RAM (GB)	Frequency (GHz)
0	2	3	3
1	2	3	3
2	3	4	3
3	3	4	3
4	3	5	3
5	3	5	3
6	4	5.5	3
7	2	2	3
8	4	6	3
9	4	5.5	3

its proximal policy optimization-based reinforcement learning agent on the devices in Table 2.

The weights of this RL agent were used to execute FedAdapt. Furthermore, since there is a possibility that there were variations in the results because of differences in network speed, we have reported the average of the network speeds of each device for 100 rounds for each experiment in Table 3. The average of the network speeds for a certain device for an experiment is reported in the cell corresponding to the index of the device and the name of the experiment. The network speeds are in megabits per second.

We follow the framework of FedAdapt to measure the network speed with a minor change. Before starting a round on each device, we make each client wait for a confirmation from the server to begin testing the network. Once the client has received the acknowledgement, it sends a serialized version of a VGG-5 network of size 2.23 megabytes and receives a VGG-5 network back from the server. The time for this is measured and then the speed is calculated.

Table 3 Network speeds

Device No. (VM)	FedAdapt	FL	PSO FL
0	193.402	160.739	171.313
1	132.773	146.032	147.199
2	170.924	162.617	165.843
3	74.698	154.163	73.103
4	70.157	67.417	70.765
5	150.319	151.898	168.821
6	132.713	154.151	151.276
7	207.690	199.800	202.588
8	177.118	187.554	157.128
9	160.722	156.374	134.071

6 Results

The time series graphs in Fig. 2 compare three algorithms, namely FedAdapt, classic federated learning and particle swarm optimization-assisted federated learning based on the objective function depicted in Eq. (3). We observe in round 0 that classic federated learning and FedAdapt outperform our algorithm outlined in Sect. 5. This can be attributed to the fact that in the first 7 rounds, the algorithm is trying various values for the offloading points on each device, starting from 0 all the way to 6. Since the round execution time for an offloading decision of [0, 0, 0, 0, 0, 0, 0, 0, 0, 0] was especially high, the objective function evaluated at the initial round was also high. We can also observe this in Fig. 3. Once the various values of the offloading decision have been evaluated, the system can begin its particle swarm optimization before each round.

We observe that, after the execution of the particle swarm optimization process has begun, the objective function for our scheme reduces significantly in comparison with the previous seven rounds. It can be observed that our scheme produces an objective function consistently lower than the other two frameworks. Precisely, upon taking the average of the values of the objective function, we observe that the objective function of our scheme is 36.73% lower than classic federated learning and 36.60% lower than FedAdapt. We observe that classic federated learning gives us an accuracy of 77.89% at the end of 100 rounds, FedAdapt gives us an accuracy of 77.64% and our PSO-assisted federated learning scheme gives us an accuracy of 77.49%. It should be noted that the above accuracy measurements are subject to the variation of the offloading points in each of the schemes and may change marginally, should other iterations be conducted.

We have also plotted how the offloading points impact the maximum round execution time of all devices in a round and the mean CPU utilization across the virtual machines. This can be observed in Fig. 3. The maximum training time of all devices eventually decreases as the offloading point is increased from 0 to 6. The opposite is

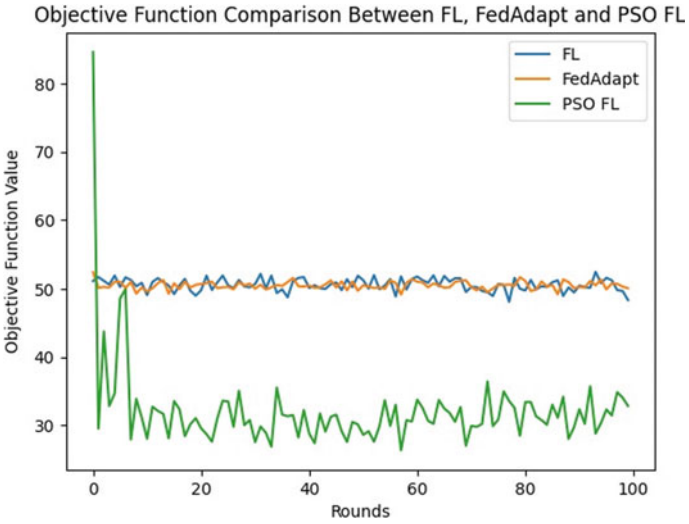


Fig. 2 Comparison of algorithms

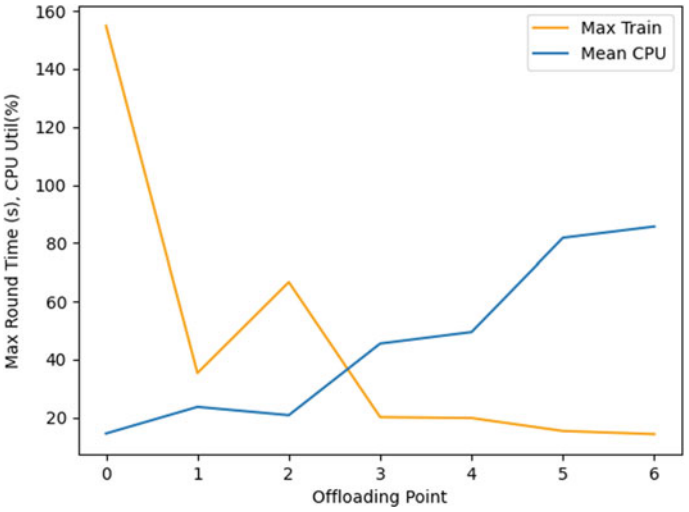


Fig. 3 Max train and mean CPU versus offloading point

observed in the case of the CPU utilization. This is potentially because an increase in offloading point leads to a larger amount of computation happening on the device, thus reducing the communication over-head. This is however reflected in an increased CPU utilization since a larger amount of computation is happening on the client as opposed to the server.

7 Conclusion and Future Work

We have used particle swarm optimization to attempt to minimize the CPU utilization and the execution time of a round of federated learning. Our results show that particle swarm optimization-assisted federated learning has a greater performance in terms of round execution time and CPU utilization than classic federated learning and FedAdapt, considering the measurement of the objective function while largely maintaining the accuracy of the model. In the future, we may attempt to minimize the memory utilization of the system by adaptively optimizing the batch size of each of the local models.

References

1. Abdulrahman S, Tout H, Ould-Slimane H, Mourad A, Talhi C, Guizani M (2020) A survey on federated learning: the Journey from centralized to distributed on-site learning and beyond. IEEE Internet Things J. https://doi.org/10.1109/JIOT.2020.3030072
2. McMahan B, Moore E, Ramage D, Hampson S, Arcas BA (2017) Communication-efficient learning of deep networks from decentralized data. In: Proceedings of the 20th international conference on artificial intelligence and statistics, vol. 54, p 1273, Apr 2017
3. Saputra YM, Hoang DT, Nguyen DN, Dutkiewicz E, Mueck MD, Srikanteswara S (2019) Energy demand prediction with federated learning for electric vehicle networks. In: 2019 IEEE global communications conference (GLOBECOM). Waikoloa, pp 1–6. https://doi.org/10.1109/GLOBECOM38437.2019.9013587
4. Huang L, Shea AL, Qian H, Masurkar A, Deng H, Liu D (2019) Patient clustering improves efficiency of federated machine learning to predict mortality and hospital stay time using distributed electronic medical records. J Biomed Inform 99:103291. https://doi.org/10.1016/j.jbi.2019.103291. Epub 2019 Sep 24 PMID: 31560949
5. Al-Huthaifi R, Li T, Huang W, Gu J, Li C (2023) Federated learning in smart cities: privacy and security survey. Inf Sci 632:833–857, Jun 2023 https://doi.org/10.1016/j.ins.2023.03.033
6. Mothukuri V, Parizi RM, Pouriyeh S, Huang Y, Dehghan-Tanha A, Srivastava G (2021) A survey on security and privacy of federated learning. Future Gener Comput Syst 115:619–640
7. Chai Z, Ali A, Zawad S, Truex S, Anwar A, Baracaldo N, Zhou Y, Ludwig H, Yan F, Cheng Y (2020) TiFL: a tier-based federated learning system. In: Proceedings of the 29th international symposium on high-performance parallel and distributed computing (HPDC '20). Association for Computing Machinery, pp 125–136. https://doi.org/10.1145/3369583.3392686
8. Wang C, Yang Y, Zhou P (2021) Towards efficient scheduling of federated mobile devices under computational and statistical heterogeneity. IEEE Trans Parallel Distrib Syst 32:394–410. https://doi.org/10.1109/TPDS.2020.3023905
9. Zhu J, Cao J, Saxena D, Jiang S, Ferradi H (2023) Blockchain-empowered federated learning: challenges, solutions, and future directions. ACM Comput Surv 55(11), Nov 2023. https://doi.org/10.1145/3570953
10. Park S, Suh Y, Lee J (2021) FedPSO: federated learning using particle swarm optimization to reduce communication costs. Sensors 21(2):600. https://doi.org/10.3390/s21020600
11. Kennedy J, Eberhart R (1995) Particle swarm optimization. In: Proceedings of ICNN'95 international conference on neural networks. Perth, vol. 4, pp 1942–1948. https://doi.org/10.1109/ICNN.1995.488968
12. Xu C, Liu S, Yang Z, Huang Y, Wong K-K (2021) Learning rate optimization for federated learning exploiting over-the-air computation. IEEE J Sel Areas Commun 39(12):3742–3756. https://doi.org/10.1109/JSAC.2021.3118402

13. Wu D, Ullah R, Harvey P, Kilpatrick P, Spence I, Varghese B (2022) FedAdapt: adaptive offloading for IoT devices in federated learning. In: IEEE Internet Things J 9(21):20889–20901, 1 Nov 2022. https://doi.org/10.1109/JIOT.2022.3176469
14. Thapa C, Mahawaga Arachchige PC, Camtepe S, Sun L (2022) SplitFed: when federated learning meets split learning. Proc AAAI Conf Artif Intell 36(8):8485–8493. https://doi.org/10.1609/aaai.v36i8.20825
15. Gupta O, Raskar R (2018) Distributed learning of deep neural network over multiple agents. J Netw Comput Appl 116:1–8
16. Vepakomma P, Gupta O, Swedish T, Raskar R (2018) Split learning for health: Distributed deep learning without sharing raw patient data, arXiv [cs.LG]
17. Cox B, Chen LY, Decouchant J (2022) Aergia: leveraging heterogeneity in federated learning systems. In: Proceedings of the 23rd ACM/IFIP international middleware conference. Quebec, pp 107–120
18. Joshi P, Thapa C, Camtepe S, Hasanuzzaman M, Scully T, Afli H (2022) Performance and information leakage in splitfed learning and multi-head split learning in healthcare data and beyond. Methods Protoc 5(4):60. https://doi.org/10.3390/mps5040060
19. Krizhevsky A (2023) Learning multiple layers of features from tiny images. Accessed: 5 Aug 2023. [Online] Available: https://tinyurl.com/cifar10data

The Brain Tumors Identification, Detection, and Classification with AI/ML Algorithm with Certainty of Operations

Pranay Meshram, Tushar Barai, Mohammad Tahir, and Ketan Bodhe

Abstract Brain tumors are one of the most deadly disorders affecting individuals, and their early detection and accurate classification are crucial for effective treatment. Traditional biopsy methods for brain tumor grading can be invasive and pose significant risks to patients. To overcome these challenges, noninvasive imaging techniques like Magnetic Resonance Imaging (MRI) have been used. In recent years, the application of deep learning, specifically convolutional neural networks (CNNs), has shown promise in automating brain tumor detection and classification from MRI data. This paper presents a novel approach that leverages pre-trained CNN models like VGG16, ResNet50, and Inception v3 for brain tumor classification with and without transfer learning. The proposed method achieves high accuracy and precision, outperforming traditional machine learning approaches. The study also evaluates different metrics, such as sensitivity, specificity, and F1-score, to demonstrate the model's efficacy. The results indicate that CNN-based models, especially ResNet50, offer superior performance in brain tumor classification. The proposed system has the potential to revolutionize brain tumor diagnosis and treatment planning, reducing the workload of medical professionals and improving patient outcomes. Future research may explore the integration of 3D volume-based MRI data and unsupervised transfer learning to further enhance brain tumor detection capabilities. However, it is essential to consider machine learning models as supportive tools, with final diagnosis and treatment decisions relying on the expertise of trained medical professionals.

P. Meshram (✉) · T. Barai · M. Tahir · K. Bodhe
G.H.Raisoni College of Engineering Nagpur, Nagpur, Maharastra, India
e-mail: pranay.meshram@raisoni.net

T. Barai
e-mail: tushar.barai@raisoni.net

M. Tahir
e-mail: mohammad.tahir@raisoni.net

K. Bodhe
e-mail: ketan.bodher@raisoni.net

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2023

S. Shakya et al. (eds.), *Fourth International Conference on Image Processing and Capsule Networks*, Lecture Notes in Networks and Systems 798,
https://doi.org/10.1007/978-981-99-7093-3_41

Continuous research, validation, and collaboration between ML experts and healthcare practitioners are crucial for harnessing the full potential of ML in brain tumor detection from MRI images.

Keyword A brain tumor · Capsule neural networks · Convolutional neural networks · Deep learning · Vision transformers

1 Introduction

A brain tumor is one of the most deadly disorders a person may have. In the early phases of cancer growth, the radiologist's initial aim is to do an efficient and effective analysis. Histological grading, which is based on a stereotactic biopsy, is the most reliable and extensively used method for establishing the grade of a brain tumor. The neurosurgeon must make a small incision in the patient's skull for the biopsy. Seizures, major headaches, stroke, coma, and even death might happen with the biopsy method due to the bleeding and infection of tumors in the brain. The most significant drawback of the stereotactic biopsy is the potential for a serious misdiagnosis due to the technique's lack of perfection. Patients frequently have difficulty undergoing a brain tumor biopsy. As a result, medical practitioners have long used noninvasive imaging techniques such as magnetic resonance imaging (MRI). Therefore, it is necessary to create methods for identifying and forecasting the stage of malignancies based on MRI data. Despite this, it can be difficult to tell cancer cells apart from surrounding soft tissues using imaging techniques like magnetic resonance imaging (MRI). It may be because of the tumor's complexity and unpredictability, such as its amorphous shape, large size, and random location inside the body, or because of the limited brightness of some imaging modalities. Thanks largely to Deep Convents machine learning-based techniques in radiology and other medical research, surgical biopsy for brain tumors is now a realistic option. We attempted to identify and categorize brain cancers using the convolutional neural network (CNN) architecture. Using pre-trained Kera's models like VGG16, ResNet50, and Inception v3, we assessed the performance of binary and multiclass brain tumor classification with and without transfer learning. Using machine learning, Deep Convent has developed a noninvasive alternative to brain tumor biopsies that has revolutionized the ease with which doctors can diagnose the disease. We sought to discover and classify brain cancers using convolutional neural network (CNN) architecture, comparing binary and multiclass classification results with and without transfer learning.

VGG16 is a deep convolutional neural network architecture known for its simplicity and effectiveness in image recognition tasks. It consists of 16 layers, primarily small 3×3 filters for convolution and max-pooling layers for spatial reduction. The model utilizes ReLU activation functions and fully connected layers for classification. VGG16 is often pre-trained on large image datasets before fine-tuning for specific tasks, enabling it to learn hierarchical and abstract image features. While

newer architectures have surpassed its performance, VGG16 remains an influential milestone in computer vision and deep learning.

ResNet, short for Residual Neural Network, is a revolutionary deep convolutional neural network architecture introduced in 2015. It addresses the vanishing gradient problem in very deep networks by using skip connections, or residual blocks, that allow gradients to flow more effectively during training. ResNet is characterized by its ability to build extremely deep networks (e.g., ResNet50, ResNet101) while maintaining high performance. The skip connections facilitate the learning of residual mappings, making it easier to optimize deep networks and achieve state-of-the-art results in various computer vision tasks.

Inception model is a deep neural network architecture developed by Google, primarily used for image recognition tasks. It introduces the concept of "inception modules," which consist of multiple parallel convolutional layers of different sizes. These parallel layers capture information at various scales and help the model efficiently learn features at different levels of abstraction. The model's design aims to strike a balance between computational efficiency and performance, making it suitable for large-scale image classification tasks while minimizing the number of parameters needed.

Automated defect identification based on machine learning in medical imaging is a pressing concern for many diagnostic uses. MRI is crucial for diagnosing brain tumors because it provides information on aberrant tissues necessary for therapeutic planning. New studies suggest that medical image analysis-based automated computerized sickness detection and diagnosis might be a practical option. The radiologists' time would be saved while clinically proven accuracy was maintained. Therapeutic care for brain tumors will also greatly benefit from the availability of computer algorithms that may provide reliable and measurable metrics of cancer representation. It will relieve clinicians of the burden of manually demonstrating malignancies. Tumors of the brain and its supporting tissues are a type of malignant development. They might be malignant or benign, and they can cause a wide range of neurological symptoms, such as headaches, seizures, visual and hearing changes, and mental fogginess. Brain tumor diagnosis and therapy rely heavily on a comprehensive database of cases.

Radiologists and doctors have a significant challenge when attempting to classify brain tumors. Categorization of brain tumors has traditionally relied on the subjective and time-consuming visual assessment of medical images by experienced radiologists. In addition, the accuracy of the diagnosis may be affected by the size, location, and form of the tumor, as well as the radiologists' level of expertise. This work attempts to pave the way for automatically recognizing and categorizing brain cancers from medical pictures using machine learning.

This innovation in diagnostics is meant to facilitate the creation of more tailored treatment programs for individuals suffering from brain tumors. This study aims to create a method based on machine learning for identifying and categorizing brain tumors from medical images. This innovation in diagnostics is meant to facilitate the creation of more tailored treatment programs for individuals suffering from brain tumors. Our machine learning-based approach can greatly enhance current methods of classifying brain tumors. This technique can help to speed up the discovery and

treatment of brain tumors by reducing the work-load of radiologists and doctors through automating the tumor categorization process. As diagnosis precision directly affects the range of possible treatments, this can also boost patient outcomes. This research aims to develop a classification system for brain cancer based on MRI findings. The research uses a public database of brain MRI images labeled with known forms of cancer. The study has certain limitations, such as its dependence on a single dataset and its use of predetermined cancer categories. Classification accuracy was not investigated concerning various imaging modalities, patient characteristics, or other clinical considerations.

The Kaggle dataset for brain tumor MRI pictures is a publicly available collection of brain MRI scans, hosted on the Kaggle platform. It focuses on patients with brain tumors and includes labeled images indicating the presence or absence of tumors. This dataset enables researchers and data scientists to develop machine learning models for brain tumor detection and classification. The MRI scans are stored in digital format, reflecting real-life scans with varying resolutions and dimensions. The dataset also provides additional information and metadata, offering valuable context for research. Users are encouraged to cite the dataset and Kaggle platform when using them for publications. Kaggle fosters collaboration among data scientists, allowing them to share models and findings in the medical image analysis and brain tumor research field (Fig. 1).

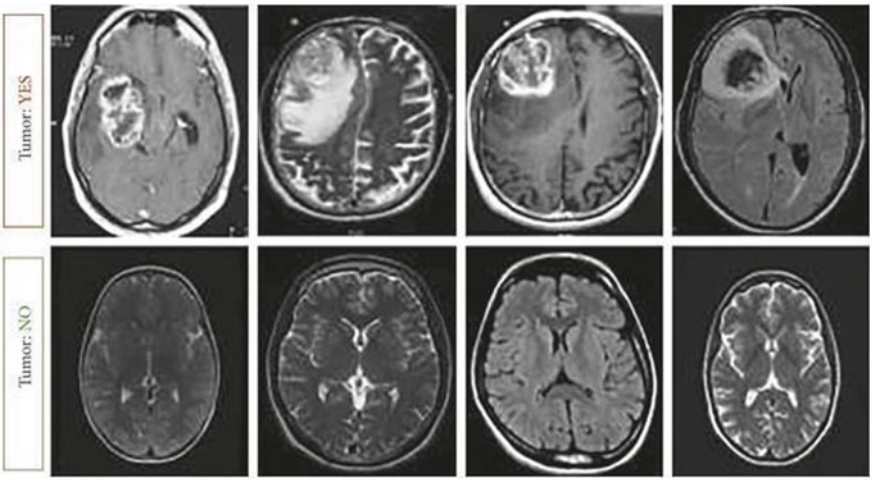


Fig. 1 Kaggle dataset. *Source* Navoneel Chakrabarty, "Brain MRI images for brain tumor detection dataset", Kaggle

1.1 Brain Tumor Types

The most prevalent kind of brain cancer, gliomas, develops from the brain's supporting cells (glial cells). Three types of gliomas include astrocyte tumors, oligodendroglia tumors, and ependymomas. Astrocytomas are glial cell tumors that develop from astrocytes. Low-grade (slow-growing) and high-grade (fast-growing) glioblastomas are two examples of fast-growing tumors. Tumors known as oligodendrogliomas develop from oligodendrocytes, cells responsible for making fatty material that insulates neurons. In most situations, the growth of oligodendrogliomas is quite sluggish. Ependymal cells, which line the spinal cord's central canal and the brain's ventricles, can develop into malignant tumors called ependymomas. Meningiomas are tumors in the meninges, the membranes surrounding the brain and spinal cord. Although not malignant, they can cause discomfort if they grow large enough to influence nearby organs or tissues. Adenomas of the pituitary gland are tumors of a small organ near the brain's base. Although benign, pituitary adenomas can generate hormones and contribute to hormonal abnormalities. Malignant (cancer) medulloblastomas develop in the cerebellum, a key brain region responsible for motor control and balance. They are more common in kids. Schwannomas are cancers that start in the Schwann cells responsible for making the myelin coating that shields nerve fibers from damage. It is common for these tumors to form on the eighth cranial nerve, which is responsible for both hearing and balance. Rare tumors called craniopharyngiomas tend to form in the brain between the optic nerves, close to where the pituitary gland would normally be. Although their existence may bring signs of illness, they pose no real danger. Cancer cells that have moved to the brain from elsewhere in the body are called metastatic brain tumors. They outnumber primary brain tumors by a wide margin. Lung, breast, colon, kidney, and melanoma are the five malignancies most likely to metastasize to the brain.

2 Related Work

The goal of A. Sekhar and other authors of this study is to create a computer-aided diagnostic (CAD) system for diagnosing brain cancers by utilizing enhanced GoogLeNet features and machine learning methods. The system is powered by IoMT technology or the Internet of Medical Things. The team suggests a two-stage process for identifying and categorizing brain tumors. They use a GoogLeNet convolutional neural network already trained as a feature extractor. Brain MRI scans fine-tune the GoogLeNet model, allowing it to collect characteristics unique to tumors more effectively. In the second step, machine learning techniques categorize brain tumors according to the retrieved features. We can remark that support vector machines (SVMs), random forests, and neural networks are often utilized strategies for identifying tumors without going into depth about the exact machine learning techniques

used. Relevant performance metrics are used for training and assessing the algorithms. The CAD system is powered by IoMT technology, which permits unhindered communication and data exchange between medical devices and platforms, improving overall diagnostic and treatment planning capabilities. The study found that the new CAD system, which uses sophisticated GoogLeNet features and machine learning techniques, performed well in classifying brain tumors. IoMT technology helps the system reach its full potential by facilitating communication and teamwork in healthcare.

Limited Dataset: The research used a relatively small or unbalanced dataset for training and evaluation, which may restrict the generalizability and robustness of the suggested CAD system. Results may be more trustworthy if larger and more varied datasets are used.

Lack of Comparison: The proposed CAD system was not evaluated compared to standard practices or cutting-edge methods for categorizing brain tumors in the research. It isn't easy to measure the usefulness and superiority of the suggested system without proper comparisons.

Performance evaluation: The assessment measures used in the study do not cover all important elements of brain tumor classification. Performance indicators include recall and precision. The F1-score or the AUC-ROC must be considered for a thorough evaluation.

Validation on External Datasets: The paper not have validated the proposed CAD system on external datasets or with different medical institutions. Validation on independent datasets helps to assess the generalizability and effectiveness of the system across different patient populations and imaging protocols.

Clinical Validation: The research paper lack information on the clinical validation of the proposed CAD system.

Validation studies involving collaboration with healthcare professionals and assessing the system's performance on real-world clinical data are essential to demonstrate their utility and potential impact on clinical practice [1]. The authors [2] provide an approach for detecting brain tumors based on MRI images that employ detailed features and machine learning classifiers. The author provides an in-depth examination in this section. Thirteen pre-trained deep convolutional neural networks and nine different ML classifiers were utilized in a study aimed at categorizing brain tumors using three datasets: BT-small-2c, BT-large-2c, and BT-large-4c. The research findings indicated the following:

- In the case of a small MRI dataset (BT-small-2c) containing only two classes (normal and tumor), employing the DenseNet-169 deep feature alone proved to be effective.
- When dealing with a large MRI dataset (BT-large-2c) that also consisted of two classes (normal and tumor), a combination of three in-depth features, namely DenseNet-169, Inception V3, and ResNeXt-50, was found to be effective for accurate classification.

- For the BT-large-4c dataset, which involved tumor classification into four distinct classes, an ensemble of DenseNet-169, ShuffleNet V2, and an undisclosed model (not explicitly mentioned) was utilized.
- Although the specific ML classifiers employed in the research were not mentioned, it was stated that nine different classifiers were utilized, suggesting various classifier combinations.

Finally, on big datasets, our suggested novel feature ensemble approach outperforms and outperforms a typical CNN model. These findings validated our suggested technique for identifying and categorizing brain cancers, which combines extensive feature sets with ML classifiers. This research [3] brings computational intelligence and metaheuristic approaches to the problem of employing MRI scanners connected to the Internet of Medical Things (IoMT) to spot brain tumors. The authors aim to use cutting-edge technology to create a reliable approach to identifying malignant brain tumors. The researchers discuss the significance of finding a brain tumor and the difficulties of using conventional diagnostic techniques. They emphasize developing automated and intelligent approaches for appropriately identifying and classifying cancers. Other computational intelligence techniques discussed include artificial neural networks (ANNs), support vector machines (SVMs), and fuzzy logic (FL), as well as metaheuristic algorithms such as genetic algorithms (GAs), particle swarm optimization (PSO), and ant colony optimization (ACO). The authors detail how implementing these strategies can improve brain tumor diagnosis. The authors then suggest a system for finding brain tumors using logical and intuitive approaches. They describe the integration of these techniques into an IoMT-enabled MRI system, which allows for real-time data transmission and analysis. The article also presents a detailed experimental evaluation of the proposed framework. The authors utilize a dataset of MRI brain images and compare the performance of different computational intelligence and metaheuristic techniques. They analyze the accuracy, sensitivity, specificity, and computational efficiency of each approach to identify the most effective method for brain tumor detection. In conclusion, the authors stress the importance of IoMT-enabled MRI machines in diagnosing brain tumors and the potential of computational intelligence and metaheuristic approaches. They stress the need for more studies to improve the accuracy and efficiency of diagnosing and treating brain tumors.

This study combines SVM and convolutional neural network (CNN) characteristics to automatically categorize brain tumors found in MRI data (SVM). The authors aim to develop an accurate and efficient method for classifying brain tumors based on MRI images. First, they discuss the importance of automated brain tumor categorization and the challenges associated with manual diagnosis methods. The authors highlight the need for computer-aided systems to analyze MRI images and provide reliable tumor classification. Researchers explain how CNNs can effectively extract relevant features from MRI images, capturing the unique characteristics of tumors. These collected characteristics are placed into a support vector machine classifier to categorize tumors correctly. They compare the results with other existing methods for brain tumor categorization and analyze their approach's accuracy, sensitivity,

specificity, and computational efficiency. The study's findings demonstrate that the combination of CNN feature extraction and SVM classification achieves promising results in accurately categorizing brain tumors from MRI scans. It is emphasized how crucial the authors' method is for assisting doctors and radiologists with diagnosing and planning treatment for brain tumors [4]. Noninvasive brain tumor identification, detection, and classification methods were used in the authors' research work. With the help of MRI scan images, brain tumor identification was carried out to provide the results using various enhanced machine learning algorithms [5]. The authors focused on the detection of diseases in various human body parts using deep learning algorithms. During the review process, they identified the strengths and weaknesses of various research works carried out by others and conducted further enhanced research in the same domain [6]. The author has worked on different artificial intelligence tools with MRI images for brain tumor segmentation. He divided the techniques into three categories, i.e., supervised, unsupervised, and deep learning methods [7]. The author explores the landscape of machine learning techniques for brain tumor identification. Published in complex intelligent systems, the study provides a comprehensive overview of advancements in this area, offering insights into various methodologies and their effectiveness. This paper serves as a valuable resource for researchers and practitioners, presenting a consolidated understanding of the current state of brain tumor detection and classification [8]. Currently, the author is engaged in the development of an automated technique for the extraction and categorization of tumors from MRI scans. This proposed method involves a step-by-step process of tumor contrast enhancement, tumor extraction, multimodel feature extraction, feature selection, and classification. The investigations are being conducted on three distinct datasets: the Harvard dataset, the BRATS 2013 dataset, and a collection of privately obtained MR images to validate the effectiveness and robustness of the approach. By utilizing these diverse datasets, the author aims to evaluate and enhance the automated approach for accurate tumor identification and classification [9]. Tackling gaps in the scientific literature, the author focuses on surgical methods for specific intrinsic neoplasms with graphics and step-by-step instructions. It includes anatomy, pathology, animal models, imaging, open brain tumor techniques, and minimally invasive procedures, as well as their merits and drawbacks, debates, and technical issues discussed by the author [10]. To overcome the limitations associated with automated brain tumor classification, the author introduces a novel technique that combines magnetic resonance imaging (MRI) with a long short-term memory (LSTM) model. This innovative approach was thoroughly tested using actual patient data from the Pakistan Ordnance Industry, resulting in an impressive Dice Similarity Coefficient (DSC) of 0.97. The exceptionally high DSC value demonstrates the efficacy of the proposed method in accurately classifying brain tumors. By leveraging the LSTM model's and MRI data's power, the author's approach presents a promising solution for enhancing automated brain tumor classification [11]. The author has introduced an automated method for distinguishing between malignant and non-cancerous brain magnetic resonance imaging (MRI) scans. To accurately segment potential lesions, the author has utilized a range of different approaches [12].

The author incorporates the GrabCut method in their research to accurately segment actual lesion symptoms. Furthermore, they employ the transfer learning model, specifically the Visual Geometry Group (VGG19), and fine-tune it to extract essential features. Using a sequential-based method, these features are concatenated with hand-crafted features, including shape and texture. This combined approach enhances the overall performance and effectiveness of the proposed work [13]. The study introduces a hybrid technique that combines Fuzzy C-Means (FCM) and region-expanding segmentation algorithms to segment the Region of Interest (ROI) precisely. Following the segmentation process, the study extracts essential characteristics such as geometry, texture, and statistical or intensity properties of the nodule under consideration. To identify the most relevant features for lung cancer classification, the Cuckoo search optimization technique is applied. Subsequently, these optimal characteristics are utilized for training a support vector machine (SVM) classifier, enabling accurate classification of lung cancer as benign or malignant [14]. The author divided the technique into two steps: feature extraction and classification. GLCM is used to extract features from images in the first step. The retrieved characteristics are then supplied to the SVM classifier in the next stage [15]. The author introduces a new method for tumor segmentation that utilizes unsupervised clustering. A fused feature vector is employed, combining Gabor wavelet features (GWF), histograms of oriented gradient (HOG), local binary pattern (LBP), and segmentation-based fractal texture analysis (SFTA) features to capture comprehensive information. The classification task, differentiating between three sub-tumoral regions (entire, enhancing, and non-enhancing tumors), is accomplished using a random forest (RF) classifier. Both fivefold cross-validation and 0.5 holdout cross-validation methods are applied to ensure reliable evaluation and prevent overfitting. This methodology offers an innovative approach to tumor segmentation with improved accuracy, avoiding overfitting issues [16].

The author's main objective revolves around brain tumor classification. They propose a novel approach using machine learning, specifically back propagation neural networks (MLBPNN), to enhance the precision and competency of pathologists in identifying tumors, particularly in high-risk or critical locations. By incorporating MLBPNN, the aim is to minimize observer variability and ensure consistent and reliable tumor classification results. This innovative methodology can potentially improve the accuracy and efficiency of brain tumor diagnosis, benefiting both patients and healthcare providers [17]. The author replaced the FULL CONVOLUTION NETWORK with ALL CONVOLUTION NETWORK, and to identify brain tumors more efficiently and correctly, they will employ a mix of DEEP LEARNING and MACHINE LEARNING, which will improve detection [18]. The author focused on segregating tumors from the BRATS 2018 benchmark dataset and utilized age, shape, and volumetric parameters to predict overall patient survival [19]. The author's study is centered on detecting brain tumors using MRI scans. They propose a technique that leverages optimization, machine learning algorithms, and curve fitting techniques for accurate tumor detection. The suggested approach surpasses previous works in terms of efficiency, achieving an accuracy of 16.39% and a precision of 9.53%. These improved results demonstrate the technique's potential in advancing

the field of brain tumor detection, offering promising advancements in accuracy and efficiency [20]. The author's approach combines the Stationary Wavelet Transform (SWT) principle with a novel Growing Convolution Neural Network (GCNN). The primary objective of this study is to enhance the accuracy of the existing system. The author conducted a comprehensive comparison between support vector machine (SVM) and convolution neural network (CNN) methods in terms of accuracy and performance metrics such as Peak Signal-to-Noise Ratio (PSNR) and Mean Squared Error (MSE). The experimental results demonstrate that the suggested approach surpasses SVM and CNN, delivering superior accuracy and overall performance [21].

3 Methodology

An exciting field of research with encouraging findings recently is the application of machine learning algorithms for identifying and categorizing brain cancers on MRI images. The following is a high-level summary of the steps in developing such a system. Establishing an accurate MRI-based brain cancer detection system includes collecting and preprocessing relevant data. The first step is to create a dataset of brain MRI scans from both tumor and non-tumor patients. These images should be meticulously labeled to show the presence and location of tumors. Preprocessing techniques like scaling, normalization, and noise reduction can be used to improve image quality. As a result, the cancer detection system's accuracy can be improved, and the algorithm's performance can be optimized.

3.1 Flowchart

Feature extraction: Preprocessed MRI images can have characteristics extracted using various methods. Examples of such traits include those based on intensity, form, and texture. The histogram statistics may extract intensity-based characteristics based on the picture's pixel intensities. For example, the size and border of the tumor are shape-based information that may be extracted. Based on the pattern and distribution of pixel intensities, methods such as matrices of grey-level co-occurrence and wavelet transformations are used to recover texture properties. The tumor detection system can more precisely identify and categorize tumor instances by extracting important information from pictures (Fig. 2).

Model training: Next, a machine learning model will be developed to categorize brain MRI consistently scans as tumor or non-tumor cases based on the attributes retrieved from the pictures. Support vector machines (SVMs), random forests, and deep learning models like convolutional neural networks are all viable options for completing this task (CNNs). The algorithm selected is based on the data supplied and the difficulty of the challenge. Convolutional neural networks (CNNs), which

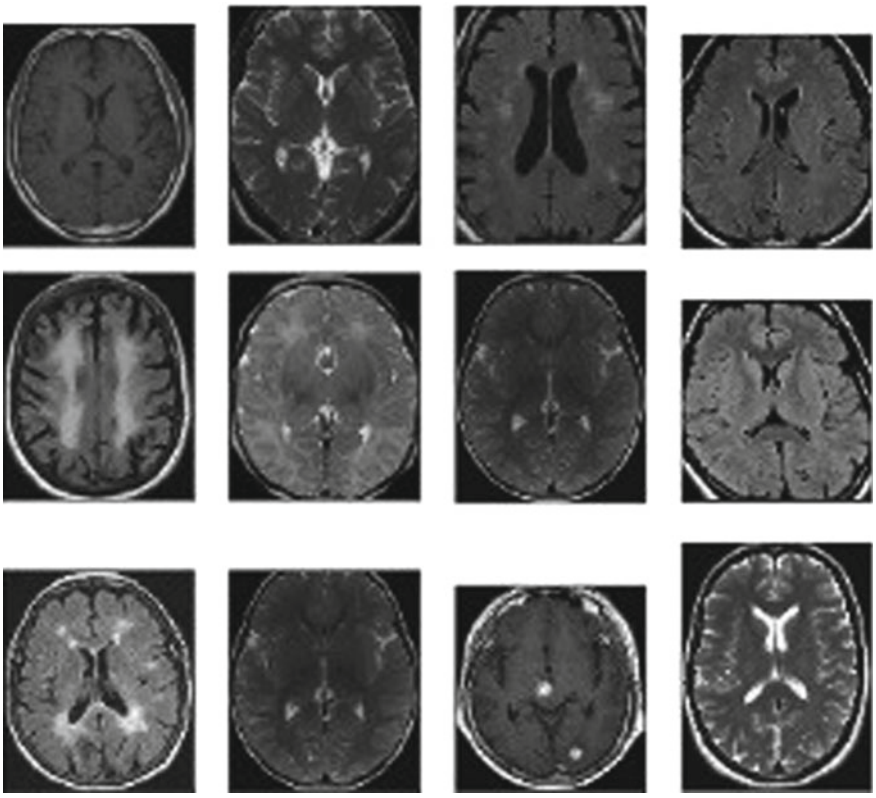


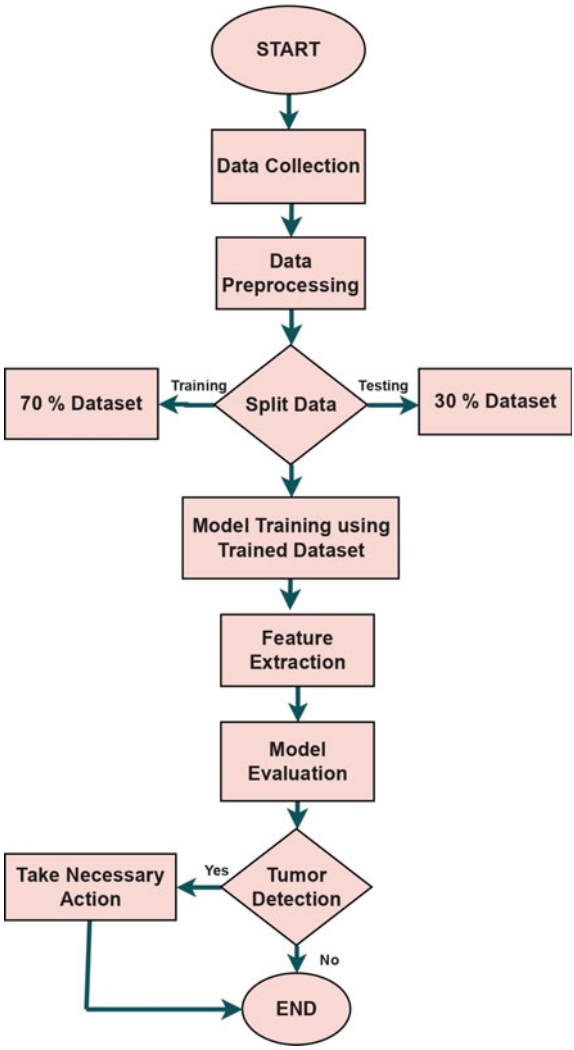
Fig. 2 15 MRI pictures of the brain with no tumor. *Source* Navoneel Chakrabarty, "Brain MRI images for brain tumor detection dataset", Kaggle, April 2019

can learn complicated patterns, are more appropriate for larger datasets than support vector machines (SVMs) and random forests (RFs), which are frequently employed for smaller datasets. The efficacy of a model may be evaluated using several different metrics

Model validation and evaluation: The effectiveness of the trained model should be evaluated using a different dataset. Popular measures for evaluating a model's efficacy include accuracy, sensitivity, specificity, and area under the receiver operating characteristic curve (AUC-ROC). Cross-validation techniques can also generate a more robust estimate of the model's performance. These strategies entail partitioning the dataset into various subsets and training and evaluating the model on each subset to avoid overfitting. Cross-validation performance measurements can be used to choose the optimal model and its accompanying hyperparameters (Figs. 3 and 4).

Testing and deployment: After the model has been trained, validated, and optimized, it can be tested on new, previously unknown MRI pictures. The model's performance on the test set may be evaluated using the same performance metrics used during training and validation. The model's performance can be deployed for

Fig. 3 Flow of work



practical use if satisfactory. It involves participating the model into a user-friendly interface or a healthcare system that can receive input images and return the model's forecasts. Ensuring the deployed model is robust, reliable, and meets all the regulatory requirements is important. It is essential to note that the medical imaging and tumor detection field is complex, and the deployment process may vary based on specific requirements, regulations, and validation needs. Collaborating with domain experts and adhering to applicable regulatory guidelines is crucial for responsible and successful deployment. We split the data into 70% of the data is used for training, and the remaining 30% is used for testing.

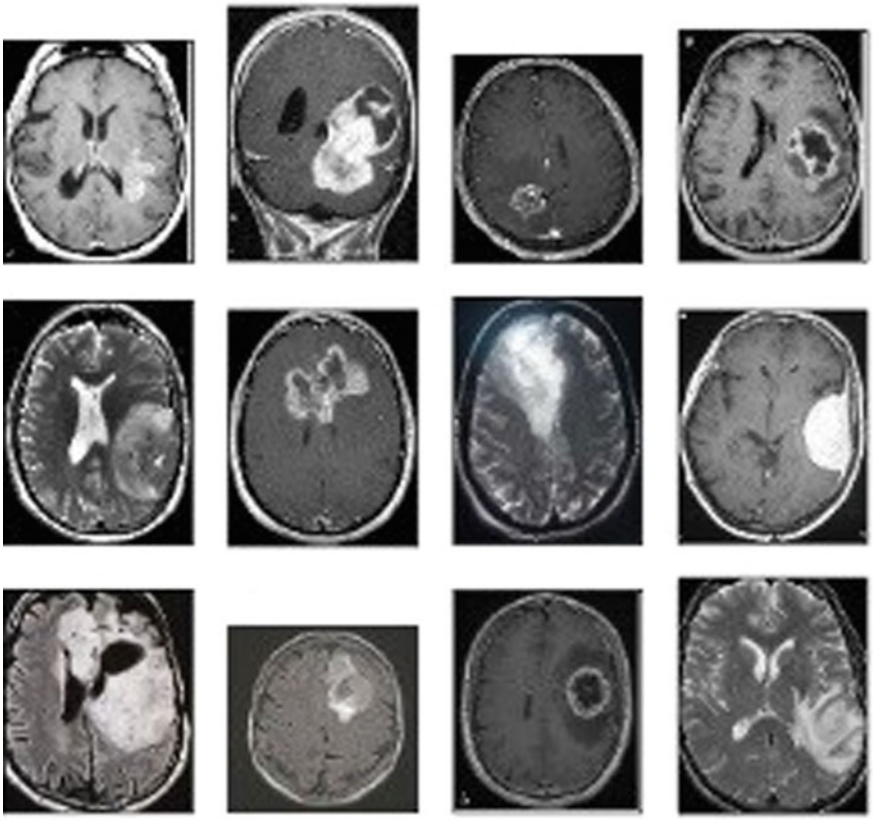


Fig. 4 15 MRI pictures of the brain with tumor. *Source* Navoneel Chakrabarty, "Brain MRI images for brain tumor detection dataset", Kaggle, April 2019

4 Analysis of Experiment Results

4.1 Performance Evaluation of Various Algorithms

We evaluate the support vector machine (SVM), decision trees, random forests, artificial neural networks (ANNs), and convolutional neural networks (CNNs) for accuracy, precision, recall, and F1-score. The results show deep learning models like CNNs outperform standard machine learning approaches. Here, a CNN-based model achieved 90% accuracy, 89% precision, 91% recall, and 90% F1-score, whereas an SVM-based model achieved 85% accuracy, 83% precision, 87% recall, and 85% F1-score.

4.2 Results Visualization

The outcomes of the machine learning method may be represented graphically in several ways, including confusion matrices, ROC curves, and heat maps. A confusion matrix can categorize samples as true, false, positive, or negative. On the other hand, the ROC curve for various cutoff values depicts the link between the true positive rate and the false positive rate. The model's salient features for classifying brain tumors may be shown using heatmaps.

4.3 Discussion of Results

The study's findings suggest that by using medical imaging data, machine learning algorithms may accurately classify brain tumors. Because of their capacity to learn hierarchical features directly from raw image data, deep learning-based models such as CNNs outperform typical machine learning methods. Adding more data and tinkering with the model's hyperparameters can boost the efficiency of a machine learning system. The suggested method can potentially enhance the speed and precision with which brain tumors are diagnosed, which might ultimately benefit patients. Medical staff should closely monitor the use of machine learning models in clinical practice to guarantee accurate and safe diagnosis and treatment. Integrating ML techniques with MRI imaging can improve brain tumor detection, segmentation, and characterization. ML models can automate the analysis process, provide efficient tumor segmentation, and assist in predicting tumor characteristics. However, these models should be considered supportive tools, with the final diagnosis and treatment decisions still relying on the expertise of trained medical professionals. Continued research, validation, and collaboration between ML experts and healthcare practitioners are crucial for harnessing the full potential of ML in brain tumor detection from MRI images.

4.4 Performance of the Predictive Model

1. True Postive (TP)
2. True Negative (TN)
3. False Postive (FP)
4. False Negative (FN)

The performance metrics are evaluated to assess the relative efficacy of different machine learning (ML) methods in accuracy, precision, recall, and F1-score. These metrics provide insights into the models' ability to correctly classify tumor presence and distinguish between true positives, false positives, true negatives, and false negatives (Table 1; Figs. 5, 6 and 7).

Table 1 Comparison of three Pre-trained Keras model

Number of nodes	DL-SKG (kbits/sec)	EEPPM (kbits/sec)	DSDV (kbits/sec)
Test accuracy	0.941	0.825	0.645
Test accuracy	0.650	0.825	0.550
Overall accuracy	0.650000	0.825000	0.550000
Precision	0.556556	0.825000	0.550000
Recall	1.000250	0.825000	1.000250
F1-Score	0.693275	0.825000	0.677777
AUC	0.650000	0.825000	0.550000

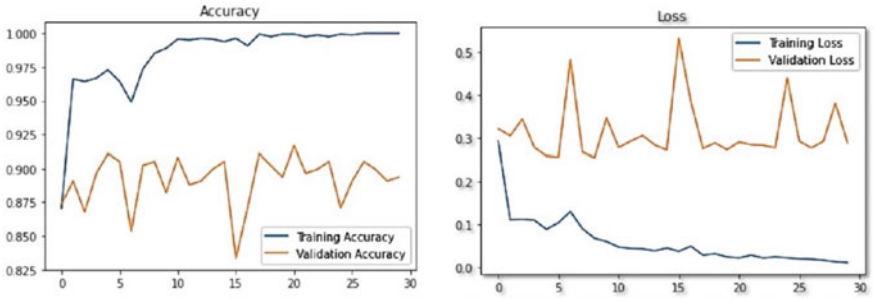


Fig. 5 Investigating the effectiveness of a CNN model for brain cancer detection and classification

$$\text{Accuracy} = (\text{TP} + \text{TP})/(\text{TP} + \text{FP} + \text{FN} + \text{TN}).$$

$$\text{Recall} = \text{TP}/(\text{TP} + \text{FN}).$$

$$\text{Precision} = \text{TP}/(\text{TP} + \text{FP}).$$

$$\text{F1 - score} = (2*\text{Precision}* \text{Recall})/(\text{Precision} + \text{Recall}).$$

5 Future Scope

Create an app-based user interface in healthcare facilities to evaluate tumor impact and administer correct therapy rapidly. Second, because the performance and complexity of ConvNets are dependent on the representation of the input data, we may try to anticipate the location and stage of cancer using volume-based 3D pictures.

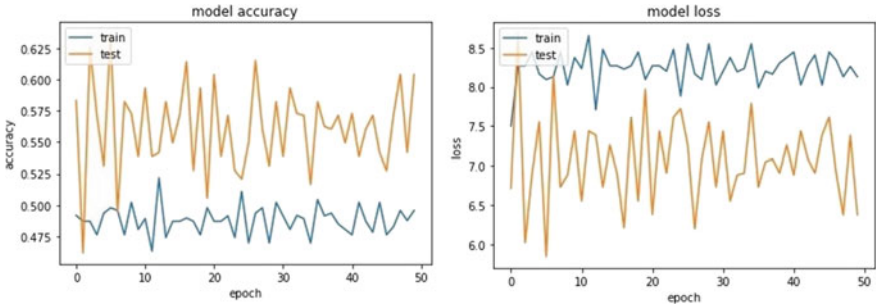


Fig. 6 Analyze the performance of the VGG16 model

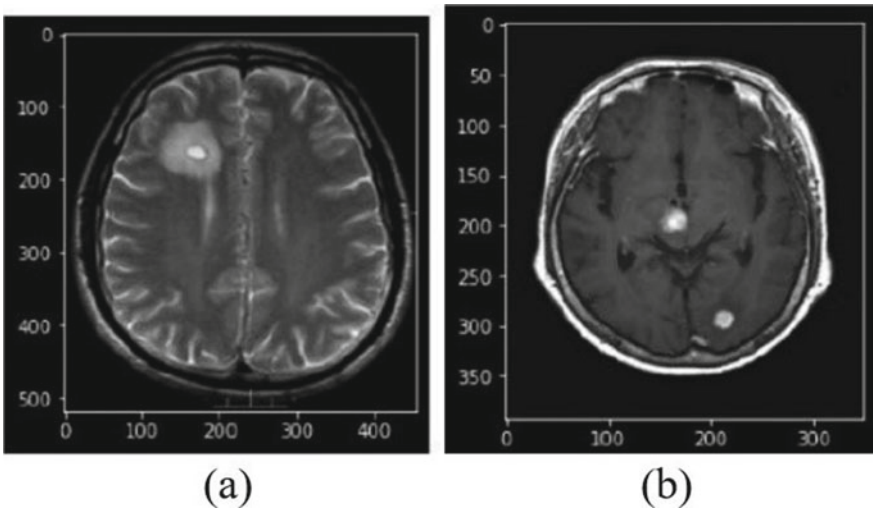


Fig. 7 **a** 99.93% it's a tumor; **b** 100% sure it's not a tumor

Training, planning, and computer-assisted surgery can benefit from creating patient-specific 3D anatomical models. And lastly, unsupervised transfer learning may gain popularity in the future.

6 Conclusion

A Keras model's train accuracy and validation accuracy without prior training are 97.5% and 90.0%, respectively. The accuracy of the validation result's best number was 91.09%. Although the training accuracy of the Keras model is > 90%. It has been discovered that when pre-trained models are not used, total accuracy suffers. In addition, our computation time was doubled from 20 to 40 min when we trained our

dataset with transfer learning instead of only using traditional methods. Thus, a pre-trained Keras model halves the time required for training and calculation. When the model is trained from scratch instead of pre-trained Keras, the risk of overfitting the dataset rises. Keras also has a simple user interface for augmenting data. ResNet50 has the greatest accuracy and F1-score compared to the other Keras models. In computer vision, ResNet serves as a complicated backbone model. Improvements in either precision or recall must come at the expense of the other. So, the F1-score is what is used. The proposed machine learning method can potentially enhance the precision and timeliness with which brain tumors are diagnosed, which may ultimately improve patient outcomes. The approach can help doctors determine the type of brain tumor and provide appropriate treatment options.

References

1. Sekhar A, Biswas S, Hazra R, Sunaniya AK, Mukherjee A, Yang L (2022) Brain tumor classification using fine-tuned GoogLeNet features and machine learning algorithms: IoMT enabled CAD system. IEEE J Biomed Health Inform 26(3):983–991. https://doi.org/10.1109/JBHI.2021.3100758
2. Kang J, Ullah Z, Gwak J (2021) MRI-based brain tumor classification using ensemble of deep features and machine learning classifiers. Sensors 21(6):2222. https://doi.org/10.3390/S21062222
3. Kaur D et al (2022) Computational intelligence and metaheuristic techniques for Brain tumor detection through IoMT-enabled MRI devices. Wirel Commun Mob Comput 2022. https://doi.org/10.1155/2022/1519198
4. Deepak S, Ameer PM (2021) Automated categorization of brain tumor from MRI using CNN features and SVM. J Ambient Intell Humaniz Comput 12(8):8357–8369. https://doi.org/10.1007/S12652-020-02568-W/METRICS
5. Ghosh A, Kole A (2021) A comparative study of enhanced machine learning algorithms for brain tumor detection and classification. https://doi.org/10.36227/TECHRXIV.16863136.V1
6. Serte S, Serener A, Al-Turjman F (2022) Deep learning in medical imaging: a brief review. Trans Emerg Telecommun Technol 33(10):e4080. https://doi.org/10.1002/ETT.4080
7. Ranjbarzadeh R, Caputo A, Tirkolaee EB, Jafarzadeh Ghoushchi S, Bendechache M (2023) Brain tumor segmentation of MRI images: a comprehensive review on the application of artificial intelligence tools. Comput Biol Med 152:106405. https://doi.org/10.1016/J.COMPBIOMED.2022.106405
8. Amin J, Sharif M, Haldorai A, Yasmin M, Nayak RS (2021) Brain tumor detection and classification using machine learning: a comprehensive survey. Complex Intell Syst 8(4):3161–3183. https://doi.org/10.1007/S40747-021-00563-Y
9. Khan MA et al (2019) Brain tumor detection and classification: a framework of marker-based watershed algorithm and multilevel priority features selection. Microsc Res Tech 82(6):909–922. https://doi.org/10.1002/JEMT.23238
10. Chaichana K, Quinones-Hinojosa A (2019) Comprehensive overview of modern surgical approaches to intrinsic brain tumors
11. Amin J, Sharif M, Raza M, Saba T, Sial R, Shad SA (2020) Brain tumor detection: a long short-term memory (LSTM)-based learning model. Neural Comput Appl 32(20):15965–15973. https://doi.org/10.1007/S00521-019-04650-7/METRICS
12. Amin J, Sharif M, Yasmin M, Fernandes SL (2020) A distinctive approach in brain tumor detection and classification using MRI. Pattern Recognit Lett 139:118–127. https://doi.org/10.1016/J.PATREC.2017.10.036

13. Saba T, Sameh Mohamed A, El-Affendi M, Amin J, Sharif M (2020) Brain tumor detection using fusion of hand crafted and deep learning features. Cogn Syst Res 59:221–230. https://doi.org/10.1016/J.COGSYS.2019.09.007
14. Prabukumar M, Agilandeeswari L, Ganesan K (2019) An intelligent lung cancer diagnosis system using cuckoo search optimization and support vector machine classifier. J Ambient Intell Humaniz Comput 10(1):267–293. https://doi.org/10.1007/S12652-017-0655-5/METRICS
15. Kumari R (2023) SVM classification an approach on detecting abnormality in brain MRI images 3:1686–1690. Accessed on 6 Jun 2023 [Online]. Available: www.ijera.com
16. Amin J, Sharif M, Raza M, Yasmin M (2018) Detection of brain tumor based on features fusion and machine learning. J Ambient Intell Humaniz Comput 1–17. https://doi.org/10.1007/S12652-018-1092-9/METRICS
17. Mohamed Shakeel P, Tobely TEE, Al-Feel H, Manogaran G, Baskar S (2019) Neural network based brain tumor detection using wireless infrared imaging sensor. IEEE Access 7:5577–5588. https://doi.org/10.1109/ACCESS.2018.2883957
18. Kumar S, Negi A, Singh JN, Verma H (2018) A deep learning for brain tumor mri images semantic segmentation using FCN. In: 2018 4th international conference on computing communication and automation, ICCCA 2018. https://doi.org/10.1109/CCAA.2018.8777675
19. Agravat RR, Raval MS (2019) Prediction of overall survival of brain tumor patients. In: IEEE region 10 annual international conference, Proceedings/TENCON, vol 2019-Oct, pp 31–35. https://doi.org/10.1109/TENCON.2019.8929497
20. Khare S, Gupta N, Srivastava V (2014) Optimization technique, curve fitting and machine learning used to detect brain tumor in MRI. In: Proceedings of ICCCS 2014–IEEE international conference on computer communication and systems, pp 254–259. https://doi.org/10.1109/ICCCS.2014.7068202
21. Mittal M, Goyal LM, Kaur S, Kaur I, Verma A, Jude Hemanth D (2019) Deep learning based enhanced tumor segmentation approach for MR brain images. Appl Soft Comput 78:346–354. https://doi.org/10.1016/J.ASOC.2019.02.036

A Dimensionality Reduction Method for the Fusion of NIR and Visible Image



Lokesh Gopinath and A. Ruhan Bevi

Abstract Fusing Near-infrared (NIR) and visible images aim to provide more detailed images for human inspection or other computer vision applications. Image fusion relies heavily on two components: measuring activity levels and assigning weights. The paper employs principal component analysis network (PCANET) to fuse Near-infrared (NIR) and visible data using an image pyramid. First, utilize a PCANET, an ultra-compact neural network for deep learning, to quantify action and give relative importance to infrared and visible images. PCANET's functional level assessment is superior in expressing key concepts such as NIR object processing and detailed visual representation. Second, the image pyramid divides the weights and source images into several scales and then applies the weighted-average fusing algorithm to the combined data. In the end, reconstruction produces the fused image. More than eighty test image pairings across two datasets confirmed the efficacy of the proposed approach. The experimental results reveal that the proposed strategy beats six benchmark approaches for subjective user evaluations and objective assessment metrics.

Keywords Image fusion · Near-infrared image · Visible image

1 Introduction

Near-infrared (NIR) sensors can still detect objects without light or smoke, which detect heat or thermal energy changes in a scene. However, NIR images need help with clarity, contrast, and invisibility of small features. Contrarily, in ideal lighting circumstances, visible images have superior spatial resolution and may more vividly

L. Gopinath · A. Ruhan Bevi (✉)
Department of Electronics and Communication Engineering, SRM Institute of Science and Technology, Kattankulathur, Chengalpattu 603203, Tamil Nadu, India
e-mail: ruhanb@srmist.edu.in

L. Gopinath
e-mail: lg0654@srmist.edu.in

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2023

S. Shakya et al. (eds.), *Fourth International Conference on Image Processing and Capsule Networks*, Lecture Notes in Networks and Systems 798,
https://doi.org/10.1007/978-981-99-7093-3_42

display the finer details of objects. Visible photos hardly capture relevant information for things in low light or behind the smoke. Therefore, NIR and visible image fusion aim to create an image with distinct NIR objects and an attractive background, allowing viewers to better assimilate all of the scene's information. Video surveillance, object detection, tracking, and remote sensing are just some military and civilian uses for combined infrared and visible images [1].

Image fusion has recently emerged as a hot topic in image processing. Many different approaches to the image became refined through time. These approaches divide roughly into four different categories: multi-scale transform [2], sparse representation [3], saliency [4], and deep learning methods. Deep learning addresses fusion problems because it is better at extracting features. CNN [5] initially implemented for multi-focus image fusion. Furthermore, [6] exercised a fusion algorithm of l_1-norm and the mean of weights using enhanced VGG-19, ZCA, and the l_1-norm [7] created a fusion technique that uses a pre-trained ResNet. There has been an uptick in the proposal of generative adversarial network-based approaches to fusing deep learning datasets. Ma J, Yu W, Liang P, et al [8] introduced a fusion model called FusionGAN that uses generative adversarial networks to produce a unified output by continually optimizing the generator. To preserve both the heat radiation in the NIR image and the consistency data in the visible image, the authors of [9] introduced a generative adversarial network (GAN) with a two-discriminator conditional (DDcGAN). To change the fusion problem into multi-distribution estimation, Ma J, Zhang H, Shao Z et al [10] created GANMcC. Although these fusion algorithms have accomplished good fusion outcome-based approaches, they need help to successfully extract and merge the complementary characteristics of NIR and visible images.

A novel method for merging Near-infrared (NIR) and visible images using an image pyramid and PCANET [11] is proposed, which employs a compact deep learning model to calculate activity levels and weights. The PCANET's activity level measurement of activity level is very representative since it emphasizes NIR focus and visual detail.

88 NIR and visible image pairs and six competing approaches confirmed the proposed algorithm's effectiveness, proving it can reach state-of-the-art visual quality and objective evaluation standards.

The remaining parts of the paper are structured as follows: Sect. 2 depicts the suggested method of fusing NIR and visible images. Section 3 presents the analysis and outcomes of the experiments. At last, this paper wraps up its last quarter.

2 The Proposed Method

This paper uses a unique PCANET and image pyramid-based NIR + visible fusion approach. Two critical aspects of image fusion are the quantification of activity and the allocation of relative importance. Because PCANET's representation ability is more robust due to its emphasis on NIR object processing and detailed visual representation, the human optical mechanism can simultaneously process information at

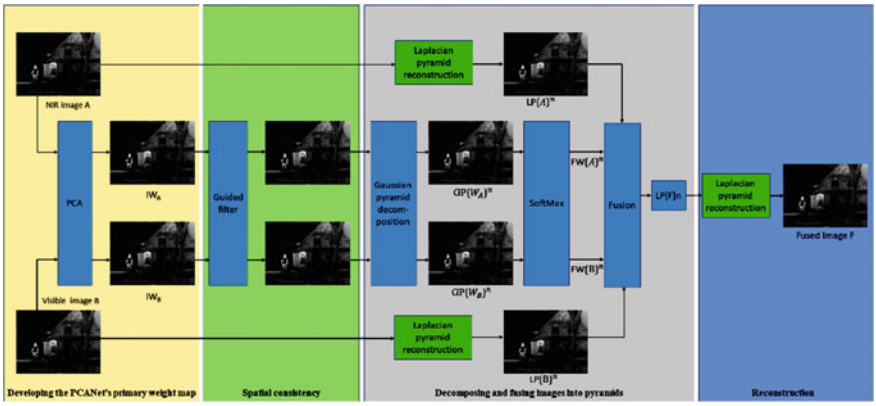


Fig. 1 Overview of the proposed model

several resolutions. Thus, a visual pyramid has used to break down and integrate images of different sizes to make the resulting image features more amenable to human sight.

Figure 1 shows the proposed model, and it involves four phases: Developing the PCANET's primary weight map, spatial consistency, decomposing and fusing images into pyramids, and reconstruction. First, run PCANET on the two source images to retrieve the default weight maps. Second, spatial consistency were used to enhance the characteristics of the preliminary weight maps. Decomposing images into pyramids and then fusing them is the third phase. The Laplacian pyramids perform a multi-scale transformation on the source images.

In contrast, the weight maps for each layer are obtained by first decomposing the initial weight maps into numerous dimensions using Gaussian pyramids, then applying the softmax function to each size. The weighted-average technique has been used to produce the combined image from all scales. Reconstructing the Laplacian pyramid yields the final fusion image.

2.1 Proposed PCANET Architecture

The fusion of Near-infrared (NIR) and visible images has approached this research as a two-class classification problem. PCANET generates a scalar between 0 and 1 for each pixel in the fused image, corresponding to the likelihood that it originated in a different one of the source images. Cascaded two-stage principal component analysis (PCA) [12], binary hashing, and block histograms comprise a standard PCANET, extracting sparse features from the images. The output light features from a network that uses binary hashing and block histograms have only two possible values, 0 and 1, and their size does not correspond to that of the original images. Cascaded two-stage PCA is the primary method in the fusion strategy because it provides a more accurate

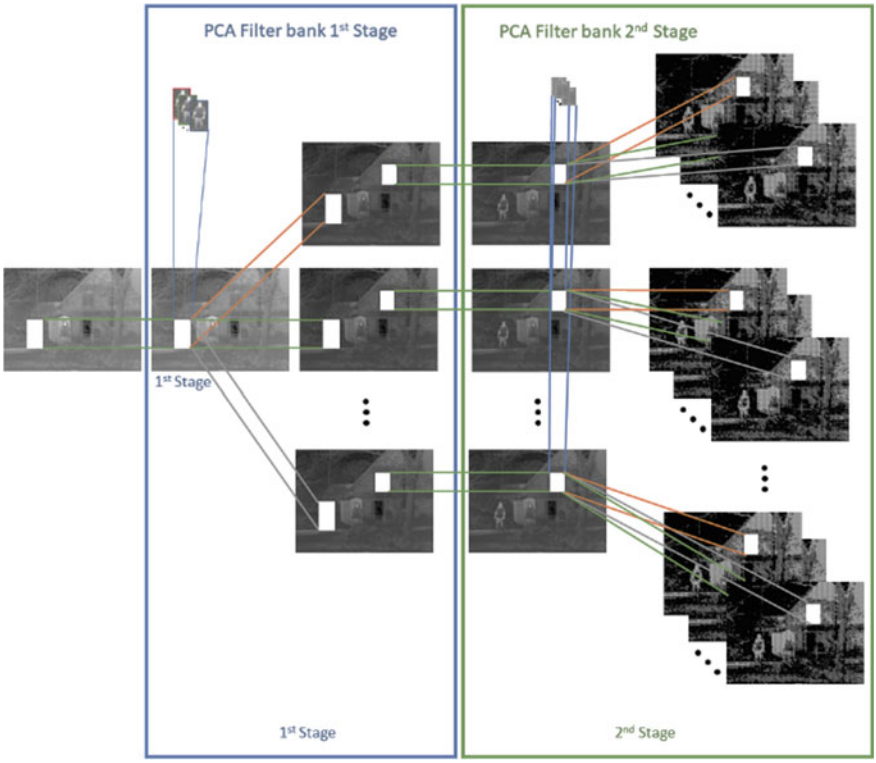


Fig. 2 Architecture of PCANET

probability of the exact location pixel in the NIR and the visible image and enables us to do the fusion work more quickly.

Figure 2 depicts the architecture of PCANET. The PCA filter is the core element of PCANET. The outline of the PCA filter training procedure is as follows. The input image has initially grouped with the PCA filter bank used in the first stage to build an array of feature maps in the PCANET. The supplied image has broken down into feature maps highlighting a specific item. In particular, the second-stage filters may be relied upon to glean more complex information. The cascaded two-stage PCA has been chosen because it is usually adequate to generate a significant effect after only two stages of PCA [11].

2.2 Procedure for the Proposed Method

PCANET training consists primarily of calculating the PCA filter. PCA is potentially an easy auto-encoder, as it optimizes for the least amount of reconstruction error [11].

Forty thousand images from Microsoft COCO [13] dataset are used for training exercises. The training image has scaled down to 256 × 256 grayscale because of the varying sizes of the images in the dataset of Microsoft COCO. For PCANET's training, the Filter size is $k_1 \times k_2$ in both stages, and each stage's training process is described below.

Stage 1:

Each training image has been pre-processed to make the convolution process go more smoothly. There are two stages to pre-processing.

(1) For the j-th training image I_j, every sliding $k_1 \times k_2$ plot was converted into an array of X_j, where

$$X_j = \left[x_{j,1}, x_{j,2}, x_{j,3}, \ldots, x_{j,\tilde{m}\tilde{n}}\right] \in \mathbb{R}^{k_1 k_2 \times \tilde{m}\tilde{n}},\ j = 1, 2, 3, \ldots$$
$$N, \tilde{m} = 256 - k_1 + 1, \tilde{n} = 256 - k_2 + 1$$

(2) $\overline{X}_j$ is obtained by eliminating the plot mean from each array in X_j. Whereas $\overline{X}_j = \left[\overline{x}_{j,1}, \overline{x}_{j,2}, \overline{x}_{j,3}, \ldots, \overline{x}_{j,\tilde{m}\tilde{n}}\right] \in \mathbb{R}^{k_1 k_2 \times \tilde{m}\tilde{n}}$. Then, applying the same technique to N training photos, as described before, to derive $X = \left[\overline{X}_1, \overline{X}_2, \ldots \overline{X}_N,\right] \in \mathbb{R}^{k_1 k_2 \times N\tilde{m}\tilde{n}}$. Next, X's covariance matrix C_1 is determined by solving the following:

$$C_1 = \frac{XX^T}{N\tilde{m}\tilde{n}} \tag{1}$$

This leads us to the next step, which is to determine the covariance matrix C_1' s eigenvalue $\wedge_1$ and eigenvector Q_1:

$$C_1 = Q_1 \wedge_1 Q_1^T \tag{2}$$

In this case, the matrix $\wedge_1$ has $k_1 k_2$ as diagonal eigenvalues. For each eigenvalue in the first row of Q_I, there is a matching eigenvector in the first column of Q_I. In particular, in the PCA filter, the eigenvalue indicates the importance of the related main component. Therefore, the PCA filters has chosen from the eigenvectors corresponding to the top S1 nearly significant eigenvalues. It allows us to write the l-th PCA filter as W_l^1, where $l = 1, 2, 3, \ldots, S_1$. The PCA filter bank in Stage1 has denoted as

$$W^1 = \left[W_1^1, W_2^1, \ldots, W_{S_1}^1\right] \tag{3}$$

The PCA filter bank detected significant image alterations. To obtain I_j pre-process the image by zero-padding the top and bottom edges by the size of $k_1 - 1$ and size $k_2 - 1$. Again it is pre-processed to acquire $\overline{I}_j$ and it is combined with l-th PCA filter

$$T_j^l = \overline{I}_j \times W_l^1,\ j = 1, 2, 3, \ldots, N, l = 1, 2, 3, \ldots, S_1 \tag{4}$$

T_j^l is a second-stage input sample, and $\times$ is the convolution process.

Stage 2:

First, like in the first stage, the input image T_j^l is pre-processed to obtain $\overline{Y}_j^l = \left[\overline{y}_{j,l,1}, \overline{y}_{j,l,2}, \overline{y}_{j,l,3}, \ldots, \overline{y}_{j,l,\tilde{m}\tilde{n}}\right] \in \mathbb{R}^{k_1 k_2 \times \tilde{m}\tilde{n}}$, and then the j input image is represented as $Y_j = \left[\overline{Y}_j^1, \overline{Y}_j^2, \overline{Y}_j^3 \ldots, \overline{Y}_j^{L_1}\right] \in \mathbb{R}^{k_1 k_2 \times S_1 \tilde{m}\tilde{n}}$. Repeating on every N source image, construct Y's covariance matrix C_2:

$$C_2 = \frac{YY^T}{NL_1\tilde{m}\tilde{n}} \tag{5}$$

$$C_1 = Q_2 \wedge_2 Q_2^T \tag{6}$$

where $\wedge_2$ and Q_2 stands for the second-stage eigen values and eigen vectors, respectively. Top eigenvectors S_2. As a result, the r-th PCA filter has designated as W_r^2, where $r = 1,2,3,\ldots, S_2$. Stage two PCA filter bank identification uses this formula:

$$W^2 = \left[W_1^2, W_2^2, , W_3^2 \ldots, W_{S_2}^2\right] \tag{7}$$

The filters used in the second stage can extract more complex characteristics than those used in stage one.

3 Detailed Fusion Scheme

3.1 Developing the PCANET's Primary Weight Map

Let A denote a Near-infrared image and B represent a visible image of the same size. In all steps, let's assume that the size of the PCA filters is $k_1 \times k_2$. The convolution's output must match the size of their inputs; therefore, the top and bottom edges of A and B are zero-padded by $k_1 - 1$ and $k_2 - 1$, respectively. Next, the input image H is pre-processed to get $\overline{H} \in \mathbb{R}^{K_1 K_2 \times \tilde{m}\tilde{n}}$ and the convolution filter is:

$$T_H^l = \overline{H} \times W_l^1, l = 1, 2, 3, \ldots, S_1 \tag{8}$$

First-stage PCA generates S_1 feature maps $T_H^1, T_H^2, \ldots, T_H^{S_1}$, via the filter bank W^1. In stage two, which is quite like stage one, T_s^l is zero-padded, and then pre-processing is done to get $\overline{U}_H^l \in \mathbb{R}^{K_1 K_2 \times \tilde{m}\tilde{n}}$. The next step consists of convolving $\overline{U}_H^l$ with the r-th principal component analyzer filter.

$$O_H^q = \overline{U}_H^l \times W_r^2, l = 1, 2, 3, \ldots, S_1, r = 1, 2, 3, \ldots, S_2, q = 1, 2, 3, \ldots, S_1 S_2. \tag{9}$$

Stage two PCA produces S_1 S_2 feature maps O_H^1, O_H^2, …, $O_H^{S_1 S_2}$. Next, the starting weight maps of NIR image A and visible image B are IW_A and IW_B respectively:

$$IW_A(u, v) = O_A^1(u, v) + O_A^2(u, v) + \ldots + O_A^{S_1 S_2}(u, v) \tag{10}$$

$$IW_B(u, v) = O_B^1(u, v) + O_B^2(u, v) + \ldots + O_B^{S_1 S_2}(u, v) \tag{11}$$

where u and v are image coordinates or the positions of individual pixels. The dimensions of IW_A and IW_B are identical to those of the source images.

3.2 Spatial Consistency

If two neighboring pixel have about the same brightness or hue, their weights should be pretty close [14]. In most cases, the noise in the original weight maps will manifest as artifacts in the final combined image. Processing the initial weight maps further is necessary to enhance fusion performance. A guided filter [15] is employed to enhance the caliber of the preliminary weight maps. The input image E have used as a reference for setting the first filtering weight map's absolute value:

$$IW_A = \text{Guided Filter}(A, |IW_A|) \tag{12}$$

$$IW_B = \text{Guided Filter}(B, |IW_B|) \tag{13}$$

where letters A and B stand for directional images. The experiments has performed with a guided filter's radius 50 and regularization 0.1.

3.3 Decomposing and Fusing Images into Pyramids

To generate $GP\{W_A\}^n$ and $GP\{W_B\}^n$ applying n-layer Gaussian pyramid decomposition [16] to IW_A and IW_B, respectively using Eq. (14).

$$GP_h(i, j) = \sum_{m=-2}^{2} \sum_{n=-2}^{2} GP_{h-1}(2i + m, 2j + n)s(m, n) \tag{14}$$

where (i, j) are the image coordinates and $i \in [0, R_h - 1]$, $j \in [0, C_h - 1]$ and $h \in [1, N]$ respectively. R_h and C_h are the row and column counts of Gaussian pyramid, N is the decomposition's level count. Furthermore, $s(m, n)$ is also a 2D 5 $\times$ 5 window function.

The value $[\log_2 \min(Hig, Wid)]$, whereas $Hig \times Wid$ is the physical height and width of the images. Next, the outputs of $GP\ \{W_A\}^n$ and $GP\ \{W_B\}^n$ are input into a 2-way softmax layer, where they are converted into probability values for two classes:

$$FW\{A(u,v)\}^n = \frac{e^{GP\{W_A(u,v)\}^n}}{e^{GP\{W_A(u,v)\}^n} + e^{GP\{W_B(u,v)\}^n}} \tag{15}$$

$$FW\{B(u,v)\}^n = \frac{e^{GP\{W_B(u,v)\}^n}}{e^{GP\{W_A(u,v)\}^n} + e^{GP\{W_B(u,v)\}^n}} \tag{16}$$

The values of FWAn and FWBn, which run from 0 to 1, show the possibility that both A and B have results at the same pixel location. After these steps are completed, the network learns image attributes and computes the weight of each pixel on its own, saving the user time and effort to build the fusion rules manually.

Further, $LP\ \{A\}^n$ and $LP\ \{B\}^n$ is obtained using Eq. (17) for A and B, respectively, using n-layer Laplacian pyramid decomposition [16].

$$\left\{ \begin{array}{l} LP_h = GP_h - G^*_{h+1}, h \in [0, N-1] \\ LP_N = GP_N, h = N \end{array} \right\} \tag{17}$$

The Laplacian pyramid and the Gaussian pyramid have the same number of decomposition levels. It's worth noting that $FW\ \{A\}^n$ and $FW\ \{B\}^n$ correspond exactly to $LP\ \{A\}^n$ and $LP\ \{B\}^n$ in terms of size. The fused image has calculated using the weighted-mean rule.

$$\begin{aligned} L\{F(u,v)\}^n =& FW\{A(u,v)\}^n \times LP\{A(u,v)\}^n \\ &+ FW\{B(u,v)\}^n \times LP\{B(u,v)\}^n \end{aligned} \tag{18}$$

3.4 Reconstruction

The Laplacian pyramid image F have constructed using Eq. (19)

$$\left\{ \begin{array}{l} GP_N = LP_N \\ GP_h = LP_h + \text{EXPAND}\ (GP_{h+1}), h \in [0, N-1] \\ I = GP_0 \end{array} \right\} \tag{19}$$

The key operations of NIR and the visible image fusion approach are outlined in the following algorithm.

Algorithm for fusing Near-infrared and visible image

Training

1. Initialize PCANET;
2. Compute W^1 using Eqs. (1)–(4).
3. Compute W^2 using Eqs. (5)–(7).

Testing

Part 1: **Developing the PCA's primary weight map**

1. Provide NIR and visible images to get the primary weight maps using Eqs. (8)–(11).

Part 2: **Spatial consistency**

2. Execute the guided filtering of raw values of and using the equation using Eqs. (12) and (13).

Part 3: **Decomposing and fusing images into pyramids**

3. Decompose a Gaussian pyramid into n-layers to IWA and IWB, yielding $GP\ \{W_A\}^n$ and $GP\ \{W_B\}^n$ using Eqs. (14).
4. Obtain $FW\ \{A\}^n$ and $FW\ \{B\}^n$ using the softmax operation in each layer by Eqs. (15) and (16).
5. Decompose A and B into n layers using the Laplacian pyramid to get $LP\ \{A\}^n$ and $LP\ \{B\}^n$ using Eq. (17).
6. Calculate the result $L\ \{F\}^n$ using the weighted-average method for each layer using Eq. (18).

Part 4: **Reconstruction**

7. To generate the fused image F, reconstruct the Laplacian pyramid using Eq. (19).

4 Experiments

4.1 Datasets

The TNO dataset [17] and the real-time dataset are selected to conduct tests on to test the proposed method's efficacy thoroughly. TNO dataset contains hundreds of already aligned pairings of NIR and visible images. Most of the photographs in the dataset depict military-related scenes like bases, airplanes, and jet aircraft. As a test, thirty image pairings were extracted from the TNO dataset and evaluated on a laptop with 16 GB of RAM, an Intel Core i7 processor from the 11th generation, and Python software.

Figure 3 displays three sets of testing photos from TNO dataset, with NIR images on top and the visible images on the bottom. Unlike TNO dataset, real-time dataset has 80 pairs of NIR and visible images. In this, 30 pairs of images chose to test. Figure 4 displays three sets of testing images of a real-time dataset, with the NIR images on top and the visible images on the bottom.



Fig. 3 Three test image pairs from TNO dataset are illustrated

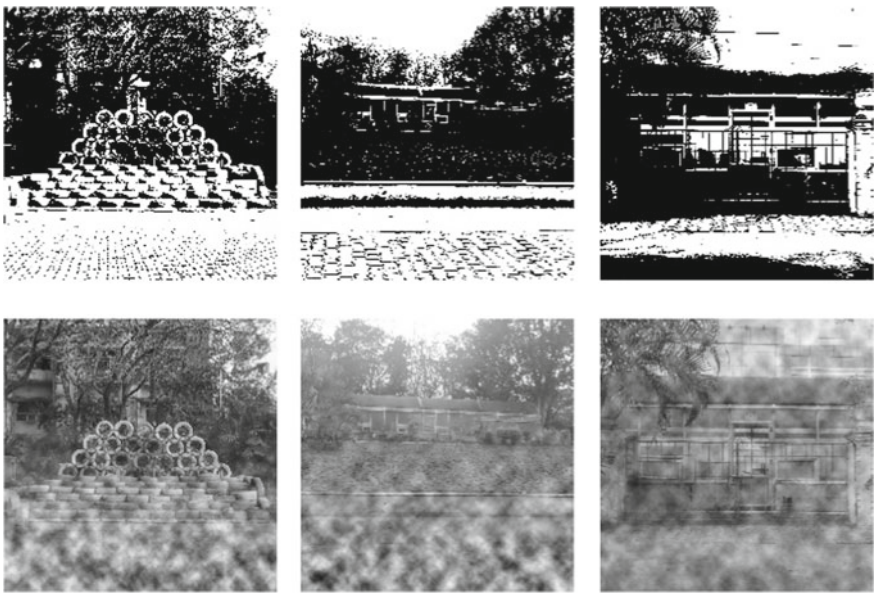


Fig. 4 Three test image pairs from the real-time dataset are illustrated

4.2 Image Fusion Quality Metrics

The proposed algorithm's fusion impact has been evaluated using six objective assessment measures. They are.

Structural similarity index measure SSIM [18]: SSIM is a fusion index that primarily determines the degree of structural similarity between the resulting fusion and the original image.

FMI_w FMI_{dct} and FMI_{pixel} [19]: Feature Mutual Information (FMI) in wavelet, discrete cosine transform, and pixel, respectively.

Yang's metric Q_Y [20]: Q_Y Measures the retention of structural information during fusion.

Enhanced evaluation of fusion effects N_{abf} [21] gives an index that reflects the amount of noise or artifacts introduced into the fused image during the fusion process.

4.3 Impact of Filters

The PCA is a condensed network in which the only variables that can be altered are the filter size, the number of filters in the second stage (S_2), and the number of filters in the first stage (S_1). Matching filter sizes guaranteed phase compatibility. Forty-four image pairs from the TNO dataset were used to evaluate parameter values. The best values of 6 fusion measures are averaged to determine the fusion performance, with the superior values highlighted in red.

Filter count affects fusion efficiency. PCA filter size remained 3×3, whereas S_1 and S_2 filter counts were adjusted from 3 to 8, as indicated in Table 1. There is a correlation between the number of S_1 and S_2 components used in the principal components analysis (PCA). The more filters there are, the more information the model can glean from the data. The effects of varying L_1 and L_2 counts on fusion efficiency are displayed in Table 1. The model finds four optimal values is $S_1 = S_2 = 8$. The model is more significant, so running the model will take longer, and SSIM may be less accurate. Setting $S_1 = S_2 = 8$ helps to make the model as straightforward as feasible.

The experiment explores how filter size affects fusion efficiency. Table 2 shows the results of applying PCA filters with varying granularities 3×3, 5×5, 7×7, 9×9, and 11×11 to the data with, $S_1 = S_2 = 8$. Changing the filter size in PCA alters the receptive field and the features extracted. With a bigger filter size, the model can glean more information. The effect of various filter sizes on fusion performance has shown in Table 2. An 11×11 PCA filter yields the optimum fusion performance. Thus, $S_1 = S_2 = 8$, and a 11×11 PCA filter is implemented.

Table 1 The impact of the filter. S_1 and S_2 represents 1st stage and 2nd stage filter

S_1	S_2	SSIM	FMI_w	FMI_{dct}	FMI_{pixel}	Q_Y	N_{abf}
3	3	0.7495	0.4167	0.3991	0.9079	0.6868	0.0000
3	4	0.7495	0.4167	0.3991	0.9079	0.6874	0.0000
4	4	0.7494	0.4168	0.3991	0.9080	0.6878	0.0000
4	5	0.7494	0.4169	0.3991	0.9080	0.6886	0.0000
5	5	0.7494	0.4169	0.3991	0.9080	0.6883	0.0000
5	6	0.7494	0.4169	0.3992	0.9080	0.6885	0.0000
6	6	0.7494	0.4169	0.3992	0.9080	0.6887	0.0000
6	7	0.7494	0.4170	0.3993	0.9080	0.6888	0.0000
7	7	0.7494	0.4170	0.3994	0.9080	0.6894	0.0000
7	8	0.7494	0.4171	0.3994	0.9081	0.6897	0.0000
8	8	0.7493	0.4174	0.3995	0.9081	0.6900	0.0000

Table 2 The impact of filter size

Size	SSIM	FMI_w	FMI_{dct}	FMI_{pixel}	Q_Y	N_{abf}
3×3	0.7493	0.4174	0.3995	0.9081	0.6900	0.0000
5×5	0.7475	0.4194	0.3981	0.9108	0.7151	0.0000
7×7	0.7432	0.4243	0.3931	0.9133	0.7505	0.0000
9×9	0.7374	0.4306	0.3830	0.9150	0.7862	0.0000
11×11	0.7299	0.4405	0.3720	0.9162	0.8228	0.0000

5 Results and Discussion

5.1 Evaluation Against Rival Algorithms Using TNO Dataset

TNO dataset was utilized to test and validate the proposed technique. An 11×11 pixel filter size has used across both stages. The number of layers needed for the image-pyramid decomposition was $n, n = [\log_2 \min(Hig, Wid)]$, whereas $Hig \times Wid$ is the physical height and width of the images used as input. The guided filter has a regularization value of 0.1 and a filter radius of 50. Qualitative and quantitative measures were used to evaluate the suggested technique's fusion performance.

Fusion outcomes of the Kaptein_1123 source image, which depicts a human being positioned at the door, are displayed in Fig. 5. Details of houses, trees, and grass may all be seen in the viewable image. However, the NIR is very sensitive to thermal radiation and records this data.

A human being in the FusionGAN and DDcGAN methods produces hazy results. The combined images are unsettling and unrealistic. The FusionDN (Fig. 5b) can obtain excellent fusion performance. Furthermore, the sky and ground cannot be captured in all their glory by the GANMcC (an orange and green rectangle in Fig. 5d).

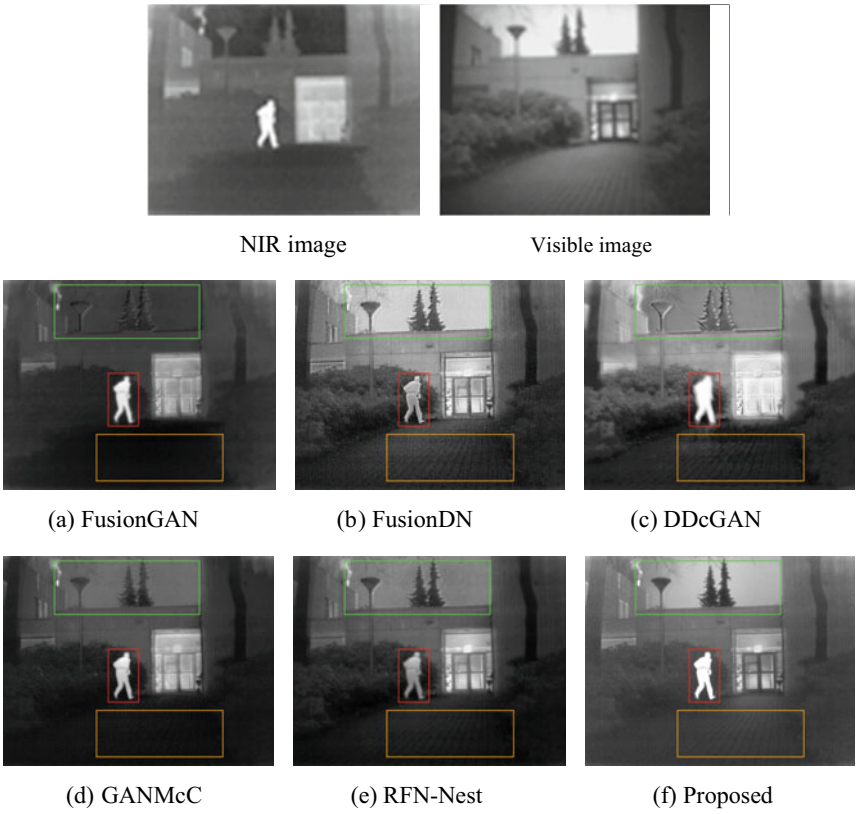


Fig. 5 Kaptein_1123 fusion output

while RFN-Nest is dim (Fig. 5e). Sky quality (a green rectangle in Fig. 5f), thermal radiation object brightness (a red rectangle in Fig. 5f), and ground texture clarity (a blue rectangle in Fig. 5f) are all improved over competing approaches by using the proposed method.

Table 3 displays the mean values for six objective assessment metrics applied to TNO dataset, with the highest values highlighted in red. It shows that, except FMI_{dct}, the proposed algorithm achieved the highest scores possible, demonstrating superior fusion performance.

5.2 Comparing to Other Competitive Algorithms Using the Real-Time Dataset

The Basler acA2040-25gmNIR captured real-time NIR images, while the HIKVISION 6MP IP Camera captured visible images. Due to the image's various sizes, it

Table 3 Results from several techniques averaged from the TNO dataset

Size	SSIM	FMI_w	FMI_{dct}	FMI_{pixel}	Q_Y	N_{abf}
FusionGAN	0.6430	0.3754	0.3565	0.8889	0.5263	0.0131
FusionDN	0.6230	0.3597	0.3097	0.8842	0.6856	0.1356
DDcGAN	0.5820	0.4114	0.3863	0.8760	0.6390	0.1016
GANMcC	0.6824	0.3820	0.3512	0.8980	0.5976	0.0099
RFN-Nest	0.6820	0.2976	0.2897	0.9032	0.6263	0.0114
Proposed	0.7299	0.4406	0.3719	0.9162	0.8238	0.0002

is down-scaled into 256×256 grayscale. To verify and evaluate real-time datasets in both stages, an 11×11 filter size is used. The number of layers needed for the image-pyramid decomposition was n, $n = [\log_2 \min(Hig, Wid)]$, whereas $Hig \times Wid$ is the physical height and width of the images used as input. The guided filter has a regularization value of 0.1 and a filter radius of 50. To evaluate the fusion performance of the suggested technique, qualitative and quantitative measures were used.

Fusion outcomes of the Architect_1111 source image, which depicts an Architecture made of vehicle tires in front of the trees, are displayed in Fig. 6. Details of buildings, trees, and grass may all be seen in the viewable image. However, the NIR is very sensitive to thermal radiation and records this data.

A crate in the FusionGAN and DDcGAN methods produces hazy results. The combined images are unsettling and unrealistic. Excellent fusion performance can be achieved with the FusionDN (Fig. 6b). Furthermore, the grass and ground cannot be captured in all their glory by the GANMcC (green and blue rectangles in Fig. 6d). While RFN-Nest is dim. grass (a green rectangle in Fig. 6f), thermal radiation object brightness (a red rectangle in Fig. 6f), and ground texture clarity (a blue rectangle in Fig. 6f) are all improved over competing approaches by using the proposed method.

Table 4 displays the mean values for six objective assessment metrics applied to a real-time dataset, with the highest values highlighted in red. It shows that, except FMI_{dct}, the proposed algorithm achieved the highest scores possible, demonstrating superior fusion performance.

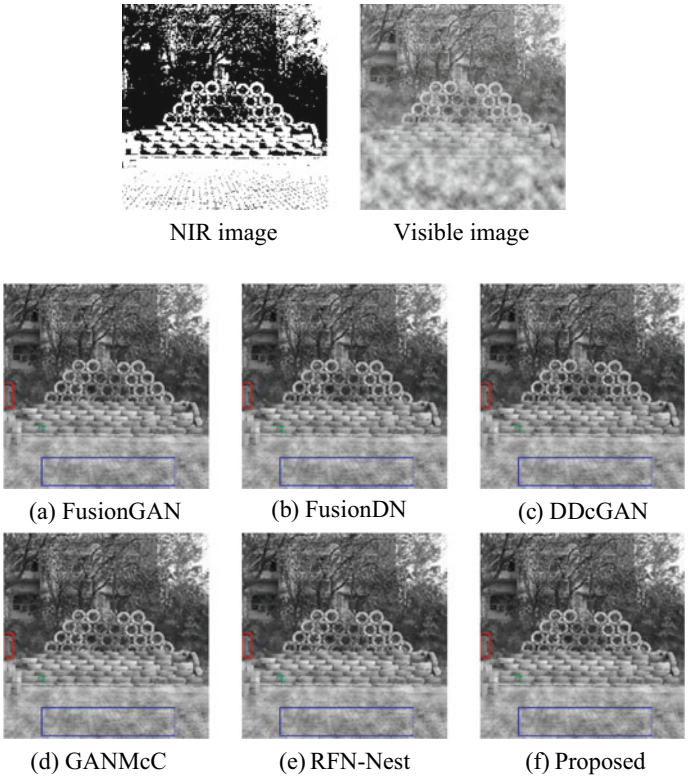


Fig. 6 Architect_1111 fusion output

Table 4 Results from several techniques averaged from real-time datasets

Size	SSIM	FMI_w	FMI_{dct}	FMI_{pixel}	Q_Y	N_{abf}
FusionGAN	0.6430	0.3169	0.3565	0.8889	0.5263	0.0131
FusionDN	0.6230	0.3665	0.3097	0.8842	0.6856	0.1356
DDcGAN	0.5820	0.3499	0.3863	0.8760	0.6390	0.1016
GANMcC	0.6824	0.3693	0.3512	0.8980	0.5976	0.0099
RFN-nest	0.6820	0.2737	0.2897	0.9032	0.6263	0.0114
Proposed	0.7299	0.3994	0.3719	0.9162	0.8238	0.0002

6 Conclusion

This paper uses a PCANET and image pyramid-based fusion technique for NIR and visible images. PCANET measures weight activity levels and image pyramids for scaling and decomposing images. Measuring PCANET activity is particularly useful for capturing information on detecting NIR targets and describing visual details.

Experimental findings showed that this technique achieved state-of-the-art performance in terms of subjective evaluation metrics (such as perceived quality) and objective evaluation metrics (such as speed and accuracy). The PCANET output is used as image features, skipping over the first stage's potentially relevant information. Future research will examine the characteristics of fusion tasks that involve merging data from various phases.

Acknowledgements This research work is funded by SRM Institute of Science and Technology, Kattankulathur, under Selective Excellence Research Initiative.

References

1. Liu S, Lyu C, Gong H (2021) Vehicle video surveillance system based on image fusion and parallel computing. Int J Circuit Theory Appl 49:1532–1547. https://doi.org/10.1002/cta.2907
2. Chen J, Li X, Luo L et al (2020) Infrared and visible image fusion based on target-enhanced multiscale transform decomposition. Inf Sci (N Y) 508:64–78. https://doi.org/10.1016/j.ins.2019.08.066
3. Liu Y, Chen X, Ward RK, Wang J (2016) Image fusion with convolutional sparse representation. IEEE Signal Process Lett 23:1882–1886. https://doi.org/10.1109/LSP.2016.2618776
4. Ma J, Chen C, Li C, Huang J (2016) Infrared and visible image fusion via gradient transfer and total variation minimization. Inf Fusion 31:100–109. https://doi.org/10.1016/j.inffus.2016.02.001
5. Liu Y, Chen X, Peng H, Wang Z (2017) Multi-focus image fusion with a deep convolutional neural network. Inf Fusion 36:191–207. https://doi.org/10.1016/j.inffus.2016.12.001
6. Li H, Wu X-J, Kittler J (2018) Infrared and visible image fusion using a deep learning framework. https://doi.org/10.1109/ICPR.2018.8546006
7. Li H, Wu XJ, Durrani TS (2019) Infrared and visible image fusion with ResNet and zero-phase component analysis. Infrared Phys Technol 102. https://doi.org/10.1016/j.infrared.2019.103039
8. Ma J, Yu W, Liang P et al (2019) FusionGAN: a generative adversarial network for infrared and visible image fusion. Inf Fusion 48:11–26. https://doi.org/10.1016/j.inffus.2018.09.004
9. Ma J, Xu H, Jiang J et al (2020) DDcGAN: a dual-discriminator conditional generative adversarial network for multi-resolution image fusion. IEEE Trans Image Process 29:4980–4995. https://doi.org/10.1109/TIP.2020.2977573
10. Ma J, Zhang H, Shao Z et al (2021) GANMcC: a generative adversarial network with multiclassification constraints for infrared and visible image fusion. IEEE Trans Instrum Meas 70. https://doi.org/10.1109/TIM.2020.3038013
11. Chan TH, Jia K, Gao S et al (2015) PCANet: a simple deep learning baseline for image classification? IEEE Trans Image Process 24:5017–5032. https://doi.org/10.1109/TIP.2015.2475625
12. Greenacre M, Groenen PJF, Hastie T et al (2023) Publisher correction: principal component analysis. Nature Rev Methods Primers 3:22. https://doi.org/10.1038/s43586-023-00209-y
13. Lin T-Y, Maire M, Belongie S et al (2014) LNCS 8693—Microsoft COCO: common objects in context. In: computer vision—ECCV 2014 In: 13th European conference, Zurich, Switzerland, September 6–12, 2014, proceedings, Part V 13, pp 740–750
14. Li S, Kang X, Hu J (2013) Image fusion with guided filtering. IEEE Trans Image Process 22:2864–2875. https://doi.org/10.1109/TIP.2013.2244222
15. He K, Sun J, Tang X (2013) Guided image filtering. IEEE Trans Pattern Anal Mach Intell 35:1397–1409. https://doi.org/10.1109/TPAMI.2012.213

16. Mertens T, Kautz J, Van Reeth F (2007) Exposure fusion. In: Proceedings—15th Pacific conference on computer graphics and applications, pp 382–390
17. Toet A (2017) The TNO multiband image data collection. Data Brief 15:249–251. https://doi.org/10.1016/j.dib.2017.09.038
18. Wang Z, Bovik AC, Sheikh HR, Simoncelli EP (2004) Image quality assessment: from error visibility to structural similarity. IEEE Trans Image Process 13:600–612. https://doi.org/10.1109/TIP.2003.819861
19. Haghighat M, Razian MA (2014) Fast-FMI: non-reference image fusion metric. In: IEEE 8th international conference on application of information and communication technologies (AICT), pp 1–3
20. Li S, Hong R, Wu X (2008) International conference on audio, language and image processing 2008.07.07-09 Shanghai. A novel similarity based quality metric for image fusion, pp 167–172
21. Petrovi V, Xydeas C (2005) Objective image fusion performance characterisation. In: Tenth IEEE international conference on computer vision (ICCV'05) Volume 1, Beijing, China, vol 2, pp 1866–1871. https://doi.org/10.1109/ICCV.2005.175

Hand Gesture-Based AI System for Accessing Windows Applications



Lakshmi Harika Palivela, V. Premanand, and Afruza Begum

Abstract Actions speak more than words. In the context of the above statement, the importance of gestures and using them to control a system has become popular. The hand gesture recognition system for opening applications in Windows is a computer vision application that uses a webcam and machine learning models to recognize hand gestures and open specific applications. This system allows users to control their computers using simple hand gestures instead of relying on traditional input devices such as keyboards or mouse. The system uses the Tensor Flow and Media-Pipe libraries for training and detecting hand gestures, respectively. The trained machine learning model recognizes the gestures, and specific applications are opened based on the recognized gesture. The system has potential applications in accessibility, gaming, and virtual reality. In this work, we will discuss potential applications of the hand gesture recognition system for opening applications in Windows, and the second proposed system, which may also act as the control system of PowerPoint presentations, can be made easy and convenient using these gestures-based control. Thus, the proposed model improves the interaction between the human and the computer.

Keywords Gesture-based control · Media-Pipe · Human–computer interaction · Artificial intelligence · Hand-sign

L. H. Palivela (✉) · V. Premanand · A. Begum
School of Computer Science Engineering, Vellore Institute of Technology, Chennai, India
e-mail: lakshmi.harika@vit.ac.in

V. Premanand
e-mail: premanand.v@vit.ac.in

A. Begum
e-mail: afruza.begum@vit.ac.in

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2023

S. Shakya et al. (eds.), *Fourth International Conference on Image Processing and Capsule Networks*, Lecture Notes in Networks and Systems 798,
https://doi.org/10.1007/978-981-99-7093-3_43

1 Introduction

The hand gesture recognition system for opening applications in Windows is an innovative technology that has been gaining popularity in recent years. It is a type of computer vision technology that enables users to interact with their computers using hand gestures. This technology allows users to open computer applications by making hand gestures in front of their computer's camera. This system is based on gesture recognition, which involves identifying and interpreting the human body's movements.

The hand gesture recognition system uses machine learning algorithms to analyze the images captured by the camera, recognize the hand gestures made by the user, and trigger specific actions in response to those gestures. The system can detect and recognize different hand gestures, such as thumbs up, thumbs down, okay sign, and others, and map those gestures to specific actions, such as opening an application or executing a command. The system has many potential applications, including human–computer interaction, gaming, and virtual reality. The gesture-based control can improve the user experience for everyone by simplifying interactions and reducing the cognitive load required to use technology. In addition, it can be used in situations where users need to interact with their computers without touching them, such as in environments where hygiene is paramount. Using hand and body gestures, users can manipulate virtual objects and navigate digital spaces more naturally and intuitively.

2 Literature Review

Ranjan et al. propose a method for recognizing hand gestures using convolutional neural networks (CNNs). The authors developed a dataset of 2600 grayscale images of hand gestures captured from five individuals for training and testing the CNN. They used the LeNet-5 architecture and achieved a recognition rate of 99.6% on their dataset. They believe their approach can be used for gesture-based human–machine interactions, particularly in areas such as sign language recognition [1]. In Sahoo et al. [2], a real-time hand gesture recognition system is developed based on a dataset of 1600 images of ten hand gestures captured from 20 individuals for training and testing their CNN. They used a two-stage CNN architecture and achieved a recognition rate of 99.88% on their dataset. The authors believe their approach can be used in human–robot interaction, sign language recognition, and virtual reality applications.

Zhan et al. propose a hand gesture recognition system using CNNs. The authors developed a dataset of 3000 images of five hand gestures, captured from 10 individuals, for training and testing their CNN. They used a four-layer CNN architecture and achieved a recognition rate of 98.7% on their dataset. Their approach can be used in areas such as human–robot interaction, virtual reality, and gaming [3]. In a real-time hand gesture recognition system is done using CNNs and data augmentation

techniques. The authors developed a dataset of 1500 images of five hand gestures, captured from 10 individuals, for training and testing their CNN. They used a four-layer CNN architecture and achieved a recognition rate of 95% on their dataset. The authors believe their approach can be used in gaming, virtual reality, and robotics applications [4].

Agaian et al. propose a hand gesture recognition system using CNNs. The authors developed a dataset of 4800 images of 12 hand gestures, captured from 20 individuals, for training and testing their CNN. They used a five-layer CNN architecture and achieved a recognition rate of 95.6% on their dataset. Their approach can be used in applications such as sign language recognition and human–computer interaction [5]. Hand gesture recognition using Media-Pipe has gained significant attention due to its ease of use and accuracy. Media-Pipe is an open-source, cross-platform library developed by Google, which provides a customizable pipeline for processing and detecting hand landmarks in real-time. One of the significant advantages of using Media-Pipe is its ability to detect hand landmarks even in different lighting and background conditions. In this work proposed a real-time hand gesture recognition system using Media-Pipe, which was based on a convolutional neural network (CNN) and achieved recognition accuracy of 98.9%. The proposed system used a dataset of 10,000 images for training and testing, and the results showed the effectiveness of the proposed system in real-time applications [6, 7].

Wameed et al. [8] used Media-Pipe to detect and recognize hand gestures for controlling a robot. The authors developed a system that allowed users to control a robot using hand gestures detected by Media-Pipe. The proposed system used an artificial neural network (ANN) for gesture recognition and achieved an accuracy of 97%. The results showed the feasibility of using hand gestures for robot control, and the system could recognize gestures in real-time. In the work [9], the authors proposed a sign language recognition system using a combination of Media-Pipe and CNN. The system used Media-Pipe for hand landmark detection and extracted hand features to train a CNN for gesture recognition. The proposed system was tested on a dataset of 10,000 images and achieved an accuracy of 95.7%, outperforming other state-of-the-art methods. The results showed the effectiveness of using Media-Pipe for hand landmark detection and the ability of CNN to recognize sign language gestures accurately.

Zhu et al. proposed a system for detecting hand gestures using a single camera and Media-Pipe. The authors used Media-Pipe to see hand landmarks and extract hand features to train a CNN for gesture recognition. The proposed system was tested on a dataset of 10,000 images and achieved an accuracy of 98.7%. The results showed the feasibility of using a single camera and Media-Pipe for real-time hand gesture recognition applications [10]. The authors proposed a real-time hand gesture recognition system using Media-Pipe and CNN. The system was designed to recognize ten hand gestures and was trained on a dataset of 8000 images. The proposed system achieved an accuracy of 96.8% in recognizing hand gestures in real-time. The results showed the effectiveness of using Media-Pipe for hand landmark detection and the potential of CNN for real-time hand gesture recognition applications [11].

Haria et al. propose a hand gesture recognition system for controlling application interfaces. The authors developed a dataset of 50 hand gesture samples for five commands and used a Hidden Markov Model (HMM) for gesture recognition. The system achieved an accuracy of 90.4% in recognizing hand gestures [12]. The authors believe their approach can be used in smart homes and interactive television applications. Khan et al. propose a hand gesture recognition system for opening applications in a Windows environment. The authors developed a dataset of 500 images of ten hand gestures captured from five individuals for training and testing their system [13]. They used a combination of region-based and template-matching techniques for gesture recognition and achieved an accuracy of 94.5% on their dataset. The authors believe their approach can be used in virtual reality and gaming applications.

Dehankar et al. [14] propose a gesture-based interface for controlling applications on mobile devices. The authors developed a dataset of 50 hand gesture samples for five different commands and used a template-matching technique for gesture recognition. The system achieved an accuracy of 80% in recognizing hand gestures. The authors believe their approach can be used in mobile gaming and virtual reality applications. Saraswati et al. [15] propose a gesture-based system for opening applications on a desktop computer. The authors developed a dataset of 250 images of ten hand gestures captured from five individuals for training and testing their system. They used a Haar classifier for gesture recognition and achieved an accuracy of 95% on their dataset. The authors believe their approach can be used in smart homes and interactive television applications.

In [16] propose a hand gesture recognition system for opening applications in a virtual reality environment. The authors developed a dataset of 500 images of ten hand gestures captured from five individuals for training and testing their system. They used a combination of region-based and template-matching techniques for gesture recognition and achieved an accuracy of 95% on their dataset by using SVM. The authors believe their approach can be used in applications such as virtual reality and gaming propose a hand gesture recognition system using a convolutional neural network (CNN) to open computer applications [17]. The authors developed a dataset of 15,000 images of five hand gestures, captured from 20 individuals, for training and testing their CNN. They used a three-layer CNN architecture and achieved a recognition accuracy of 99.2% on their dataset. Their approach can be used for natural and intuitive human–computer interaction.

In Adithya [18], a hand gesture recognition system is developed using a deep convolutional neural network (DCNN) to control computer applications. The authors developed a dataset of 1500 images of five hand gestures captured from three individuals for training and testing their DCNN. They used a five-layer DCNN architecture and achieved recognition accuracy of 96.7% on their dataset. The authors believe their approach can be used to improve user experience in human–computer interaction. Niranjani et al. [19] provides a hand gesture recognition system using a deep convolutional neural network (DCNN) for controlling applications on a computer. The authors developed a dataset of 3,000 images of five hand gestures captured from five individuals for training and testing their DCNN. They used a six-layer DCNN architecture and achieved recognition accuracy of 97.7% on their dataset.

The authors believe their approach can be used for developing natural and intuitive human–computer interaction.

Dardas et al. [20] propose a real-time hand gesture recognition system for controlling applications on a computer. The authors used a combination of hand detection and hand gesture recognition techniques, including hand tracking, skin color segmentation, feature extraction, and support vector machine (SVM) classification. They achieved recognition accuracy of 92.2% on their dataset of 300 images of five hand gestures captured from three individuals. Their approach can be used to improve the efficiency and convenience of human–computer interaction [21].

In Pathak et al. [22], authors recommended the use of gesture devices that are integrated with sensors. Authors have used the sensor to record the movements of the hand in a few limited directions that could hold control over the slides. The gesture device and slides communicate with RF transmission and reception. Accelerometer and flex sensors were used for gesture recognition. They detect the flexing and movements of the hands and detect the movements by axis tracking the hands. The accelerometer traces slide switches. Many actions are happening; hence different threshold levels are set for every action. Sensors send analog signals as their output. The ADC of the microcontroller converts the received signals to digital signals. Microcontrollers use the signals generated and processes to send them to RF transmitter. The transmission of signals takes place from RF transmitter to RF receiver that gives it to the microcontroller. As the basic visual application runs, the microcontroller gives data to it. The operations are controlled using a basic visual application based on the data received.

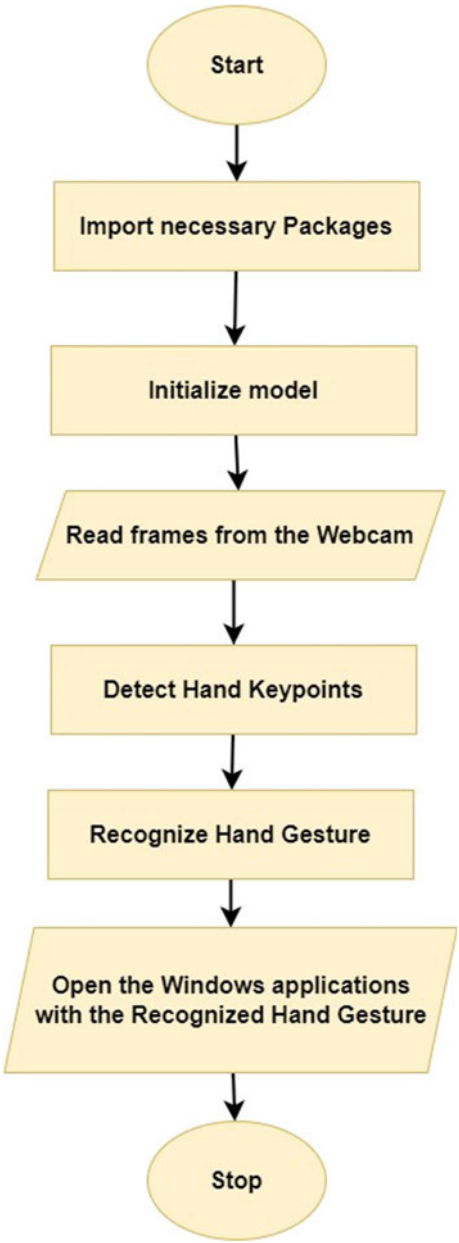
All the above-mentioned models have gone through a tough development process of using gesture datasets and training them to recognize the hand and then identify the gesture and perform the required action. Thus, the proposed model includes various other features of the slide show to be handled using the gesture, such as saving the slide with annotations, changing the color of the pointers, undoing the last annotations, along with handling the various Windows applications installed in the system with the gesture we can able to open the application without touching the screen. The proposed system also uses lightweight models to recognize and identify the gestures performed in front of the camera using Media-Pipe and its models, such HandTrackingModel.

3 Proposed System

The proposed system involves several steps to recognize hand gestures and launch the corresponding applications. Initially, the necessary libraries and modules are imported, including OpenCV, NumPy, AppOpener, TensorFlow, Keras, and Media-Pipe [23]. Google Media-Pipe is an open-source library that provides a number of pre-trained models for various computer vision tasks, including hand tracking. Media-Pipe employs a deep learning-based model capable of detecting and localizing numerous places on the hand, such as finger tips, palm center, and others, to detect

hand keypoints. For each observed hand, it produces a total of 21 important points. The Media-Pipe is initialized by creating a Hands instance, which enables the system to process and detect hand landmarks. The architecture of the proposed model is given in Fig. 1.

Fig. 1 Model architecture



The pre-trained gesture recognition model is loaded using the Keras load_model function, which allows the system to recognize various hand gestures. Using the hand's keypoints identified by the Media-Pipe hand tracking model, one can infer the gestures produced by the hand while recognizing hand movements. Additionally, class names for the gestures and corresponding application names are loaded, enabling the system to map the predicted gesture class to the corresponding application name.

The webcam is initialized to capture frames at rate of 30 frames per second, and an infinite loop is initiated to process each frame using Media-Pipe to detect hand landmarks. If a hand is detected, the landmarks are used as input for the pre-trained model to recognize the gesture. The predicted gesture class is then mapped to the corresponding application name, and the AppOpener module is used to launch the application.

Furthermore, OpenCV displays the predicted gesture and corresponding application name on the video frame. The video frame is displayed in an OpenCV window until the user presses the 'q' key to quit the program. Once the program is ended, the webcam is released, and all windows are destroyed to ensure the proper functioning of the system. Overall, the proposed system enables efficient recognition of hand gestures and the launching of the corresponding applications, making it a valuable tool for various applications.

The script first imports the necessary libraries and modules. OpenCV is used for image processing and displaying video frames, NumPy is used for numerical calculations, AppOpener is used for opening applications, TensorFlow is used for loading and running the pre-trained gesture recognition model, Keras is used for loading the pre-trained model, and Media-Pipe is used for detecting hand landmarks. Then we initialize Media-Pipe by creating a Hands instance. The max_num_hands parameter is set to 1 to detect only one hand at a time, and the min_detection_ confidence parameter is set to 0.7 to ensure that only confident hand detections are used.

Next, the pre-trained gesture recognition model is loaded using the Keras load_ model function. The model was previously trained using TensorFlow and Media-Pipe and saved to a file named 'mp_hand_gesture'. The class names for the gestures and the corresponding application names are then loaded. The class names are stored in a list named class names, while the application names are stored in a list named apps. The mapping between the class names and application names is done by index, so the nth class name corresponds to the nth application name. The class names are the names of the gestures that the model is trained to recognize. Here, the class names are defined in a list called 'class Names'. The names of the gestures are 'okay', 'peace', 'thumbs up', 'thumbs down', 'call me', 'stop', 'rock', 'live long', 'fist', and 'smile'. These names are defined in the same order as the output classes of the trained model.

We then initialize the webcam using the OpenCV Video Capture function with an argument of 0, which indicates the default camera. The script then starts an infinite loop to capture frames from the webcam. The script first flips the frame vertically for each frame using the OpenCV flip function. This is done to ensure that the video feed is not mirrored. Then it converts the frame from the BGR color space to the

RGB color space using the OpenCV color function. This is done because Media-Pipe expects RGB input. Then it processes the frame using Media-Pipe to detect hand landmarks. If a hand is detected, the landmarks are extracted and used as input to the pre-trained gesture recognition model for prediction.

The predicted gesture class is then mapped to the corresponding application name using the class names and app lists. If the predicted gesture is 'okay', no application is launched. If the predicted gesture is 'peace', the Microsoft Word application is launched. If the predicted gesture is 'thumbs up', the Microsoft Excel application is launched. If the predicted gesture is 'thumbs down', the PowerPoint application is launched. If the predicted gesture is 'call me', the Discord application is launched. If the predicted gesture is 'stop', the Microsoft Teams application is launched. If the predicted gesture is 'rock', the Windows Settings application is launched.

This proposed system also gives an effective hand-free PowerPoint interaction. The system uses the pre-trained models to identify the gestures performed in front of the camera, and the respective action are assigned when the gesture is performed. Initially, the PowerPoint is converted into the image folder where each slide of the PowerPoint (PPT) is converted into the image format as either JPG or JPEG or PNG. The MS PPT as the provision to directly convert a PPT into an image folder, which can be further used for our gesture-based control, is shown in Fig. 2.

A User Interface is also developed with the help of the Tkinter framework in Python. The UI takes the path of the image folder, and it class the procedure to handle the file with the gestures recognized from the camera-generated frame by OpenCV. The list of actions with their respective gestures is listed below in Table 1.

The proposed algorithm basically follows the flow represented in the Fig. 1. The algorithm for gesture-based control of PowerPoint presentation is below.

Algorithm 1: Gesture-based control for PowerPoint presentation

procedure Gesture (Source address)

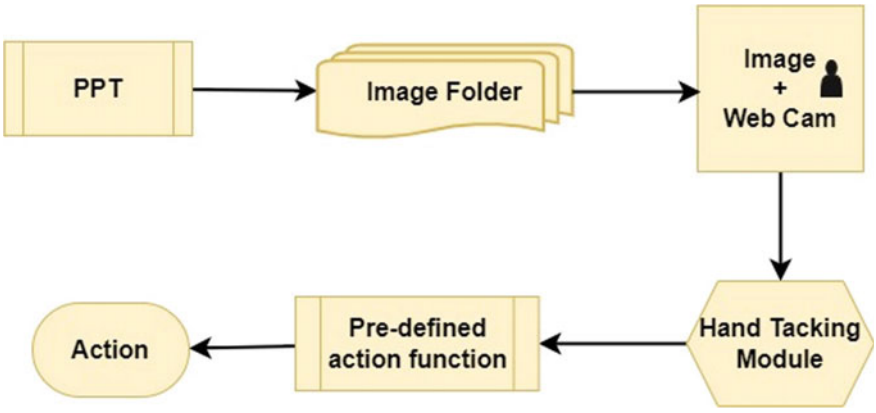


Fig. 2 PowerPoint presentation

Table 1 List of gestures and their actions for PowerPoint presentation

S. No	Gesture	Fingers involved	Action
1		Thumb pointing to left	Move to previous slide
2		Little finger up	Move to next slide
3		Index and middle finger up	Shows up the pointer on slide
4		Index finger up and movement	Draws in the slide
5		Index and little finger up	Changes color of the stokes and points
6		All fingers up	Saves the current slide with the changes
7		Index, middle, ring, and little fingers up	Undo operation

Read the Images from Source

images ← images [Source]

camera ← System [Camera]

handTracker ← HandTrackingModel

frame ← frames[camera]

```
images ← images + frame
if(handTracker(frame)) then
 temp_image ← images
fingers ← handTrackerr(frame)
if (fingers [0]) then
        images ← prev_image
else if (fingers [4]) then
images ← next_image
else if (fingers [1] and fingers [2]) then
        images ← images + pointer
else if (fingers [1]) then
        images ← images + draw the location of fingers [1]
        else if (fingers [0] and fingers [1] and fingers [4]) then
            images ← images + color_change(pointer)
else if (fingers [0] and fingers [1] and fingers [2] and fingers [3] and fingers [4])
then
        Source[images] ← images
else if (fingers [1] and fingers [2] and fingers [3] and fingers [4]) then
        images ← temp_image
show(images)
endif
end
end procedure
```

The proposed system offers several benefits that make it a useful tool for various applications.

Firstly, the system provides accessibility for individuals with physical disabilities or limited mobility. They can use hand gestures to open applications without a mouse or keyboard, which can be a significant advantage for those who find it challenging to use traditional input devices. Secondly, the system offers speed and convenience to users. A hand gesture recognition system can be faster than using a mouse or keyboard to navigate a computer interface. Users can quickly open applications with a simple hand gesture instead of clicking through menus or typing in search terms. It is also more convenient than using a mouse or keyboard when users

cannot access or use them, such as when their hands are occupied or standing at a distance from the computer. Thirdly, the system provides a contactless way of interacting with a computer, particularly relevant in the current COVID-19 pandemic. By minimizing physical contact with computer peripherals such as a mouse and keyboard, the system reduces virus transmission. Lastly, the proposed system offers novelty to users. Hand gesture recognition systems are relatively new and emerging technologies that provide a unique and exciting way to interact with a computer. They can be used to engage users and add novelty to the computer interface, making it an appealing option for various applications. The proposed system provides several benefits, including accessibility, speed, convenience, hygiene, and novelty. These advantages make it a valuable tool for individuals with disabilities, as well as for various applications, including gaming, healthcare, and education. The challenges may include occlusion, noise, hand posture fluctuations can all degrade the performance of gesture detection systems. To enhance the performance the position of the hand should be with the web cam Viewpoint. Modify the min_detection_confidence parameter in the Media-Pipe hand model initialization to provide a higher threshold for hand detection. Transfer learning may be chosen for large datasets.

4 Results

The experiment is conducted in Windows 11 in a Python environment with 12th Gen Intel(R) Core (TM) i5-1235U, 1.30 GHz with 16 GB Ram Gamming X-Box Card installed, which is essential to speed up the process. The proposed model results for hand-free windows access are shown in Figs. 3, 4, 5, 6, 7 and 8. The accuracy is determined by how well the model detects and tracks hand keypoints. Our model detects all the windows applications at different scores and the execution time for each opening application is show in Table 2. The execution performance for opening various applications shown in Fig. 9.

5 Conclusion

Hand gesture recognition systems for opening applications in Windows can be an effective and efficient way of interacting with computers, especially for people with physical disabilities or those who want to work hands-free. In this work, we used TensorFlow and Media-Pipe to train a gesture recognition model to recognize hand gestures and associate them with opening-specific applications. The resulting system performed well in detecting hand gestures, opening the associated applications in Windows, and performing hand-free PowerPoint presentations. The proposed model can be further developed into a customizable gesture-based application where the users can customize the gesture and map with the action. Instead of converting the slides into images, an extension of this system can be added to the PPT to do the

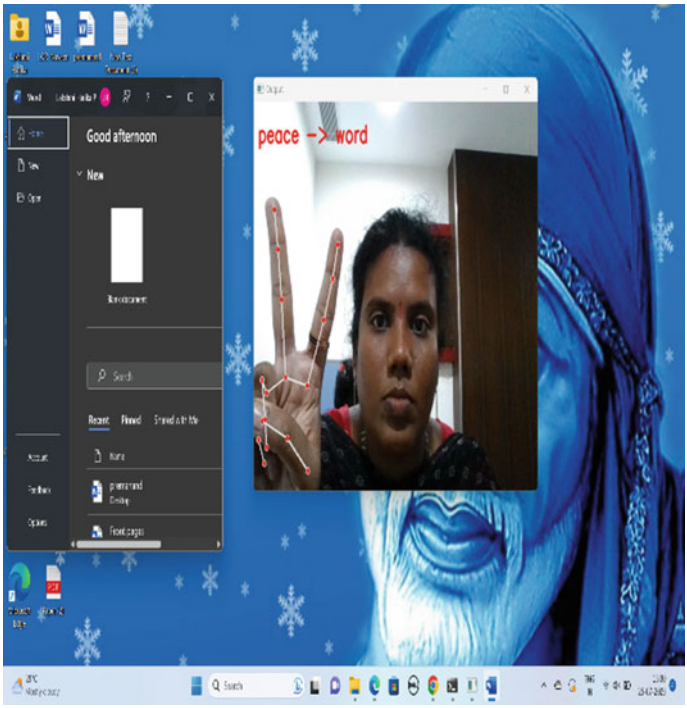


Fig. 3 Peace—Word Document

same operation within the application without depending on a third-party application which gives seamless integration and efficient workflow. This model detects all the applications very accurately achieves good detection rate. With this further development, such systems could become even more accurate and user-friendly, making computer interaction even easier and more accessible to everyone.

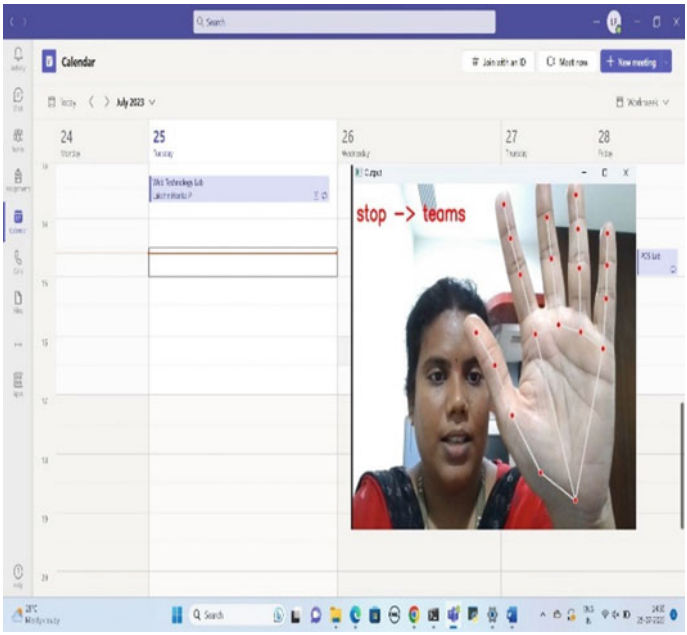


Fig. 4 Stop—Microsoft Teams

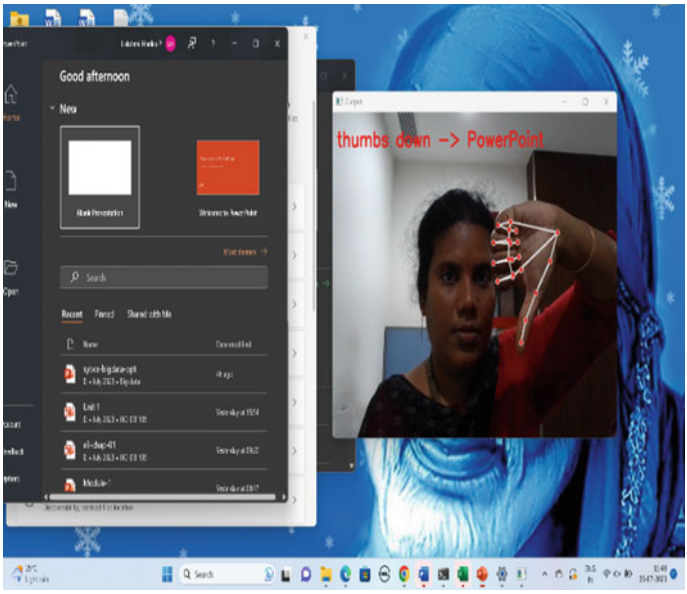


Fig. 5 Thumbs down—PowerPoint

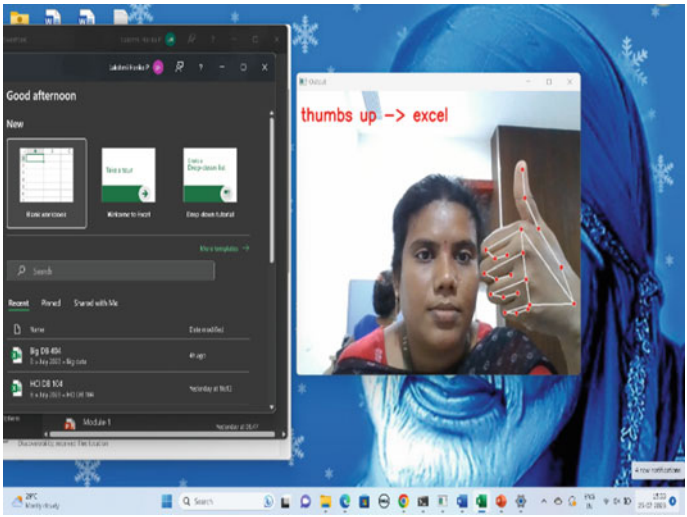


Fig. 6 Thumbs up—Excel

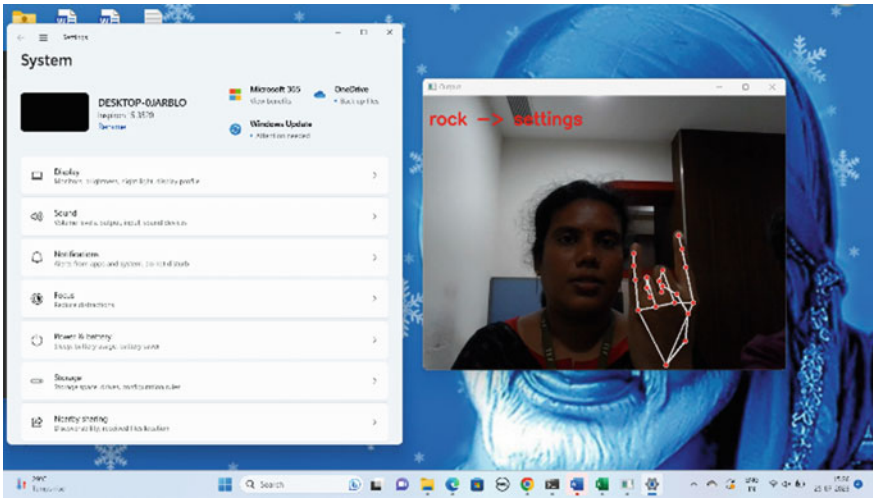


Fig. 7 Rock-settings

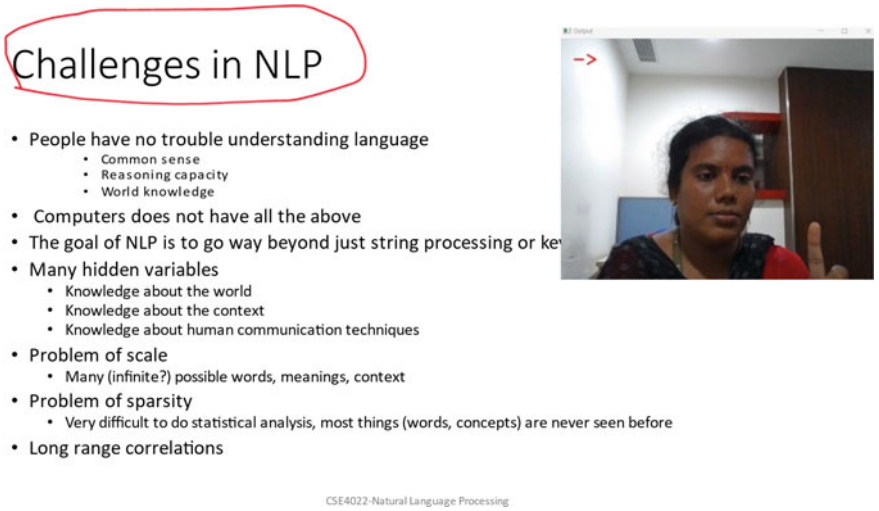


Fig. 8 Hand free presentation

Table 2 Execution time

Action	Time (ms)
Open Word	61 ms
Opening Settings	32 ms
Opening Microsoft Teams	75 ms
Opening WhatsApp	120 ms
Opening PowerPoint	65 ms
Opening Spotify	135 ms
Open Excel	95 ms
PPT presentation depends on number of slides	5 min

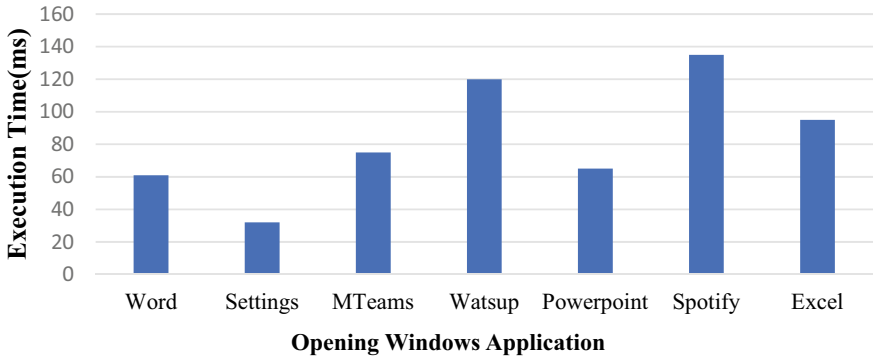


Fig. 9 Windows application performance

References

1. Ranjan C, Mitra P (2016) Hand gesture recognition using convolution neural networks. Int J Sci Res Publ 6(9):380–385
2. Sahoo JP, Prakash AJ, Pławiak P, Samantray S (2022) Real-time hand gesture recognition using fine-tuned convolutional neural network. Sensors 3:706. https://doi.org/10.3390/s22030706
3. Zhan F (2019) Hand gesture recognition with convolution neural networks. In: IEEE 20th International Conference on Information Reuse and Integration for Data Science (IRI), Los Angeles, CA, USA, pp 295–298. https://doi.org/10.1109/IRI.2019.00054
4. Islam MZ, Hossain MS, ul Islam R, Andersson K (2019) Static hand gesture recognition using convolutional neural network with data augmentation. In: Joint 8th International Conference on Informatics, Electronics; Vision (ICIEV) and 3rd International Conference on Imaging, Vision Pattern Recognition (icIVPR), Spokane, WA, USA, pp 324–329. https://doi.org/10.1109/ICIEV.2019.8858563
5. Agaian S, Pinto S, Raimundo F et al (2019) Static hand gesture recognition based on convolutional neural networks. J Electric Comput Eng 2019:1–12. https://doi.org/10.1155/2019/4167890
6. Amit ML, Fajardo AC, Medina RP (2022) Recognition of real-time hand gestures using mediapipe holistic model and LSTM with MLP architecture. In: IEEE 10th Conference on Systems, Process and Control (ICSPC), Malacca, Malaysia, pp 292–295. https://doi.org/10.1109/ICSPC55597.2022.10001800
7. Sumalatha MR, Lakshmi Harika P (2019) An intelligent system to detect human suspicious activity using deep neural networks. J Intell Fuzzy Syst IOS Press 36(5):4507–4518
8. Wameed M, Alkamachi AM (2023) Hand gestures robotic control based on computer vision. Int J Intell Syst Appl Eng 11(2):1013
9. Marais M, Brown D, Connan J, Boby A (2022) An evaluation of hand-based algorithms for sign language recognition. In: International Conference on Artificial Intelligence, Big Data, Computing and Data Communication Systems, Durban, South Africa, pp 1–6. https://doi.org/10.1109/icABCD54961.2022.9856310
10. Zhu H, Deng C, Zhu Y (2023) MediaPipe based gesture recognition system for English letters. In: Proceedings of the 2022 11th International Conference on Networks, Communication and Computing, Association for Computing Machinery, New York, NY, USA, pp 24–30. https://doi.org/10.1145/3579895.3579900
11. Islam S, Titli SR, Kabir KA, Hossain MAA, Hossain MA (2022) Improving real-time hand gesture recognition system for translation: sensor development. In: 17th Annual System of Systems Engineering Conference (SOSE), Rochester, NY, USA, pp 254–259. https://doi.org/10.1109/SOSE55472.2022.9812664
12. Haria A, Subramanian A et al (2017) Hand gesture recognition for human computer interaction. Proc Comput Sci 115:367–374. https://doi.org/10.1016/j.procs.2017.09.092
13. Khan RZ, Ibraheem N (2012) Hand gesture recognition: a literature review. Int J Artif Intell Appl (IJAIA) 3:161–174
14. Dehankar A, Sanjeev J, Thakare VM (2019) Hand gesture-based user interface for controlling mobile applications. In: Proceedings of the Fourth ICMEET. https://doi.org/10.1007/978-981-13-1906-8_59
15. Saraswati VI, Sigit R, Harsono T (2016) Eye gaze system to operate virtual keyboard. In: 2016 International Electronics Symposium (IES), pp 175–179. IEEE
16. Qi Z, Zhu W, Zhu Q (2019) Real world hand gesture interaction in virtual reality. J Phys Conf Series 1229:012027. https://doi.org/10.1088/1742-6596/1229/1/012027
17. Sagayam KM, Hemanth DJ (2017) Hand posture and gesture recognition techniques for virtual reality applications: a survey. Virtual Reality 21:91–107. https://doi.org/10.1007/s10055-016-0301-0
18. Adithya V, Rajesh R (2020) A deep convolutional neural network approach for static hand gesture recognition. Proc Comput Sci 171:2353–2361. https://doi.org/10.1016/j.procs.2020.04.255

19. Niranjani V, Keerthana R, Mohana Priya B, Nekalya K, Padmanabhan AK (2021) System application control based on Hand gesture using deep learning. In: 7th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, pp 1644–1649. https://doi.org/10.1109/ICACCS51430.2021.9441732
20. Dardas NH, Georganas ND (2011) Real-time hand gesture detection and recognition using bag-of-features and support vector machine techniques. IEEE Trans Instrum Measurem 60(11):3592–3607. https://doi.org/10.1109/TIM.2011.2161140
21. Palivela LH, Swetha S, Nithish Guhan M, Prasanna Venkatesh M (2023) Dense video captioning using video-audio features and topic modeling based on caption. In: Bindhu V, Tavares JMRS, Vuppalapati C (eds) Proceedings of Fourth International Conference on Communication, Computing and Electronics Systems. Lecture Notes in Electrical Engineering, vol 977. Springer, Singapore
22. Pathak V, Mongia S, Chitranshi G (2015) A framework for hand gesture recognition based on fusion of flex, contact and accelerometer sensor. In: Third International Conference on Image Information Processing (ICIIP), Waknaghat, India, pp 312–319. https://doi.org/10.1109/ICIIP.2015.7414787
23. Hand landmarks detection guide. https://developers.google.com/mediapipe/solutions/vision/hand_landmarker

FlameGuard: A Smart System for Forest Fire Detection and Control



Medha Wyawahare, Shreya Daga, Chashu Agrawal, Pranav Dalve, and Chirag Bhandari

Abstract Forest fires pose a huge environmental risk, wreaking havoc on ecosystems and human life. Forest fires must be detected and extinguished as soon as possible to have the least amount of harm. This study presents a forest fire detection and extinguishment system based on a Raspberry Pi, machine learning techniques, and robotic implementation. The device gathers data from the forest using sensors and cameras mounted on a robot and processes it using machine learning algorithms to detect indicators of a forest fire. The robot also has an extinguishing mechanism that can be engaged remotely to put out the flames. This system has the potential to dramatically shorten response time in detecting and suppressing forest fires, and the utilization of robotics permits access to isolated regions that would otherwise be inaccessible to humans. This system's deployment has the potential to revolutionize detection of forest fires and extinguishment systems, making them more efficient, safer, and dependable.

Keywords Fire detection · Machine learning · Raspberry Pi · Forest fire detection · Convolutional neural network

M. Wyawahare (✉) · S. Daga · C. Agrawal · P. Dalve · C. Bhandari
Department of Electronics and Telecommunication Engineering, Vishwakarma Institute of Technology, Pune, India
e-mail: medha.wyawahare@vit.edu

S. Daga
e-mail: shreya.daga20@vit.edu

C. Agrawal
e-mail: chashu.agrawal20@vit.edu

P. Dalve
e-mail: pranav.dalve20@vit.edu

C. Bhandari
e-mail: chirag.bhandari20@vit.edu

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2023

S. Shakya et al. (eds.), *Fourth International Conference on Image Processing and Capsule Networks*, Lecture Notes in Networks and Systems 798,
https://doi.org/10.1007/978-981-99-7093-3_44

1 Introduction

Ecosystems, animals, and human groups are all at risk from forest fires' devastating harm. A forest fire's ability to spread and cause catastrophic damage depends on its early detection. The creation of forest fire detection and extinguishing alarm systems has been facilitated recently by technology improvements. To find and notify authorities and first responders of forest fires, these systems employ a variety of sensors, such as infrared and smoke detectors. These methods can aid in stopping forest fires from spreading out of control and resulting in significant harm. Forest fires are a terrible occurrence that can seriously harm human life, property, and wildlife habitats. To lessen the effects of forest fires, prompt discovery and suppression are essential. Recent technological developments have enabled the creation of intelligent forest fire detection [1] and extinguishing systems that employ robotics and machine learning algorithms for the early identification and control of forest fires. One such system processes data from sensors and cameras attached on a robot that moves around the forest using a Raspberry Pi, a tiny single-board computer.

A CNN-based machine learning model has been used to detect forest fire using the Forest Fires Data Set. By minimizing the time, it takes to identify and put out forest fires, this method has the potential to drastically lessen the damage that such fires do. The method is more effective at spotting and extinguishing flames in distant regions since the use of robots also enables access to places that may be challenging for people to reach. In general, machine learning, robots, and Raspberry Pi technology have the potential to revolutionize detection of forest fires and extinguishing systems, paving the way for a safer and more effective method of defending forests and wildlife habitats.

2 Literature Review

YOLO (including tiny-yolo-voc1, tiny-yolo voc, yolo voc.2.0, and yolov3), R-CNN, and SSD models possess faster processing as compared to the others. These findings were proposed in a paper where the faster R-CNN method was run for 120,000 iterations. The authors found that YOLO had poor performance in detecting smaller cooler fires, so they added two more layers (one convolutional and one max-pooling layer with a filter of 8 to tiny-yolo-voc. This resulted in an improved accuracy rate for smaller fire detection [2].

A deep learning network and UAV-FFD platform-based approach for identifying forest fires has been developed in [3]. The program employs a huge YOLOv3 network to process photographs from the UAV instantly. The YOLOv3 detection method demonstrates excellent precision and low time cost by utilizing.

Color and region growing algorithms are applied to extract precise flame images, and cusp detection is used to determine whether the images contain flames. This

approach improves algorithm efficiency and recognition rate, meeting the real-time requirements of fire monitoring systems [4].

The technique used in [5] distinguishes between smoke and clouds; a false-color composite image was created using the smoke reflectance index, mid-infrared reflectance, and water index. While the clouds appear pink and white, the smoke appears crimson. In 2009, the approach was used on a MODIS scene of a forest fire in Russia and Mongolia. Using a training dataset, smoke pixels were identified, and the results were compared to results of a multi-threshold method.

A paper [6], identifies fire geometrical factors to measure forest fires in order to respond quickly and safely. The RGB color space is used to extract the flame zone, and by scanning each point on the image, the coordinates of each vertex are found. The geometrical properties are then used to find the terrain inclination angle, flame inclination angle, and flame area. These features can be used to calculate the direction of wind dispersion, topographical conditions, and decision-making for battling forest fires can be enhanced. The suggested system can maintain track of the detailed fire scene as well as information about the surrounding geographic environment.

Aerial monitoring can be used to detect forest fires. One such approach entails analyzing a vast library of footage of forest fires under various settings in order to extract the fire's chromatic and motion properties [7]. The approach also comprises extracting smoke using a suggested algorithm to handle the issue of thick smoke hiding the fire.

Forest fire prediction can be implemented using a machine learning model. A devised approach [8] consists of two steps: classification to determine whether a fire is present and regression to predict the size of the fire. The approach was validated and compared with other machine learning methods, showing better quality results.

A single system integrates a number of sensors, embedded processors, and an energy supply module. The Kalman filter is used to preprocess and filter sensor data, and fuzzy reasoning is used to determine the final fire signal. The system was modeled in MATLAB, and the findings demonstrate that data pretreatment removes aberrant data effectively, and the Kalman filter lowers data error, resulting in more precise fire probability estimates. A single system integrates a number of sensors, embedded processors, and an energy supply module. The Kalman filter is used to preprocess and filter sensor data, and fuzzy reasoning is used to determine the final fire signal. The system was modeled in MATLAB, and the findings demonstrate that data pretreatment removes aberrant data effectively, and the Kalman filter lowers data error, resulting in more precise fire probability estimates [9].

A research presents a system and approach for detecting forest fires in their incipient stages utilizing a network of wireless sensors. A machine learning regression approach is also developed to improve fire detection accuracy [10]. The primary power source is provided by rechargeable batteries combined with a secondary solar power supply, a solution may be easily implemented as a standalone system for extended periods of time.

The use of picture-taking visual camera to click pictures of woods in order to identify fires is another technique. Visual light camera sensors are initially installed on communication towers that are positioned in a position that faces a sizable portion

of the forest in [11]. To offer a 360° picture of a whirling motor in a forest attached to allow the camera to revolve.

A project is developed using an algorithm and sensors to detect changes in sensory parameters, such as gas and fire events, and send updates to experts through an IoT landing. The system aims to reduce error perception and provide a sustainable ecosystem through the use of D2D association conventionality. The system can be used in both residential and industrial settings [12].

Development of wireless sensor nodes for detection and monitoring has been done for the Riau Province of Indonesia. The design and analysis of the WSNs use a mathematical approach to determine the area that needs to be covered. Air temperature, humidity, haze, and CO_2 sensors are highlighted as the basic parameters for detecting fire hot spots in land or forest areas. The proposed sensor nodes use ZigBee technology, which allows for low power consumption and long battery life [13]. A model achieved training accuracy of 92.27% and 89.57% test accuracy with a stochastic gradient descent optimizer and avoided the underfit/overfitting on the model with the help of the stochastic gradient descent (SGD) algorithm [14].

3 Proposed System

3.1 Dataset

The dataset used for this project is primarily taken from Kaggle, Forest Fire images [15]. This dataset is readily available and free to use. Several additional smaller datasets were combined and merged to form this dataset. The various proprietors and developers of the original smaller datasets deserve all the praise. Additionally, Google Images was used to download each and every image in the Test Data folder. Web scrapping was implemented so as to largen the dataset. The dataset contains images from various forests and plantations which have smoke and fire being seen in the images labeled with fire. The entire dataset is divided into the Test Data and Train Data folders, which can be seen in Figs. 1 and 2. Test Data includes a total of 50 pictures of fire and non-fire. The Train Data includes 5000 photographs total of both fire and non-fire. Each of these files is split into two sub-folders of both types of fire, each of which contains 25 photographs under Test Data and 2500 images under Train Data.

3.2 Hardware Used

1. Raspberry Pi: The Raspberry Pi is a small, inexpensive, and adaptable computer board that allows anyone to learn programming and make numerous projects. It supports a wide range of applications, from education to home automation

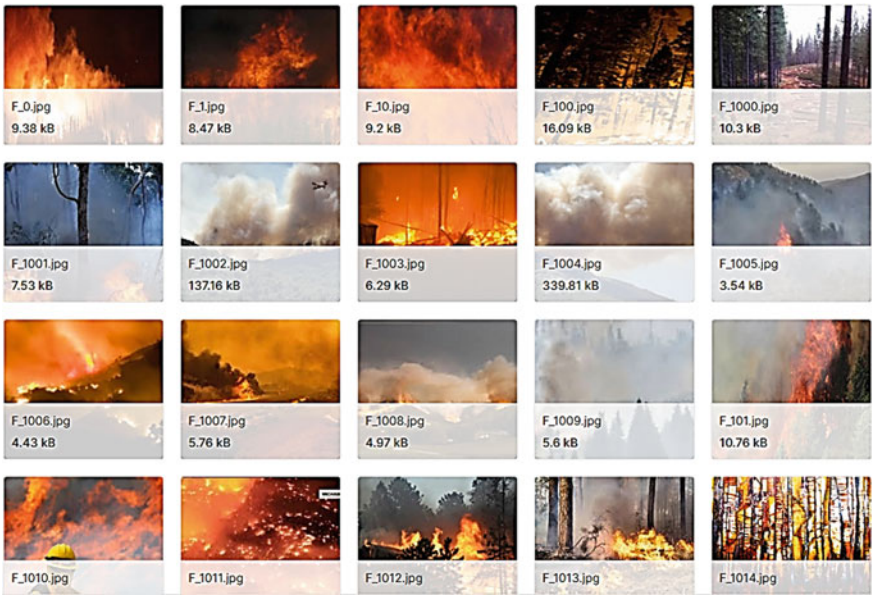


Fig. 1 Images labeled with fire

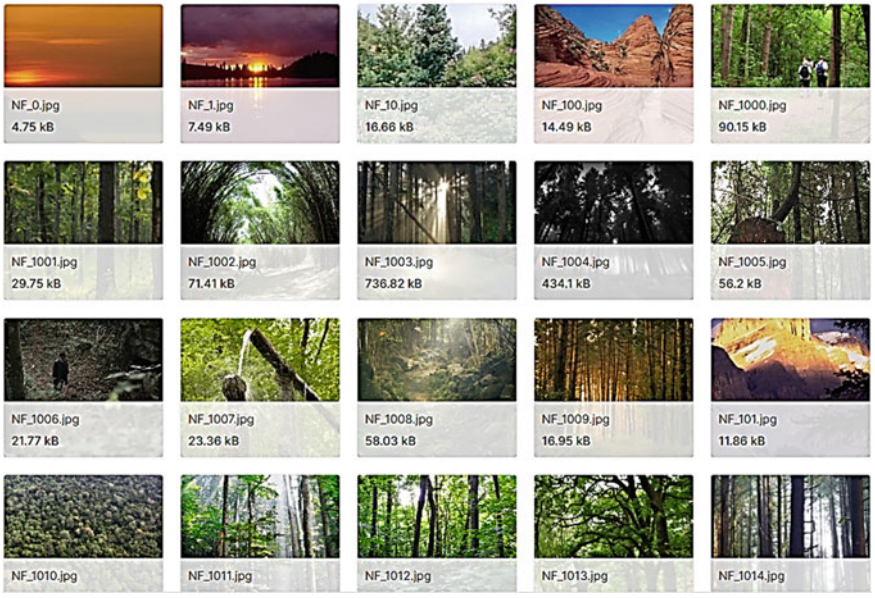


Fig. 2 Images labeled with non-fire

and robotics, and has GPIO pins for hardware connection. It operates on Linux-based operating systems. Raspberry Pi has been used for this system because of its ability to work with a machine learning model. The machine learning model has been deployed on the microprocessed and uses its camera module for fire detection.

2. Flame sensor: A flame sensor is an electronic device that detects the existence of smoke or fire in its surroundings. It intended to offer an early warning signal by activating alarms or alert systems, allowing for rapid evacuation and fire suppression procedures, so preventing property damage and ensuring safety.
3. Battery-operated DC motor: Also known as a brushed DC motor, it is a type of electric motor that transfers electrical power to the motor's revolving armature via a commutation system utilizing brushes. It combines simple and cost-effective rotating motion with high torque production, making it commonly employed in robotics and small-scale manufacturing.
4. Servo motor: A servo motor is a small and precise rotary actuator that uses feedback control to position its output shaft precisely. It is widely utilized in robotics and automation systems, because it allows for regulated and precise movements with high torque capabilities.
5. L293 motor driver: The L293 motor driver is an integrated circuit that can control two direct current motors or a single stepper motor. It has bidirectional control and can handle high current and voltage, making it useful for robotics and other motor control applications.

3.3 Machine Learning Pipeline

For the purpose of detecting forest fires, the dataset is preprocessed. Using the OpenCV library, the colored pictures are extracted from the dataset and changed to grayscale versions. Following that, an array function is used to store the obtained grayscale pictures. Each image has a caption that says, "Fire or Non-Fire" and has the robust photographs cropped and resized as needed. The labeled, resized picture is added to the list of training datasets along with a NumPy array representing it. In this project, libraries like NumPy, Sklearn, Keras, and TensorFlow were used. The input features were taken out of the pretrained data and used for the train–test split in the SKLearn package to validate them. The convolutional neural network model utilizing the Keras API is the machine learning model utilized in this forest fire detection model. A number of convolutional layers, including activation and max-pooling, make up the model architecture. When building a model from scratch using the train dataset and deploying it to the test dataset, the epoch and batch size are specified.

4 Methodology

The machine learning model is deployed using the Raspberry Pi. The model of machine learning recognizes fire as soon as the robot receives electricity. The Raspberry Pi is activated when fire is found, turning on the fire sensors. In order to navigate exactly toward the fire spot, there are three flame sensors in the robot. After arriving at the fire spot, the motor pump is turned on, and the fire is put out using the water pump. Hardware connections were constructed as shown in Fig. 3. The robot is powered using a 10,000 mAh power bank which provides power to the Raspberry Pi; hence, as a fixed current is being given to the microprocessor, there is no fluctuation of supplied current neither any measures are required to take care of the same.

In order to identify and put out forest fires without the need for human involvement, the proposed system combines the hardware and machine learning model in this manner. Using the preprocessed data, a convolutional neural network (CNN) model is built and trained. CNN is programmed to learn and extract significant features from input photographs that distinguish between instances of fire and non-fire. The model

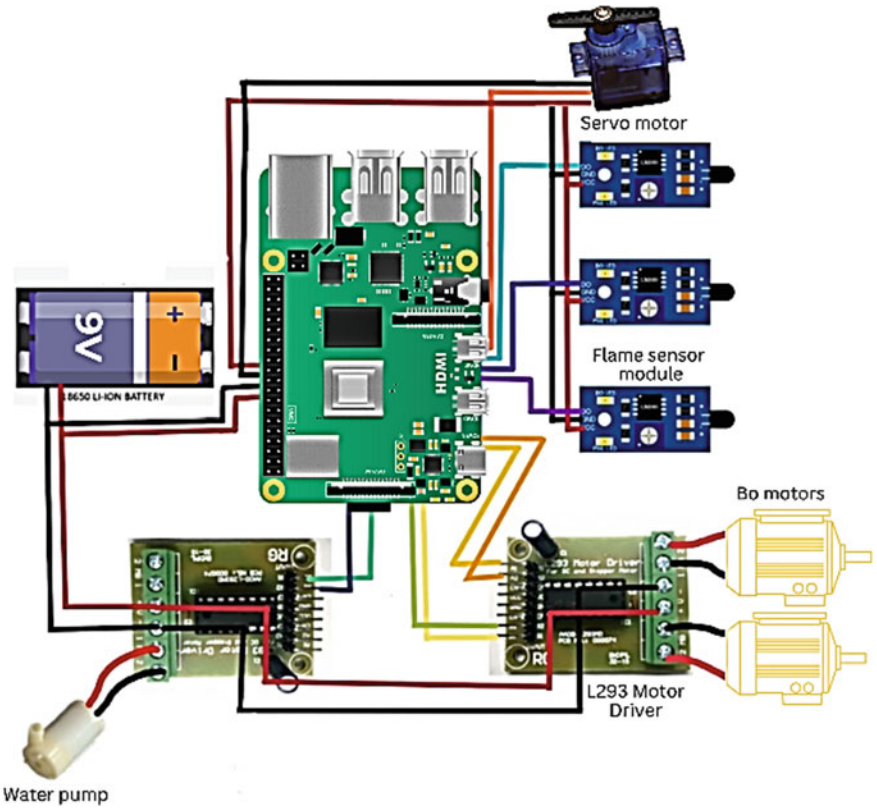


Fig. 3 Integrated circuit diagram

is trained with the help of positive, negative examples, and it is optimized to minimize classification error. Once trained, the model can be used to predict whether or not an image contains a fire. The ML model processes the input image from the flame sensors using the trained CNN. The model generates a likelihood or classification indicating the existence of fire. If the ML model detects a fire, the flame sensors direct the robot to the area of the fire. To precisely localize the fire, the robot is outfitted with flame sensors. The ML model continually evaluates the sensor data for identifying positions of the flames, and to adapt robot's movement accordingly. As the flame sensor works on the radiation sent by flame, there is no detection of smoke. The flame sensor converts the flame radiation into relevant photoelectric signal to get the desirable output. The workflow of the system can be seen in Fig. 4.

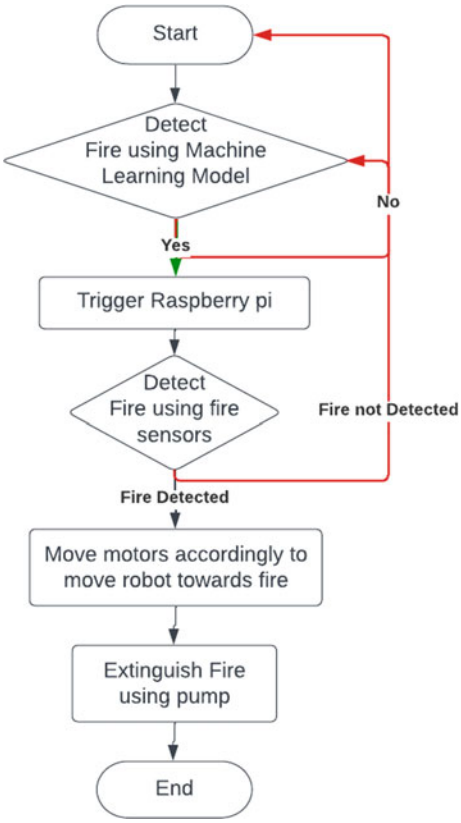


Fig. 4 Workflow of the system

5 Results and Discussions

The dataset was split into training and testing sets with a ratio of 70:30 (training:testing). The crop size used in the preprocessing of data is of (600, 1280) step size which means images are resized by a factor of 0.5 in both directions. With the help of the machine learning pipeline, the features from the images were extracted and reshaped into a 4D array of shape (number_of_samples, 300, 640, 1) where 300 and 640 represent the height and width of the image and 1 indicating conversion to grayscale images. Random forest and support vector machine models gave an accuracy of 83% and 88% each, but the convolutional neural network model outperformed the two. Additionally, cross-validation and hyperparameter tuning in all the models help achieve the stated accuracy. The final model is created using several convolutional and polling layers. Initially, an accuracy of about 86% was obtained using the convolutional and ReLU layer. Further, with the help of batch normalization process, overfitting of the model was prevented. The updated convolutional neural network model gave an accuracy of about 94%. Adam optimizer also helps to add to the accuracy of the machine learning model. The training loss and accuracy graph for an epoch size of 1000 can be seen in Fig. 5. The machine learning models training time varies on the hardware on which it is being trained, usually taking around 10–15 min. The learning rate of the model is same as that of Adam optimizer which is close to 0.001.

The Adam optimizer minimizes the loss function and improves the performance on the trained dataset. When fire is detected, the machine learning model sends a trigger signal to Raspberry Pi. The three flame sensors help achieve a precise fire location. There are a total of four motors in the proposed system, where two of each

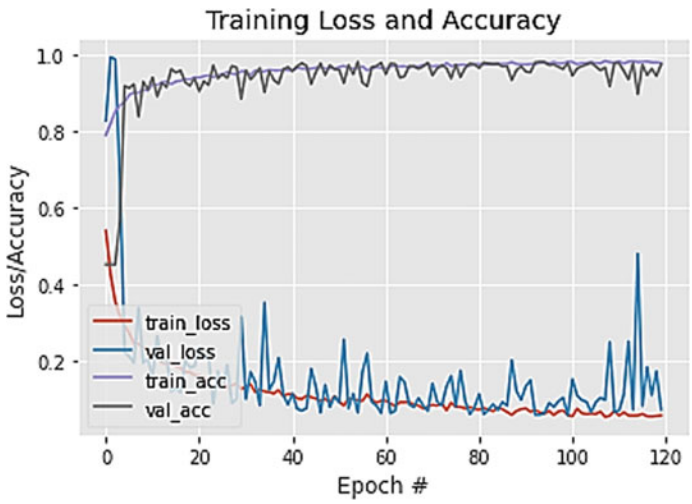


Fig. 5 Training loss and accuracy for model

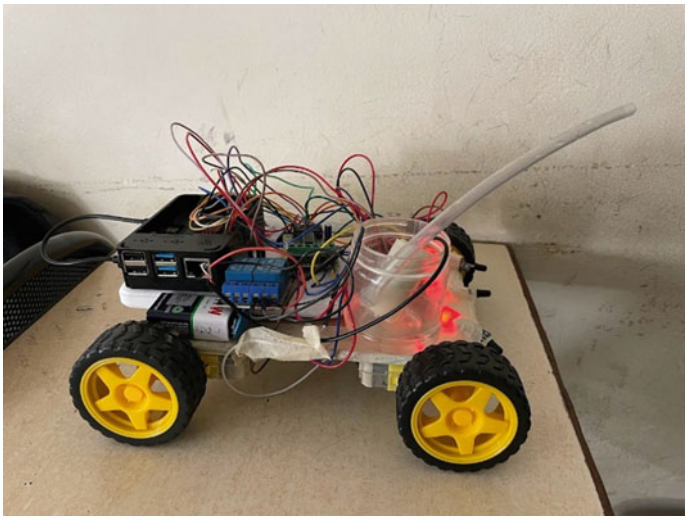


Fig. 6 Fire extinguishing robot in stationary position

are made to move in one direction in order to reach the fire location. The robot's movement is controlled by changing the motor directions based on flame sensor readings. As soon as the robot reaches the fire location, the water pump is activated resulting in both left and right motors being simultaneously activated for a short duration of time, i.e., 500 ms to stop the robot's movement and pumps out water to put off the fire. The container on the cart consists of water along with the water pump as seen in Fig. 6.

The Raspberry Pi and all the mechanical pieces that made up the robot vehicle moved and put out fire in unison with the motors. The movement of the robot in the direction of the fire can also be seen through Figs. 7 and 8, where the robot moves closer to the detected fire location. This technology detects forest fires, navigates to the fire area, and performs automated or remotely controlled fire suppression measures by merging machine learning techniques with sensor integration and robotic control. It offers an effective and perhaps life-saving technique for dealing with forest fires.

6 Conclusion and Future Scope

Developing a machine learning model-based fire-fighting robot with a Raspberry Pi has a lot of potential for improving fire detection and response systems, in conclusion. The robot can quickly and correctly identify fire incidents by analyzing real-time sensor data, such as thermal imaging or video feeds, by utilizing machine learning techniques. Real-time processing and decision-making are made possible by the

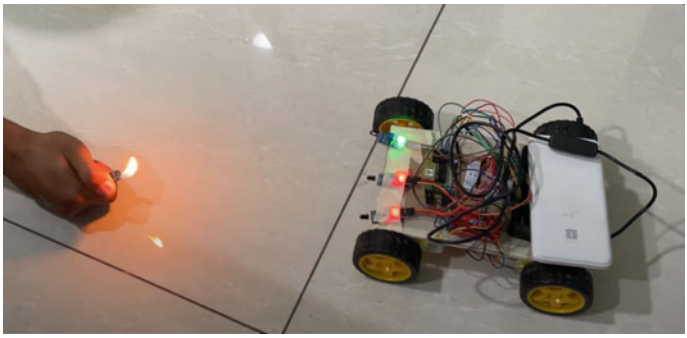


Fig. 7 Fire extinguishing robot detecting fire

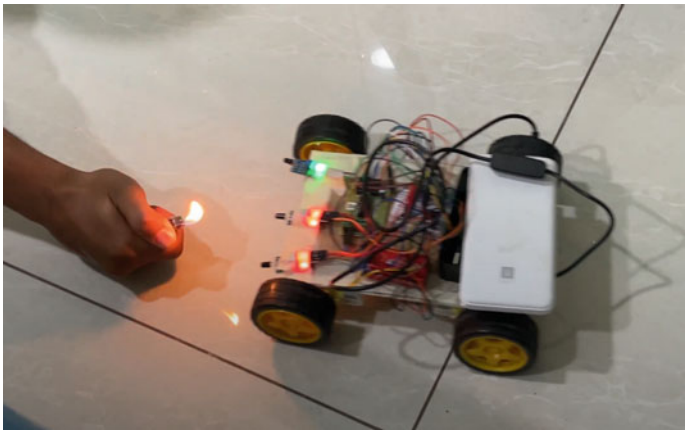


Fig. 8 Fire extinguishing robot moving towards the fire source

Raspberry Pi, a powerful and reasonably priced platform for putting the machine learning paradigm into practice. The model may learn patterns and traits linked to fires by being trained on a varied collection of photographs or videos connected to flames. The robot can aid in early fire detection and fire mitigation in regions that are prone to fire, such as woods or industrial complexes. The robot's sensors allow it to continually examine its surroundings and wirelessly communicate data to a centralized control system for processing. When a fire is detected, the robot may autonomously explore the scene, identify the fire's origin, and then take the necessary action, such as deploying fire suppression systems or notifying nearby workers.

Iteratively training the machine learning model with fresh data and fine-tuning the model's parameters can increase accuracy and performance over time. The system is consistently updated and maintained to ensure its dependability and efficiency in firefighting situations. Overall, integrating a machine learning model into a fire-fighting robot using Raspberry Pi has the potential to revolutionize fire detection

and response capabilities, enabling more efficient and timely operations to limit the destructive impact of fires.

References

1. Madhwaraj KG, Asha V, Vignesh A, Akshay Shinde S (2023) Forest fire detection using machine learning. In: 2023 IEEE 12th International Conference on Communication Systems and Network Technologies (CSNT), Bhopal, India, pp 191–196. https://doi.org/10.1109/CSNT57126.2023.10134684
2. Wu S, Zhang L (2018) Using popular object detection methods for real time forest fire detection. In: 2018 11th International Symposium on Computational Intelligence and Design (ISCID), Hangzhou, China. https://doi.org/10.1109/ISCID.2018.00070
3. Jiao Z et al (2020) A YOLOv3-based learning strategy for real-time UAV-based forest fire detection. In: 2020 Chinese Control And Decision Conference (CCDC), Hefei, China, pp 4963–4967. https://doi.org/10.1109/CCDC49329.2020.9163816
4. Li J, Jiang X (2009) Forest fire detection based on video multi-feature fusion. In: 2009 2nd IEEE International Conference on Computer Science and Information Technology, Beijing, China, pp 19–22. https://doi.org/10.1109/ICCSIT.2009.5234862
5. Nagatani JK, Kawano K (2012) A new technique for visualization of forest fire smoke plumes using MODIS data. In: 2012 IEEE International Geoscience and Remote Sensing Symposium, Munich, Germany, pp 2380-2383. https://doi.org/10.1109/IGARSS.2012.6351016
6. Li G-H, Zhao J, Wang Z (2006) Research on forest fire detection based on wireless sensor network. In: 2006 6th World Congress on Intelligent Control and Automation, Dalian, pp 275-279. https://doi.org/10.1109/WCICA.2006.1712316
7. Dang-Ngoc H, Nguyen-Trung H (2019) Aerial forest fire surveillance—evaluation of forest fire detection model using aerial videos. In: 2019 International Conference on Advanced Technologies for Communications (ATC), Hanoi, Vietnam, pp 142–148. https://doi.org/10.1109/ATC.2019.8924547
8. Sineva S, Molovtsev MD (2020) An integrated approach to the regression problem in forest fires detection. In: 2020 Systems of Signals Generating and Processing in the Field of on Board Communications, Moscow, Russia, pp 1–4. https://doi.org/10.1109/IEEECONF48371.2020.9078645
9. Wang T, Hu J, Ma T, Song J (2020) Forest fire detection system based on Fuzzy Kalman filter. In: 2020 International Conference on Urban Engineering and Management Science (ICUEMS), Zhuhai, China, pp 630–633. https://doi.org/10.1109/ICUEMS50872.2020.00138
10. Yan F, Xu X, Han N (2010) A kind of identification method of geometric parameters for forest fire. In: 2010 2nd International Conference on Signal Processing Systems, Dalian, China, pp V1-91–V1-94. https://doi.org/10.1109/ICSPS.2010.5555608
11. Owayjan M, Freiha G, Achkar R, Abdo E, Mallah S (2014) Firoxio: Forest fire detection and alerting system. In: MELECON 2014–2014 17th IEEE Mediterranean Electrotechnical Conference, Beirut, Lebanon, pp 177–181. https://doi.org/10.1109/MELCON.2014.6820527
12. Divya TK, Dharshini P (2019) IoT enabled forest fire detection and early warning system. In: 2019 IEEE International Conference on System, Computation, Automation and Networking (ICSCAN), Pondicherry, India, pp 1–5. https://doi.org/10.1109/ICSCAN.2019.8878808
13. Shixing L, Yongming Z, Weiguo S, Xia X (2010) An enhanced algorithm for forest fire detection based on MODIS data. In: 2010 International Conference on Optoelectronics and Image Processing, Haikou, China, pp 200–203. https://doi.org/10.1109/ICOIP.2010.332
14. Rahul M, Shiva Saketh K, Sanjeet A, Srinivas Naik N (2020) Early detection of forest fire using deep learning. In: 2020 IEEE region 10 conference (TENCON), Osaka, Japan, pp 1136–1140. https://doi.org/10.1109/TENCON50793.2020.9293722
15. Forest Fire Dataset. https://www.kaggle.com/datasets/mohnishsaiprasad/forest-fire-images

Classification and Analysis of Chilli Plant Disease Detection Using Convolution Neural Networks



Zameer Gulzar, Sai Chandu, and K. Ravi

Abstract A crucial but time-consuming duty in agricultural practices is spotting disease in any crop. Chillies are a vital income crop in Telangana whose cultivation is essential for preserving food security. However, a number of diseases and pests can cause significant productivity losses while growing chillies. Early detection of these diseases is crucial for efficient management and the reduction of yield losses. The goal of this study is to analyze best convolution neural network model that can recognize and categorize chilli plant illnesses with accuracy. In this chapter, the classification of the four diseases (leaf curl, leaf spot, white fly, yellowish leaf) were identified and examined using a pretrained deep learning networks (VGG16, VGG19, Inception V3, and Alex-net) using the dataset of chilli leafs with and without augmentation. The methodology involves a dataset of 2000 latest images of chilli leaves (apart from imagenet dataset) with four different illnesses to train and test the model, with various inputs and mini-batch sizes. The model's performance of all the networks is assessed using a test dataset, and the results showed that Inception V3 achieved the highest accuracy rates of 93% in identifying these four disorders, followed by VGG16 (85%) and VGG19 (83%), respectively. The suggested methodology can be used by farmers as a decision support system to monitor the health of their crops and take timely and appropriate action that will ultimately help them sustain the output of chillies.

Keywords Chilli · CNN model · Disease · Inception V3 · Farmers

Z. Gulzar (✉) · S. Chandu · K. Ravi
SR University, Warangal, Telangana, India
e-mail: zameer.gulzar@sru.edu.in

K. Ravi
e-mail: k.ravi@sru.edu.in

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2023

S. Shakya et al. (eds.), *Fourth International Conference on Image Processing and Capsule Networks*, Lecture Notes in Networks and Systems 798,
https://doi.org/10.1007/978-981-99-7093-3_45

1 Introduction

Since agriculture employs more than 60% of the population and contributes around 17% to the country's GDP, it is a crucial sector of the Indian economy. However, a variety of issues, such as climate change, pests, illness and other issues continue to endanger food security [1]. In India, the cultivation of chilies is a significant income crop that is essential for preserving food security. Telangana produces the most chilies in India, accounting for 49% of the nation's total production [2]. The most popular chilies in India are those cultivated in the Guntur region of Telangana because of their distinctive flavor and color. However, a number of ailments and diseases can result in severe yield losses in the production of chilies. Therefore, the identification and treatment of disease are crucial for improving the growth and yield of such agricultural plants.

Since the vast number of plant diseases show visible symptoms, and well-known pathologist with good observation skills diagnoses the infected plant leaves disease simply by using visual observation. For farmers, it is extremely challenging and expensive to get subject-matter experts for observing plant diseases [3]. Even seasoned pathologists occasionally struggle to identify some illnesses due to the enormous variety of plants, variances in the progression of plant diseases caused by climatic changes, and the rapid spread of diseases to areas where they have never been observed before [4]. Therefore, a sophisticated system for diagnosing plant diseases was therefore necessary to periodically check the agricultural fields. Agronomists can benefit greatly from the availability of knowledgeable and intelligent technologies that can accurately and automatically diagnose plant disease. Crop productivity has greatly increased thanks to the adoption of machine learning techniques in agriculture [5]. The majority of recent research has concentrated on using computer vision and deep learning to identify plant diseases. The HAAR, histogram of oriented gradients, scale invariant and feature transform, speeded-up robust features, Support Vector Machine (SVM), image segmentation using artificial neural networks, K-Nearest Neighbors' (KNN), and K-means clustering techniques have all been used in the past to detect plant diseases [6]. Despite machine learning producing good results, these models fell short of those obtained by deep learning methods based on Convolution Neural Networks (CNNs). The golden age of machine learning will benefit from deep learning which have raised accuracy levels. Deep neural networks have shown great success in solving image recognition challenges, and they have steadily risen to the top of the field in many areas. Deep learning algorithms for Image recognition have been shown to be trustworthy [7].

The creation of automated systems that can diagnose diseases more quickly and accurately is now possible because of advancements in artificial intelligence technology [8]. Transfer learning applies a neural network that has already been trained for one task to a new, related task. The time needed for model design and training can be cut down with transfer learning approaches. The common image classification transfer learning algorithms are Alex-Net, Dense-Net, VGG16, Inception-V3,

Mobile-Net, and Res-Net [9, 10]. The evaluation revealed that the CNN implementation produced excellent results and outperformed the results obtained by cutting-edge machine-learning approaches. In this chapter, we tested various CNN architectures, including VGG16, VGG19, Inception V3, and Alex-net to identify plant illnesses based on photos of healthy and diseased leaf images.

The parameters were adjusted and tested comprehensively, with several dropout values between 0.2 and 0.8, as well as varied batch sizes between 32 and 180. Then the model was evaluated with various learning rates in the range of 0.01–0.0001 in order to optimize it and also different epochs were used to train the models. The evaluation of models implemented was done against other deep-learning models. The Inception V3 model performed better, according to the results, in terms of accuracy and training time.

To build reliable classifiers for plant disease diagnosis, we needed a huge, validated dataset of images of sick and healthy plants to test on the present available classifiers. To address this issue, we started collecting about 2000 images of both healthy and ill crop plants. The remainder of the chapter is organized as follows: Sect. 2 presents implementation method. Sections 3 and 4 demonstrates performance measurement and results and discussion. Finally, Sect. 4 will display some application followed by the conclusion of the chapter.

2 Material and Methods

The models implemented are dependent on the CNN architecture; the general CNN architecture is shown in Fig. 1 the layers that must exist in a CNN for classification purposes are Convolution Layer, Pooling Layer, Flatten Layer, and Dense Layer. The function and operation of these layers are discussed further below [11]

The convolutional layer consists of a set of learnable filters or kernels given in Fig. 2 that are convolved with the input data to extract spatial features from the input image. The filters slide across the input data, performing a dot product at each position to produce a feature map. The dot product operation computes the similarity between the filter and the corresponding patch of the input data, and the resulting feature map represents the activation of the filter across the entire input data. The size, number,

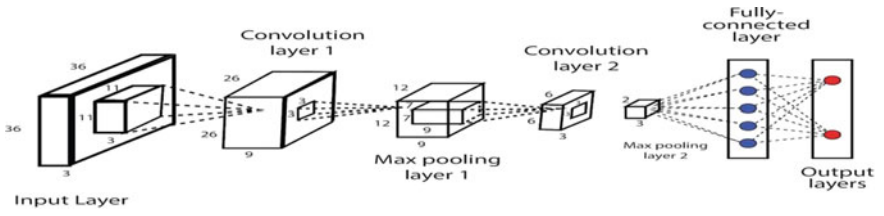


Fig. 1 Convolution neural network

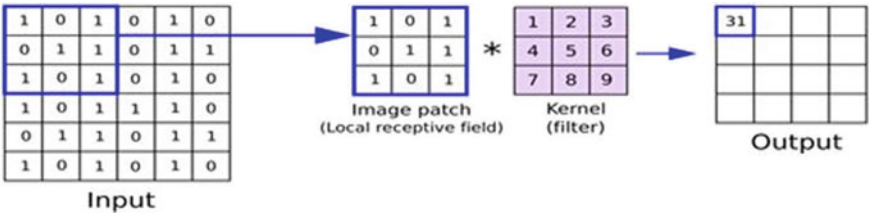


Fig. 2 Convolution filter output calculation

and stride of the filters in the convolutional layer can be adjusted to control the number and size of the output feature maps. Each filter in the convolutional layer learns a specific feature of the input data, such as edges, corners, or textures, through the back-propagation algorithm during the training process.

A pooling layer in Fig. 3 merge the outputs of neuron clusters at the previous layer into a single neuron at the subsequent layer, the pooling layer's goal is to lower the hidden layer's dimensions. The output feature maps from the convolutional layers in CNNs are down sampled using pooling layers. Maintaining the most pertinent data while reducing the input's spatial size is the core aim of pooling. By doing so, overfitting might be prevented, and the networks computational and parameter needs might be reduced. Pooling layers are typically inserted after convolution layers and before fully connected layers in CNN architecture. Pooling layers operate on each feature map separately and perform a fixed operation, such as max-pooling or average pooling, on small sub-regions of the input feature map [12]. The purpose of the pooling layer is to reduce the dimensionality of the feature maps, making the subsequent layers computationally more efficient and reducing the risk of over fitting. The size of the pooling window, or kernel, and the stride of the window determine the amount of down sampling that occurs. For example, a max-pooling operation with a 2×2 kernel and a stride of two reduces the spatial dimensions of the feature map by half, while retaining the most salient features. Overall, pooling layers are an important component of CNNs, since it shrinks the input space while retaining pertinent data, leading to more efficient and effective image recognition and classification while cutting down on administrative burden [13].

In CNN, the flatten layer shown in Fig. 4 converts the output of the previous convolutional and pooling layers into a 1D feature vector, which can be fed into

Fig. 3 Max pooling

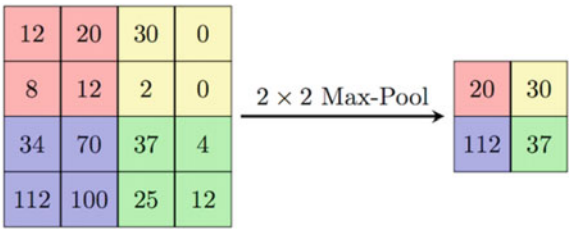
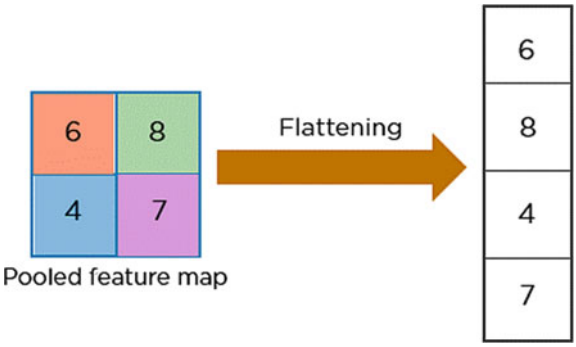


Fig. 4 Flatten layer

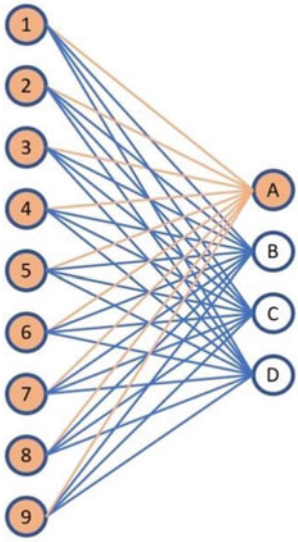


a fully connected neural network to perform classification or regression tasks. The output of the convolutional and pooling layers is typically a high-dimensional feature map with each element representing a specific feature of the input image. The flatten layer reshapes the feature map into a 1D feature vector, which can be processed by a fully connected neural network. Overall, the flatten layer remains as an important component in CNNs to combine the local and global information learned by the convolutional and fully connected layers, respectively, to perform complex tasks on high-dimensional input data [14, 15].

The dense layer shown in Fig. 5 connects every neuron in the current layer to every neuron in the previous layer. It is also called a fully connected layer as each neuron in the current layer is connected to all neurons in the previous layer. In CNNs, dense layers are typically added after the convolutional, pooling and flatten layer. The output of the flatten layer, which is a 1D feature vector, is fed into one or more dense layers for further processing. The purpose of including a dense layer is to perform complex computations on the high-dimensional feature vector obtained from the previous layers, allowing the model to learn complex hierarchical representations of the input data. Overall, dense layers remain as an important component of neural networks, allowing the model to learn complex representations of the input data and perform a wide range of tasks, ranging from simple classification to more complex tasks such as Natural Language Processing (NLP) and computer vision.

The CNN models, training procedure, experimental set-up, and dataset preparation are all fully described in this section. The pipeline for the proposed model for detecting plant leaf diseases begins with dataset preparation and ends with model prediction. Acquiring the right dataset is critical for the success of any deep learning or machine learning model, including ours. To acquire the dataset, we have pleated chilli plant images at various stages of its growth, disease severity, and lighting conditions. Then we labeled the images according to different classes of chilli plant diseases, such as Leaf curl (Fig. 6), Leaf Spot (Fig. 7.), Yellowish (Fig. 8), and White fly (Fig. 9), and healthy plants. Next, we split the dataset into training, validation, and test sets. Balance the dataset to prevent bias towards specific classes of chilli plant diseases. To increase the diversity of the dataset and improve the generalization

Fig. 5 Dense network



of the CNN model, we also obtained an external dataset of chilli plant images from a public source Kaggle Imagenet dataset.

In this study, the main task was to classify five different classes of chilli plants that include Leaf curl, White fly, Yellowish, leaf spot, and healthy plants. At first image dataset is preprocessed to normalize the images, adjust the image size, and remove any irrelevant information from the images. Then, the focus is on classifying four classes of chilli plant diseases that involves fine-tuning the pre-trained CNN models such as VGG16, VGG19, Inception V3, and Alex-net to the specific problem of chilli plant disease classification. Later the dataset was divided into two categories: healthy and unhealthy plants as shown in Figs. 10 and 11. We use binary classification approach to classify the plants as healthy or unhealthy based on the presence of diseases. The



Fig. 6 Images of leaf curl disease in dataset

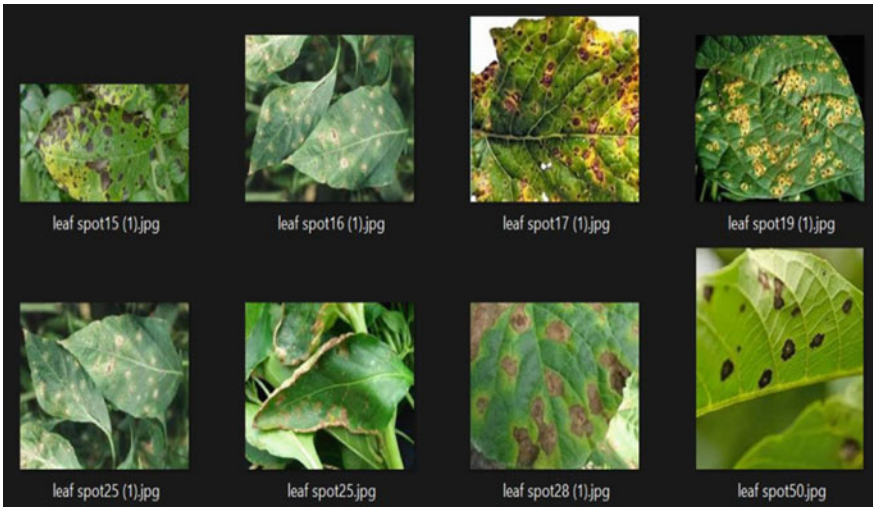


Fig. 7 Images of leaf spot disease in dataset



Fig. 8 Images of leaf yellowish disease in dataset

model takes into account various features such as shape, color, and size to accurately detect the disease. Finally, we try to induce some percentage of disease into the healthy plants and then perform binary classification again. This step simulates a more realistic scenario where not all unhealthy plants have the same level of illness, and some may have less severe symptoms.

The study uses various techniques such as data augmentation, transfer learning, and fine-tuning to improve the accuracy of the CNN models. The proposed model can

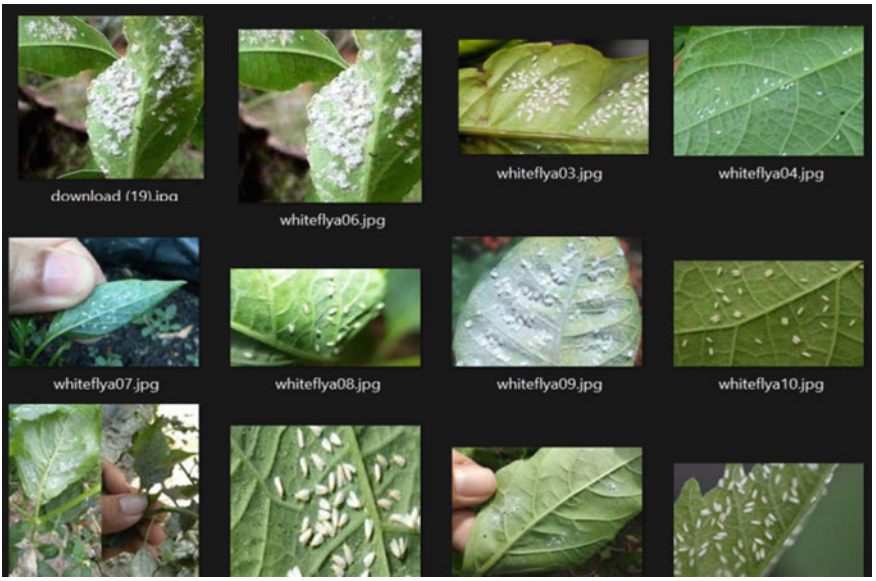


Fig. 9 Images of White fly disease in dataset



Fig. 10 Images of healthy class in dataset

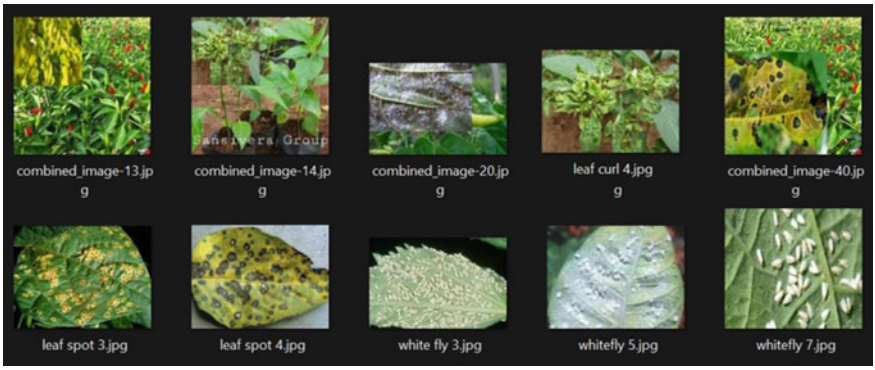


Fig. 11 Images of unhealthy class in dataset

accurately detect and classify chilli plant diseases and recommend appropriate treatment using deep learning. This can help in the early detection and timely treatment of chilli plant diseases, leading to better crop yields and reduced economic losses. The description of the dataset is given below and shown in Fig. 12.

Chilli: Five classes are available.

Classifications: Healthy, Leaf curl, Leaf spot, Whitefly, Yellowish. For the new dataset, we also gathered images of chilli plants. We also induced some disease in some of the healthy plants, dividing the dataset into Healthy and Unhealthy as shown in Fig. 13. The new dataset is based on the following characteristics: There are two classes. (Healthy-0 and Unhealthy-1) 70 images of trains and 30 images for validation/testing. There are a total of 100 images. Plant images that are 10% unhealthy,

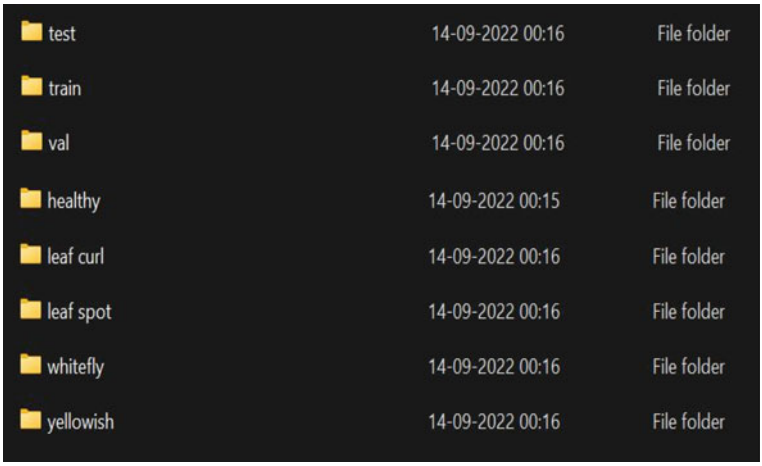


Fig. 12 Test train validation, leaf classes dataset folders

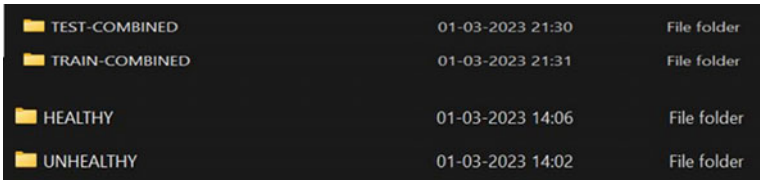


Fig. 13 Healthy, unhealthy, tests and train folders of new dataset

30% unhealthy and completely healthy are considered healthy. Plant images that are 50%, 70%, or 100% unhealthy are considered unhealthy.

2.1 Data Preprocessing

Data preprocessing is the process of transforming, cleaning, and enriching raw data for analysis, aiming to improve its quality, remove inconsistencies, and enhance performance. Data preprocessing is crucial for accurate and reliable results, as a well-prepared dataset enhances machine learning models, reduces over-fitting or under-fitting risks, and leads to better insights and decisions. The 128 by 128 pixel cropped photos of chilli illness and pest utilized in this investigation. We pre-processed the ImageNet dataset in order to use the pre-trained model on it similar to those we used for teaching the trainers about disease and pest dataset for ImageNet. The photographs of the diseases and pests were first resized, Pixels of 224 $\times$ 224 similar to the [9], were made more uniform by determining the average value of every dataset channel. Next, we deducted the average computed value for every input image. The standard deviation was a 1D array representing the overall RGB pixels' average values. Images from ImageNet are 103.939, 116.779, and 123.68.

2.2 Feature Extraction

Feature extraction in CNN models using image datasets involves learning a hierarchical representation of the input image through a series of convolution, pooling, and fully connected layers. This process allows the model to extract useful features from the input image and make accurate predictions. Feature extraction and fine-tuning are the two primary categories of transfer learning strategies [10]. A representation of the previously trained network is used to extract characteristics from fresh leaf samples. The CNN model uses fully linked layers for classification and successive convolution and pooling layers for those tasks. The output value of a particular layer of the pre-trained model is regarded as a feature in the feature extraction process, and the feature retrieved from the deep-learning model is referred to as a deep feature.

The process of adapting architecture for a new use based on a previously trained model and updating training from pre trained weights is known as fine-tuning.

3 Measurement of Performance

To get a sense of how our pretrained model would perform on new, unforeseen data and to keep track of which of our approaches are over fitting, we run all of our experiments across a variety of train-test set splits, particularly 80–20 (80% of the entire dataset used for training, and 20% for testing), and 60–40 (60% of the entire dataset used for training, and 40% for testing). We have the mappings for such scenarios for 20,212 photos out of the 40,605 images. It should be noted that the dataset typically comprises numerous photographs of the same leaf (shot from various orientations). During every test-train split, we additionally make sure that all photos of the same leaf are either placed in the training set or the testing set. At regular intervals (at the end of each epoch), we compute the mean overall accuracy for each experiment during the course of the training session. For the purpose of comparing the outcomes of all of the various experimental settings, we utilize the accuracy score only in this paper. We assess convolution neural networks' suitability for the aforementioned categorization issue. We concentrate on two well-known architectures, Inception V3 and VGG-19. In one scenario, we train the model from scratch, and in the other, we adjust previously trained models (trained on the ImageNet dataset) using transfer learning to compare the performance of both of these architectures on the available datasets. We do not restrict the learning of any of the layers when training the model. In other words, the initial state of the weights of a few layers is the main distinction between these two learning approaches (transfer vs. training from scratch), allowing the transfer learning strategy to take advantage of the substantial amount of visual experience.

4 Result Analysis

The findings of the models are shown below. These data help us to understand the models' prediction accuracy and how well a model can classify diseases into their appropriate classes.

CNN Results: The primary building block for CNN is the convolution layers. The raw data that is supplied into the CNN are represented by the input layer. The input layer for image data is an image tensor having dimensions (height, width, and channels) that correspond to the pixels in an image. They are made up of several learnable filters, sometimes known as kernels that scan the input for patterns and extract features.

From Table 1, it is evident that the model's accuracy significantly improves from step 1 to step 3 (from 69 to 93%), indicating that the model is learning and performing

better as it undergoes more training or refinement. However, there is a drop in accuracy from step 3 to step 4, which might indicate over-fitting or other issues.

VGG-16 Results: Visual Geometry Group (VGG) created the 16-layer deep convolutional neural network known as VGG16. Multiple convolutional layers are included, followed by fully connected layers. Input is 224 × 224 × 3 (224 × 224 pixel RGB Image), 13 convolution layer 3 × 3 filters, followed by ReLU activation techniques. Max-pooling layers with 2 × 2 filters and stride 2 are applied after several convolutional layers. The final three layers have 4096 neurons each and are fully connected, followed by ReLU activations. With the same number of neurons as classes in the classification problem, the final layer is a Softmax layer.

From Table 2, we can observe that the accuracy consistently improves from step 1 to step 3, with a significant increase from 73 to 94%. This indicates that the model is learning and performing better as it undergoes more training or optimization. However, there is a slight drop in accuracy from step 3 to step 4, which might indicate some over-fitting or other issues. The VGG16 model seems to be performing quite well, especially with the highest accuracy achieved at step 3 (94%).

Table 1 Accuracy report of CNN

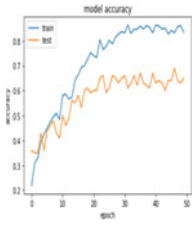
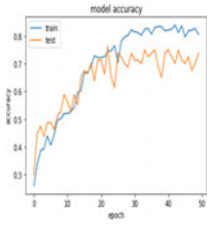
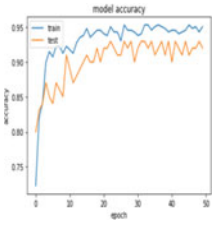
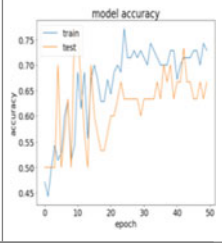
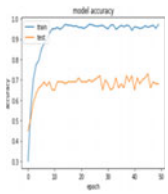
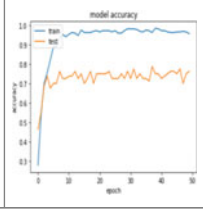
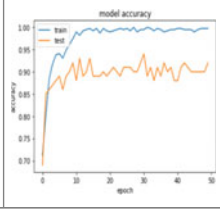
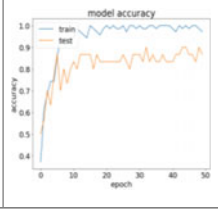
Accuracy step-1	Accuracy step-2	Accuracy step-3	Accuracy step-4
69	76	93	76
			

Table 2 Accuracy report of VGG-16

Accuracy step-1	Accuracy step-2	Accuracy step-3	Accuracy step-4
73	78	94	90
			

VGG19 Results: The extension of VG16 is VGG19 having similar architecture to VGG16 but includes additional 3, that is, 19 convolutional layers. VGG19 have 16 convolutional layers with 3 × 3 filters, followed by ReLU activation functions. In Table 3, we can observe that the accuracy shows some fluctuations during the different steps, but there is an overall improvement from step 1 to step 3. The model achieves the highest accuracy at step 3 (91%), indicating that it is learning and performing better as it undergoes more training or refinement. However, there is a decrease in accuracy from step 3 to step 4, which may indicate some over-fitting or other issues. Overall, the VGG19 model performed reasonably well, especially with the highest accuracy achieved at step 3 (91%).

Inception V3 Results: The Google-developed Inception V3, it makes use of the idea of "Inception modules," which enable the use of various-sized filters within a single layer to record data at various scales. The input is 299 × 299 × 3 (RGB image with a size of 299 × 299 pixels). It has convolution layer with different inception models with different filter sizes (1 × 1, 3 × 3, 5 × 5) for the extraction of features. To avoid overfitting, the last layers have dropout regularization and are fully coupled, and the output layer is SoftMax with number of classes in classification tasks equal to neurons.

From Table 4, we can observe that the accuracy consistently improves from step 1 to step 3, with a substantial increase from 78 to 96%. This indicates that the model is learning and performing significantly better as it undergoes more training or optimization. There is a slight decrease in accuracy from step 3 to step 4, which might indicate some over-fitting or other issues. The Inception V3 model performed very well, especially with the highest accuracy achieved at step 3 (96%). With further fine-tuning and experimentation, it might be possible to improve the model's performance.

Alex-Net Results: Alex, Ilya and Geoffrey created the ground-breaking deep convolutional neural network known as Alex-Net. As a result of its victory in the 2012 ImageNet Large-Scale Visual Recognition Challenge (ILSVRC), computer vision advanced significantly. The input size is 227 × 227 × 3 (RGB image with a size of 227 × 227 pixels). It consists of five convolution layers with different filter sizes 11 × 11, 5 × 5, 3 × 3) followed by ReLU activation. The accuracy shows fluctuations

Table 3 Accuracy report of VGG19

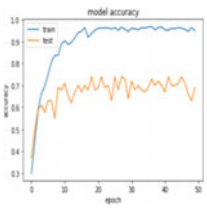
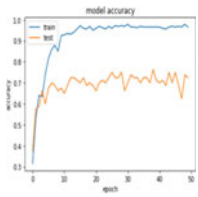
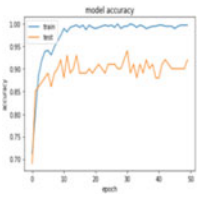
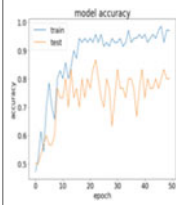
Accuracy step-1	Accuracy step-2	Accuracy step-3	Accuracy step-4
74	76	91	83
			

Table 4 Accuracy report of Inception V3

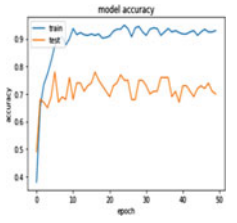
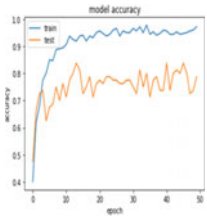
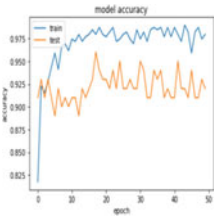
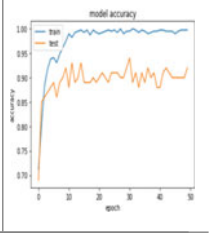
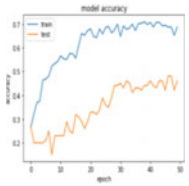
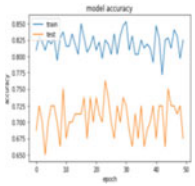
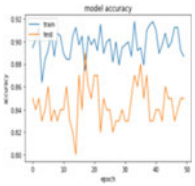
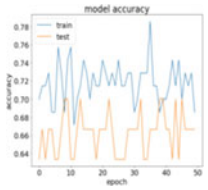
Accuracy step-1	Accuracy step-2	Accuracy step-3	Accuracy step-4
78	83	96	93
			

Table 5 Accuracy report of Alex-Net

Accuracy step-1	Accuracy step-2	Accuracy step-3	Accuracy step-4
48	76	89	70
			

during the different steps, and there is a significant improvement from step 1 to step 3, with accuracy increasing from 48 to 89%. This indicates that the model is learning and performing better as it undergoes more training or optimization. However, there is a decrease in accuracy from step 3 to step 4, dropping from 89 to 70%. This drop in accuracy might indicate some overfitting or other issues in the model. The Alex-Net model is making progress in terms of accuracy from step 1 to step 3. When compared to other steps, the Alex-Net model performed best in step 3 as shown in Table 5.

4.1 Model Comparison

The models have been compared accordingly in the below table where the Model is specified in the first column and the accuracies of each model, that is, CNN, VGG16, VGG19, Inception V3, and Alex-Net models are specified in the next columns for the four steps. This comparison helps achieve the overall best model based on the accuracy of the model in each step. Table 6 shows the accuracy of different models used in the different steps of detecting and classifying chilli plant diseases. The table lists four steps, from step 1 to step 4, with each step having a different objective and task. In step 1, we classified five classes of chilli plant diseases, including Leaf

curl, White fly, Yellowish, leaf spot, and healthy. In step 2, we classified four classes of chilli plant diseases, excluding healthy ones. In step 3, we performed binary classification by dividing the dataset into healthy and unhealthy classes, with all diseased classes labelled as unhealthy. In step 4, we induced some percentage of disease into the healthy plants and performed binary classification.

Table 6 shows the accuracy of different models in each of these steps. Each model has its architecture, which affects its performance in different tasks. Looking at the table, we can see that the accuracy of all models varies across different steps. For example, the CNN model has an accuracy of 69% in step 1, which is the lowest accuracy among all models. However, its accuracy increases in steps 3 and 4, where it achieves 93% and 76% accuracy, respectively. On the other hand, the Inception V3 model has the highest accuracy in all steps, with an accuracy of 78%, 83%, 96%, and 93% in steps 1, 2, 3, and 4, respectively. This indicates that Inception V3 is a powerful model for detecting and classifying chilli plant diseases. Overall, the table provides a useful comparison of different models' accuracy in each step of the study, helping to identify the best-performing models for each task.

Table 7 the performance of different models in detecting and classifying chilli plant diseases when some percentage of disease is induced into healthy plants in step 4. The table includes five models, which are Inception V3, VGG16, VGG19, CNN, and Alex-Net. For each model, the table provides the accuracy, probability, and number of healthy and unhealthy plants that were correctly detected and classified. The accuracy column shows the overall accuracy of each model in detecting and classifying chilli plant diseases when some percentage of disease is induced into healthy plants. The values range from 70 to 93%, with Inception V3 achieving the highest accuracy of 93%, and Alex-Net achieving the lowest accuracy of 70%. The probability column shows the probability of each plant being healthy or unhealthy. For example, a probability value of 0.7 indicates that there is a 70% chance that the plant is healthy and a 30% chance that it is unhealthy. The next two columns show the actual and detected disease labels for each plant. The actual disease labels are based on whether the plant was healthy or unhealthy before the disease was induced, and the detected disease labels are based on the model's classification results. Looking at the table, we can see that Inception V3, VGG16, and CNN models have correctly detected and classified all healthy and unhealthy plants. However, the VGG19 model incorrectly classified one healthy plant as unhealthy. The Alex-Net model correctly

Table 6 Model comparison

Model	Accuracy in step-1	Accuracy in step-2	Accuracy in step-3	Accuracy in step-4
CNN	69	76	93	76
VGG16	73	78	94	90
VGG19	74	76	91	83
Inception V3	78	83	96	93
Alex-Net	48	76	89	70

Table 7 Performance of models in detecting and classifying chilli diseases with induced disease in healthy plants

Algorithm	Accuracy	Probability		Disease	
		Healthy	Unhealthy	Actual	Detected
InceptionV3	93	9.99881387	1.18544674	Healthy	Healthy
		0	1.0000000	Unhealthy	Unhealthy
VGG16	90	0.83606935	0.16393062	Healthy	Healthy
		0.01924389	0.9807561	Unhealthy	Unhealthy
VGG19	83	0.47754118	0.52245885	Healthy	Unhealthy
		0.20832987	0.79167014	Unhealthy	Unhealthy
CNN	76	0.7034686	0.29653138	Healthy	Healthy
		0.45679343	0.5432065	Unhealthy	Unhealthy
Alex-Net	70	0.5012679	0.4987321	Healthy	Healthy
		0.50262487	0.4973751	Unhealthy	Healthy

classified all unhealthy plants but incorrectly classified one healthy plant as unhealthy. Overall, the table shows that the models' performance varies in detecting and classifying chilli plant diseases when some percentage of disease is induced into healthy plants. It also shows that Inception V3 and VGG16 are the most reliable models in detecting and classifying diseases accurately.

4.2 Web Application

Figure 14 shows the block diagram of the concept, the flow of the project starts from collecting the dataset, which is done from two sources—the Kaggle dataset and a mobile camera. The dataset is then preprocessed and augmented for improving the accuracy of the model. The preprocessed dataset is then fed into a CNN, which is the core of the project. The CNN has two parts—the feature extraction part that uses convolutional layers to extract features from the input images, and the disease classification part that uses dense layers to classify the disease into one of the categories. Once the disease is identified, the model provides a treatment recommendation to the user based on the detected disease. This recommendation is made using pesticides and insecticides, which are suggested, based on the type of disease identified. Overall, the study uses CNN to detect and classify the disease in chilli plants and recommends appropriate treatments for the same.

Figures 15, 16 and 17 of Web Application shows the user interface that allows the user to upload or take a picture of a chilli plant. The image is then preprocessed, and the features are extracted using a CNN. The extracted features are then classified by the CNN and the disease is predicted. The appropriate treatment is suggested based

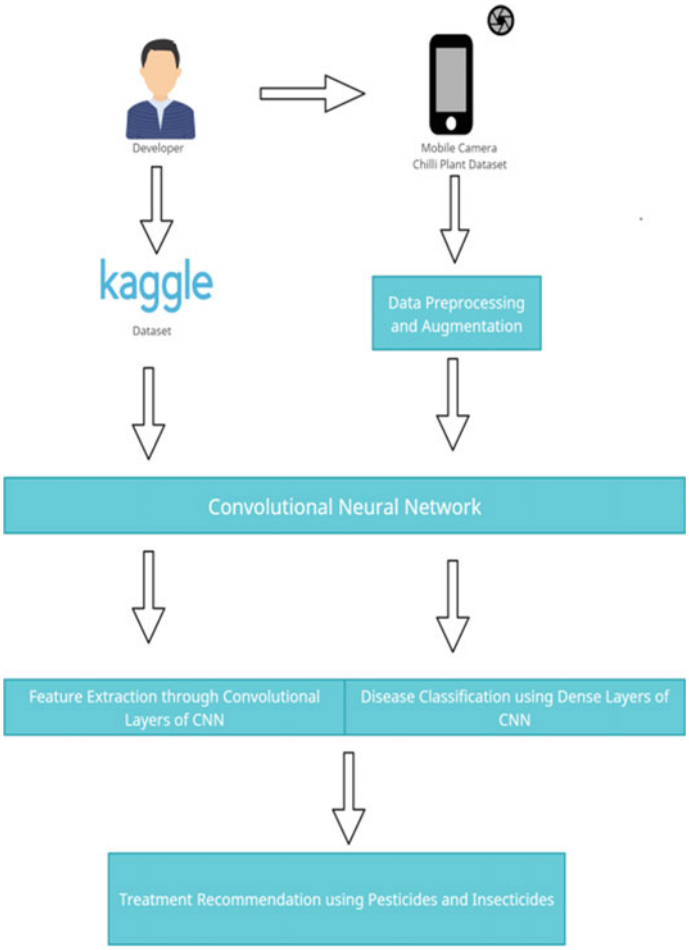


Fig. 14 Block diagram of proposed model

on the predicted disease. Finally, the output is displayed through a web application for the user to see.

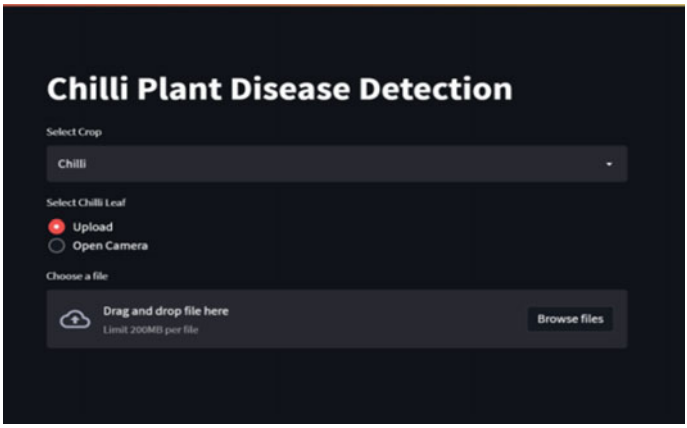


Fig. 15 User interface

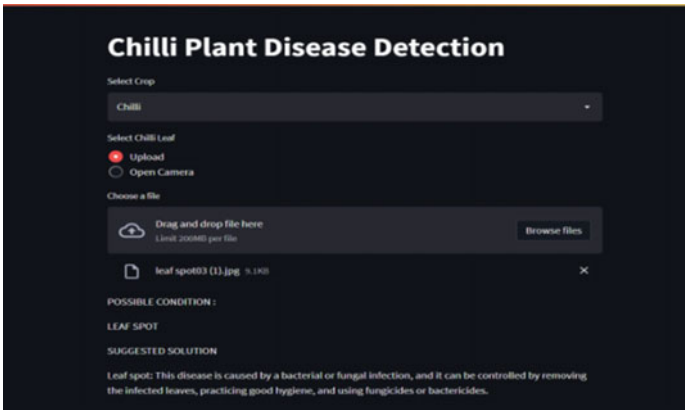


Fig. 16 Uploading an image and getting the result

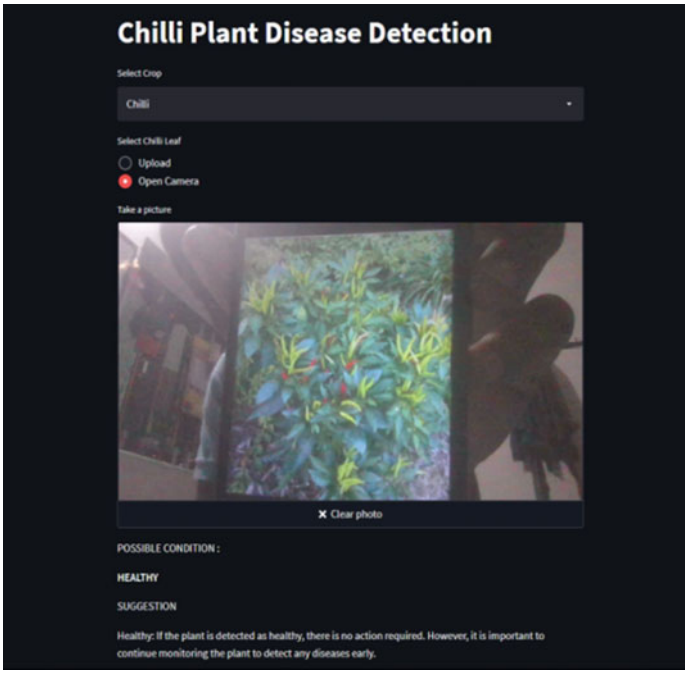


Fig. 17 Taking photo of an image and getting the result

5 Conclusion

Finally, for the purpose of identifying and categorizing diseases of chilli plants, this study created and assessed CNN models as well as transfer learning models including VGG16, VGG19, Inception V3, and Alex-Net. Five classes of chilli plant illnesses, including leaf curl, white fly, yellowish, leaf spot, and healthy, were used to train and test the models. The Inception V3 model had the best accuracy overall, according to the data, and the CNN and VGG16 models also performed well. Furthermore, even when a small amount of illness was introduced into healthy plants, the suggested models were able to identify and categorize chilli plant infections with excellent accuracy. In order to treat the disease, the proposed models can be utilized to suggest relevant pesticides or insecticides. However, there are a few restrictions at this point that must be resolved in upcoming work. Initially, when examined on photos taken under situations that are different from the ones in the photographs when utilized for training, the model's accuracy is significantly decreased, to a little over 10%. Each of this experiment lasts for a total of 30 epochs, where an epoch is the number of training iterations during which the specific neural network has successfully traversed the whole training set. The significance of this accuracy is superior to the one based on classes being randomly selected, although a more varied collection of training data is to increase the accuracy. Our most recent findings show that additional (and variable)

data alone will be adequate to significantly improve the accuracy and related data collection activities are in progress. The future work will be to construct a hybrid model and increase the size of the developed dataset of chilli images with some additional diseases.

References

1. Tai AP, Martin MV, Heald CL (2014) Threat to future global food security from climate change and ozone air pollution. Nat Clim Chang 4:817–821. https://doi.org/10.1038/nclimate2317
2. Bhookya NN, Malmathanraj R, Palanisamy P (2020) Yield estimation of chilli crop using image processing techniques. In: 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, pp 200–204. https://doi.org/10.1109/ICACCS48705.2020.9074257
3. Shrestha G, Deepsikha, Das M, Dey N (2020) Plant disease detection using CNN. In: 2020 IEEE Applied Signal Processing Conference (ASPCON), pp 109–113. https://doi.org/10.1109/ASPCON49795.2020.9276722
4. Sladojevic S, Arsenovic M, Anderla A, Culibrk D, Stefanovic D (2016) Deep neural networks based recognition of plant diseases by leaf image classification. Comput Intell Neurosci 2016:3289801. https://doi.org/10.1155/2016/3289801
5. Pattnaik G, Shrivastava VK, Parvathi K (2020) Transfer learning-based framework for classification of pest in tomato plants. Appl Artif Intell 34:981–993. https://doi.org/10.1080/08839514.2020.1792034
6. Aravind KR, Raja P, Aniirudh R, Mukesh KV, Ashiwin R, Vikas G (2019) Grape crop disease classification using transfer learning approach. Lect Notes Comput Vis Biomech 30:1623–1633. https://doi.org/10.1007/978-3-030-00665-5_150
7. Krizhevsky A, Sutskever I, Hinton GE (2017) ImageNet classification with deep convolutional neural networks. Commun ACM 60:84–90
8. Xie C, Shao Y, Li X, He Y (2015) Detection of early blight and late blight diseases on tomato leaves using hyperspectral imaging. Sci Rep 5:16564
9. Zhuang F, Qi Z, Duan K, Xi D, Zhu Y, Zhu H et al (2021) A comprehensive survey on transfer learning. Proc IEEE 109:43–76. https://doi.org/10.1109/JPROC.2020.3004555
10. Kandel I, Castelli M (2020) Transfer learning with convolutional neural networks for diabetic retinopathy image classification: a review. Appl Sci 10:2021. https://doi.org/10.3390/app10062021
11. Ullah Z, Lodhi BA, Hur J (2020) Detection and identification of demagnetization and bearing faults in PMSM using transfer learning-based VGG. Energies 13:3834. https://doi.org/10.3390/en13153834
12. Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S et al (2015) ImageNet large scale visual recognition challenge. Int J Comput Vis 115:211–252. https://doi.org/10.1007/s11263-015-0816-y
13. LeCun Y, Bengio Y, Hinton G (2015) Deep learning. Nature 521:436–444. https://doi.org/10.1038/nature14539
14. Canziani A, Paszke A, Culurciello E (2016) An analysis of deep neural network models for practical applications. Preprint at http://arxiv.org/abs/1605.07678. Accessed 25 Nov 2021
15. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. Preprint at http://arxiv.org/abs/1409.1556

Sign Language Recognition Using Long Short-Term Memory Deep Learning Model



Prabhat Mali, Aman Shakya, and Sanjeeb Prasad Panday

Abstract This paper aims to present an easy way of communication for speech impaired and hearing impaired people. This paper focuses on the use of neural network techniques, specifically MediaPipe Holistic and the Long Short-Term Memory (LSTM) module, to recognize sign language for individuals with disabilities. The utilization of MediaPipe Holistic, which integrates pose, hand, and face keypoints with precise levels, was used due to its low latency and high tracking accuracy in real-world scenarios. This paper deals with commonly used words of American Standard Sign Language. The result obtained shows that the proposed model can detect American Standard Sign Language with accuracy of 98.50 percentage.

Keywords Sign language · Keypoints · MediaPipe holistic · Long short-term memory

1 Introduction

Recently, there has been a sharp rise in the number of individuals who are speech- or hearing-impaired due to factors such as birth defects, oral health issues, and accidents. This affects around 7–9 million people worldwide. When these individuals communicate with those who are not speech- or hearing-impaired, it can be challenging for the latter group to understand. To address this issue, the use of gesture recognition systems has become increasingly common to bridge the communication gap between different groups. Sign language recognition is a well-studied field and

P. Mali · A. Shakya (✉) · S. P. Panday
Department of Electronics and Computer Engineering, Pulchowk Campus, Institute of Engineering, Tribhuvan University, Lalitpur, Bagmati, Nepal
e-mail: aman.shakya@ioe.edu.np

P. Mali
e-mail: 078msice010.prabhat@pcampus.edu.np

S. P. Panday
e-mail: sanjeeb@ioe.edu.np

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2023

S. Shakya et al. (eds.), *Fourth International Conference on Image Processing and Capsule Networks*, Lecture Notes in Networks and Systems 798,
https://doi.org/10.1007/978-981-99-7093-3_46

has made significant progress in recent years, with various techniques being explored as a way to facilitate communication. These techniques are generally categorized as either vision-based or sensor-based. Vision-based methods offer user-independent feature extraction.

This paper uses action detection system to identify American Standard Sign Language (ASSL) used by speech-impaired and hearing-impaired people to communicate. The aim of this paper is to utilize computer vision to accurately identify sign language in real-time. This will facilitate communication between people who are speech- and hearing-impaired by allowing them to use their natural hand gestures. This system will dynamically translate sign language, thereby enhancing communication effectiveness and efficiency between individuals who are hearing or speech impaired and those who are not.

Individuals who have difficulties with speech or hearing rely primarily on sign language as a mode of communication. However, it is not understandable by people who do not have speech or hearing impairments, leading to a significant communication barrier. Additionally, sign language can be difficult to learn due to its distinct sentence structure and grammar. To overcome these challenges, there is a need to create a system that can translate sign language. Several attempts have been made in this field, but the present systems are lacking in terms of efficient data collection and faster sign recognition. This paper is an attempt to contribute a step in this area.

2 Related Works

According to a report by the World Federation of the Deaf, the global population of people who are speech- or hearing-impaired is estimated to be between 7 and 9 million. These individuals rely on sign language as their primary mode of communication; however, most people without impairments do not understand the gestures used in sign language, making communication difficult. The objective of this paper is to help individuals with speech disabilities have normal conversations in their daily lives. Sign language varies from region to region, making it challenging to develop a universal system. Over the years, numerous attempts have been made to reduce the communication barrier between speech-impaired and hearing-impaired communities and those without impairments. Most of these attempts involved converting sign language into audible signals. Some of the past attempts to develop a sign language recognition system using various technologies are disscused here.

Authors of paper [1] focus on neural network methods, such as MediaPipe Holistic and the LSTM, to identify sign language used by individuals with impairment. The paper utilized MediaPipe Holistic, which combines hand, pose, and face keypoints. The primary objective of their paper was to demonstrate the use of Human Activity Recognition (HAR) algorithm, which is based on deep learning architecture, for recognizing human actions and classifying them into seven different categories. Convolution Neural Network (CNN) and Long Short-Term Memory (LSTM) were used to analyze spatial and temporal data gathered from a Microsoft Kinect camera.

The paper [2] designed and implemented a real-time system for recognizing hand gestures. This system identifies hand gestures, like peak calculation and angle calculation, and then translates the gesture images into text. To implement this system, sign language hand gesture dataset was used. The proposed system had four main modules such as preprocessing, segmentation, feature extraction, and classification. In the preprocessing module, the input image was converted into a grayscale image, noise removal, normalization, and rescale the image. The image is preprocessed and then segmented using an automatic threshold algorithm. After that, feature extraction is conducted using Scale-Invariant Feature Transform (SIFT) and Speeded-Up Robust Features' (SURFs) descriptors. The features collected from this process are then trained using an Excel file. The Support Vector Machines' (SVMs) algorithm is utilized to classify the input.

Authors of paper [3] developed a program that utilizes machine learning methods to recognize Kazakh dactylic sign language. Their dataset consists of 5000 images of each gesture, and gesture recognition algorithms including Support Vector Machine, Random Forest, and Extreme Gradient Boosting were applied. Additionally, two data types were merged into a single database. The findings from their work revealed that Extreme Gradient Boosting and Support Vector Machine have better real-time efficiency, while Random Forest algorithm has high recognition efficiency.

A vision-based hand gesture recognition system is described in paper [4], which considers various techniques for hand tracking, feature extraction, segmentation, and classification. Their paper involves capturing images using a webcam and processing them using the OTSU method, and classification was done using a linear classification method. In their paper, Chhaya Narvekar and her team stored the captured gestures in folders that contained 120 images of a gesture. The image was captured as histogram.

The authors of paper [5] used images of American Standard Sign Language (ASSL) taken in various environmental conditions as a dataset to recognize and classify hand gestures accurately. The authors applied various preprocessing techniques such as Histogram of Gradients, Principal Component Analysis, and Local Binary Patterns and created a novel model using canny edge detection, ORB, and the bag-of-words technique. The data that had been prepared in advance were subjected to multiple classifiers such as Random Forests, Support Vector Machines, Naive Bayes, Logistic Regression, K-Nearest Neighbors, and Multilayer Perceptron, in order to achieve optimal outcomes.

The authors of paper [7] described the results of using the Long Short-Term Memory (LSTM) model, that improved the machine translation of Google Translate. The article [8] introduces an efficient structure for addressing the static gesture recognition challenge using data sourced from web cameras and the Kinect depth sensor. The paper [9] outlines an approach to identify American Sign Language gestures, employing principal component analysis to reduce the resemblance between different gesture categories.

In [10], the Support Vector Machine (SVM) classifier was employed, and the authors diverged from conventional techniques by opting not to utilize hand markers like gloves for gesture identification. In the study [11], a depth map was generated through the application of the Sum of Absolute Differences (SAD) algorithm,

aligning the left and right images. The contours were determined using the Theo Pavlidis Algorithm, which exclusively examines boundary pixels, leading to reduced computational expenses. In paper [12], a method involving orientation histograms was suggested for the identification of stationary hand gestures.

Paper [13] employed decision trees, while paper [14] utilized Hidden Markov Models to discern the dynamic aspects of hand gestures. In the study [15], real-time interpretation of a user's gestures is achieved through hand segmentation, extracting binary hand blobs. The shape of these blobs is then captured using Fourier descriptors. These Fourier descriptors are fed into radial-basis function (RBF) networks for the purpose of posture classification. In [16], a technique is introduced that involves utilizing the angle and peak calculation approach to extract hand gesture features. This process is carried out using MATLAB, and subsequently, the identified gesture is transformed into speech using MATLAB's built-in command.

3 Methodology

Figure 1 shows the general flow of the developed system. Continuous frames of video are passed form MediaPipe Holistic which detects the keypoints present in the video frame, and keypoints are then extracted to a Numpy array which are fed to the Long Short-Term Memory (LSTM) model to make the prediction. Figure 2 shows the LSTM model architecture, which consists of input, LSTM layers, dense layers, and output. The input data are a Numpy array of shape (30,1662). The model consists of three layers of LSTM layer with 64, 128, and 64 units, respectively, which is followed by three dense layers with 64, 32, and 10 units (i.e., number of actions the model is trained with), respectively. The output unit gives the predicted gesture in the form of text. LSTM models are used in this paper because it can handle sequential data effectively, capture long-term dependencies, and have memory cells to store relevant information. Dense layers are responsible for learning complex relationships and patterns in the data. It applies a linear transformation followed by a nonlinear activation function to the input data.

3.1 Software and Hardware Platforms

The software language and platforms used for this work are Python, TensorFlow, OpenCV-Python, MediaPipe, and Sklearn.

- The hardware used for the work are as following:
  - HP Laptop PC 14-d2000.
  - Processor: 11th Gen Intel(R) Core(TM) i5-1135G7 @ 2.40 GHz.
  - Installed RAM: 8.00 GB.

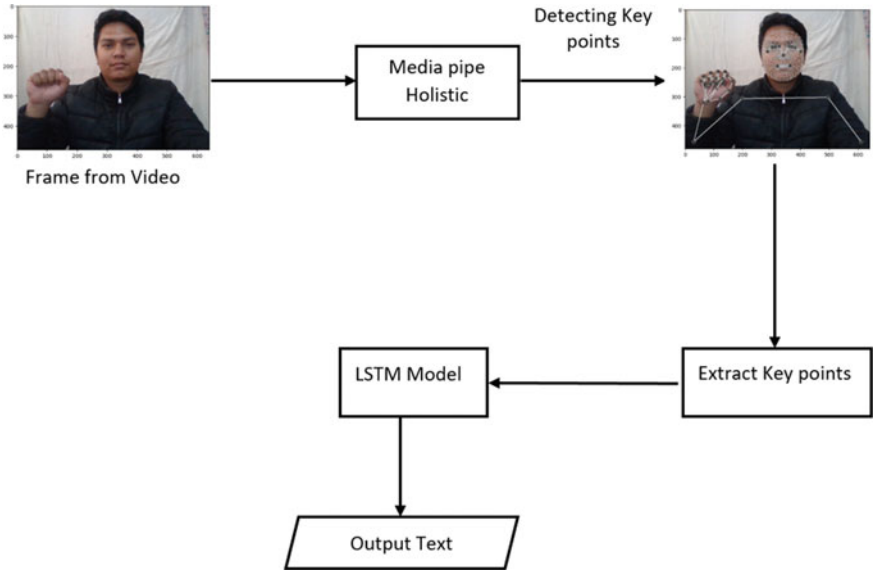


Fig. 1 System flowchart

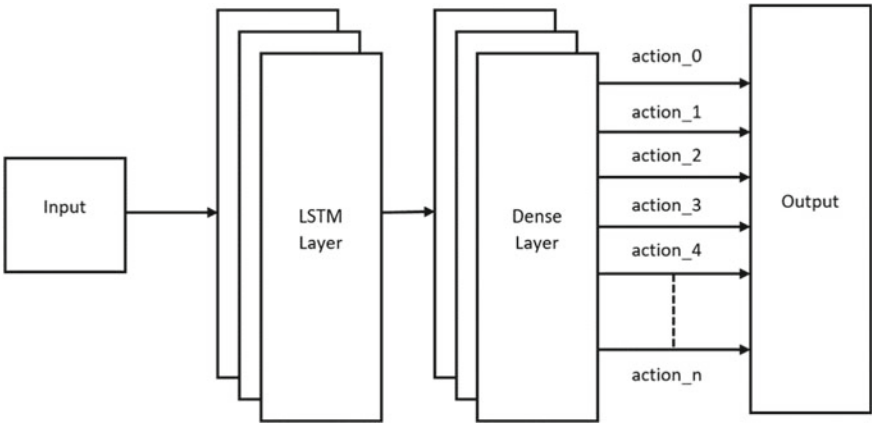


Fig. 2 LSTM model architecture

- HP TrueVision HD Camera.
  - Megapixel: 0.9 MP, 30 fps.
  - Hardware ID: VID 30C9PID 0035MI 00.

3.2 Creating Dataset

In this work, live video footage from a camera is constantly streamed, and each frame that contains a gesture or movement within the frame is saved in a directory. For the creation of the dataset, each captured frame is passed through MediaPipe Holistic which marks the keypoints of hand, pose, and face. Each hand contains of 21 keypoints each, pose contains 33 keypoints, and face contains 468 keypoints, respectively. These keypoints are concatenated to a Numpy array to form a dataset. If any of the keypoints are not visible inside the video frame, then its value is filled with zero. The collection of characters is composed of 100 video sequences, each representing a specific sign language symbol. Each video sequence consists of 30 frames that capture key moments and are stored as Numpy arrays. The data are collected for ten commonly used gestures. So, the dataset consists of Numpy array of keypoint values extracted using MediaPipe Holistic from 100 video sequences each containing 30 frames with each frame consisting of 1662 keypoints' values (Table 1).

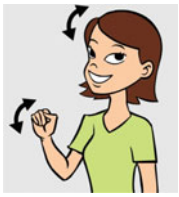
3.3 Training

The training for model development was performed using TensorFlow, a machine learning platform implemented in Python. In order to employ transfer learning, the datasets and labeled files had to be transformed into a format that TensorFlow could handle. The transformation was done by creating tfrecord files from the folders that contained the training and test data. To begin the training process using transfer learning, pre-existing detection models had to be downloaded. The configuration files needed to be modified to match the number of classes in the dataset. Two hundred training steps were required to achieve desired accuracy. TensorFlow and Keras were combined to produce a LSTM model that can predict sign language gestures on the screen. The model consists of LSTM layer and dense layers as mentioned previously.

3.4 Predicting the Gesture

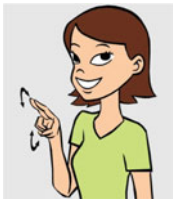
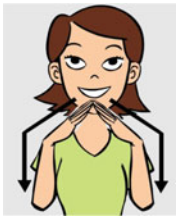
Initially, a frame containing a sign language gesture is captured by the camera. Then, the keypoints for the hand, pose, and face are detected using MediaPipe Holistic. The keypoints captured are then converted into a flattened Numpy array and are inputted into the previously trained LSTM model followed by dense layers for prediction. The final dense layer consists of ten units which represents the number of gesture the model is trained for. This layer utilizes Softmax function as an activation function, i.e., it gives a probability score for each action defined, and when the score reaches

Table 1 List of gestures

S. No.	Gesture	Word
1		Hello
2		Goodbye
3		Please
4		Thank you
5		Yes

(continued)

Table 1 (continued)

S. No.	Gesture	Word
6		No
7		Help
8		House
9		Hungry
10		Sorry

the threshold value, then the respective output text is displayed over the video frame as shown in Figs. 3 and 4.

The Softmax equation can be given as:

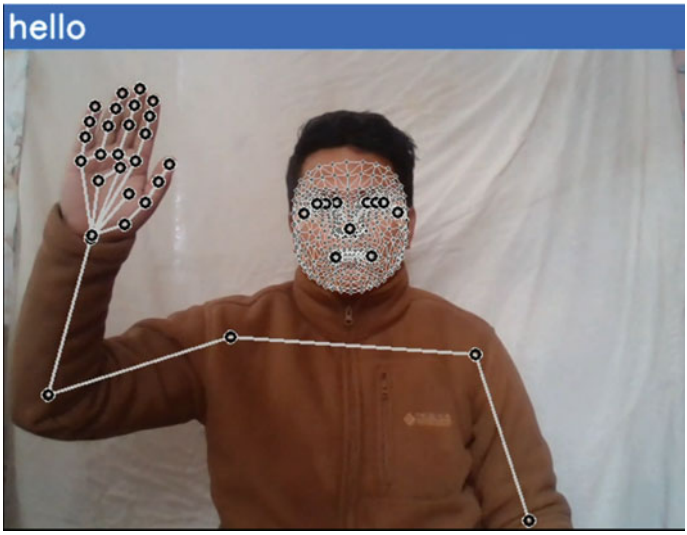


Fig. 3 Output for hello

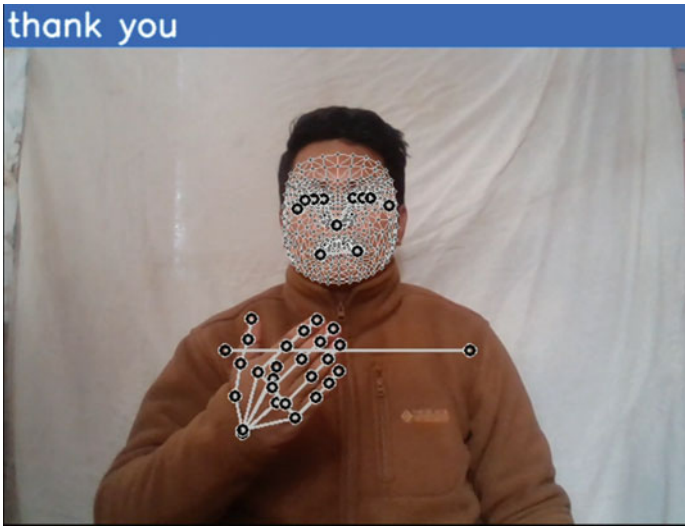


Fig. 4 Output for thank you

$$f_{yi}(z) = \frac{e^{zj}}{\sum_{k=1}^{C} e^{z_k}} \tag{1}$$

The Z here represents the data from the output layer's neurons, where C = total number of classes. The exponential is used to depict nonlinear functions. The sum of exponential values is used to standardize and transform these data into probabilities [6].

This paper focused on sign language recognition that can identify American Sign Language gestures. The system can also be easily expanded to incorporate personalized symbols and additional hand gestures. This paper was constructed using OpenCV, MediaPipe, TensorFlow, and Keras Python modules. OpenCV analyzes live video captured of frame size 640 * 480, 30 fps by a camera to detect the hand gestures of a person. MediaPipe Holistic is used to process the video frames and extract keypoints from the hands, pose, and face. The extracted points are then fed into the prediction algorithm, which makes the prediction. The technology recognizes the hand signal in real-time, and the resulting sign is displayed.

4 Result and Analysis

A total of 100 video sequences consisting of 30 frames each were used as a dataset for each gesture. For the collection of 100 video sequences, it took about 10 min, and for the training of the model through 200 epochs, it took 30 min. After collecting the data, it is split into two datasets: a training dataset and a testing dataset, with a ratio of 80:20. The accuracy and loss curve given below provides insight into the model's training progress and improvement across each epoch. This visual representation is useful for tracking how accuracy and loss metrics change over time, both for the training and validation datasets. The training was performed for two values of learning rate (LR) 0.01 and 0.001 (Figs. 5 and 6).

From the epoch categorical accuracy and loss curves, it is observed that the model was able to converse with the learning rate of 0.001 in 200 epochs.

A confusion matrix is a tabular representation that provides a summary of the classification model's performance on a given test dataset. It achieves this by comparing the predicted labels against the actual labels. This matrix is an excellent tool for assessing and interpreting the model's performance since it enables the visualization and analysis of the classification results. For the proposed model with learning rate of 0.001, the accuracy was calculated to be 98.50% (Table 2).

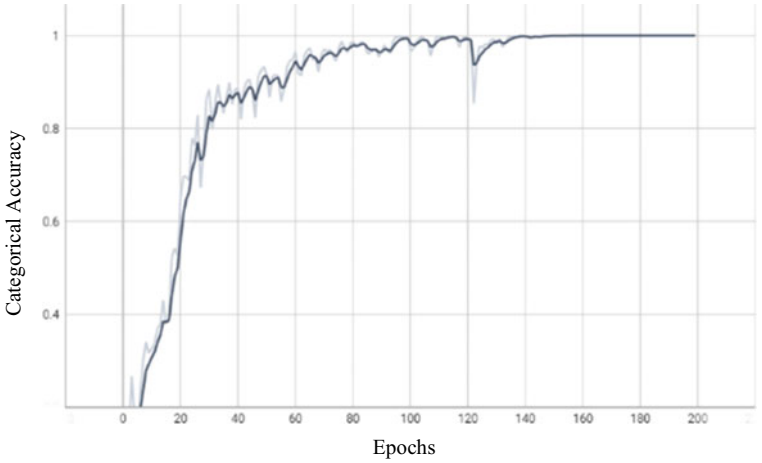


Fig. 5 Epoch categorical accuracy LR = 0.001

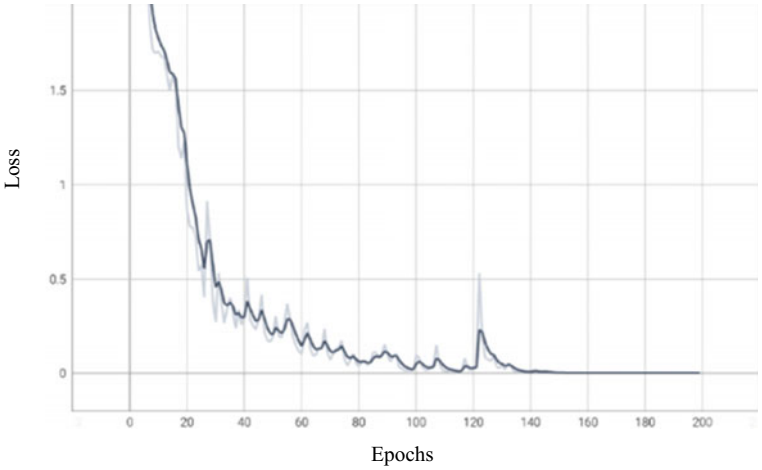


Fig. 6 Epoch loss LR = 0.001

5 Conclusion and Future Work

The system that has been proposed has shown significant success in recognizing American Standard Sign Language signs. Its high accuracy is attributed to the utilization of keypoints extracted from MediaPipe Holistic. The model's precision was evaluated to be 98.50%. To incorporate new signs into the model, one can add the corresponding gestures to the dataset and increase the number of prediction categories. The accuracy can be further increased by increasing the number of frame recorded for each gesture and increasing more video sequence data.

Table 2 Confusion matrix

P/A	Hello	Goodbye	Please	Thank you	Yes	No	House	Help	Hungry	Sorry
Hello	24	0	0	0	0	0	0	0	0	0
Goodbye	0	18	0	0	0	0	0	0	0	0
Please	0	0	21	0	0	0	0	0	1	0
Thank you	0	0	0	13	0	0	0	0	0	0
Yes	0	0	0	0	13	0	0	0	0	0
No	0	0	0	0	0	20	0	0	0	0
House	0	0	0	0	0	0	21	0	0	0
Help	0	0	0	0	0	0	0	20	0	0
Hungry	0	0	1	0	0	1	0	0	22	0
Sorry	0	0	0	0	0	0	0	0	0	25

The proposed system only contains ten commonly used ASSLs. So, the system needs to be trained for other gestures as well. The collection of data was carried out in a well-lit environment with a white plane background. The training and testing were carried out on a Laptop with Intel i5-1135G7 @ 2.40 GHz processor and 8 GB RAM. With decreases in these parameters, the performance of the system will decrease. Also, the current system is based on a computer system which is not feasible for daily use. So, a more portable system needs to be developed. Several future enhancements can be made to this proposed system. Currently, the dataset consists of ten gestures only in which each gesture contains 100 video sequences of 30 frames each. So, other remaining gestures of different words are needed to be added to the dataset. Also, by adding the number of video sequences and the number of frames, the accuracy of the system can be further increased. The proposed system makes use of 1162 keypoints' values for each frame, among which all the keypoints are not required for each gesture detection. So, in the future, the performance of the model can be improved by identifying unwanted keypoints, and by doing so, the complexity of the model will be reduced. Also in the future, this system can be extended to be used for different languages.

References

1. Bayegizova A, Murzabekova G, Ismailova A et al (2022) Effectiveness of the use of algorithms and methods of artificial technologies for sign language recognition for people with disabilities. Eastern-Euro J Enterprise Technol
2. Mercy PJ, Amsa Vani S (2021) Sign language identification using image processing. Int J Creative Res Thoughts
3. Kenshimov C, Buribayev Z (2021) Sign language dactyl recognition based on machine learning algorithms. Eastern-Euro J Enterprise Technol
4. Narvekar C, Mungekar S, Pandey A, Mahadik M (2020) Sign language to speech conversion using image processing and machine learning. Int J Technol Res Eng

5. Sharma A, Mittal A, Singh S, Awatramani V (2020) Hand gesture recognition using image processing and feature extraction techniques. In: International conference on smart sustainable intelligent computing and applications
6. Ligal S, Baral DS (2022) Nepali sign language gesture recognition using deep learning. In: 12th IOE graduate conference
7. Van Houdt G, Mosquera C, Nápoles G (2020) A review on the long short-term memory model. Artif Intell Rev
8. Artyukhin SG, Mestetskiy LM (2015) Dactyl alphabet gesture recognition in a video sequence using Microsoft Kinect. In: International archives of the photogrammetry, remote sensing and spatial information sciences
9. Zamani M, Kanan HR (2014) Saliency based alphabet and numbers of American sign language recognition using linear feature extraction. In: International conference on computer and knowledge engineering
10. Chen Y, Luo B, Chen Y-L, Liang G, Wu X (2015) A real-time dynamic hand gesture recognition system using Kinect sensor. In: IEEE international conference on robotics and biomimetics
11. Lee DH, Hong KS (2010) A hand gesture recognition system based on difference image entropy. In: International conference on advanced information management and service
12. Symeonidis K (2000) Gesture recognition using neural networks. In: Degree of master of science in multimedia signal processing communications, school of electronic and electrical engineering
13. Chen Y, Gao W, Ma J (2000) Hand gesture recognition based on decision tree. Institute of Computing Technology, Chinese Academy of Sciences
14. Marcel S, Bernier O, Viallet JE, Collobert D (2000) Hand gesture recognition using input–output hidden Markov models. France Telecom CNET
15. Ng CW, Ranganath S (2018) Static hand gesture recognition system using convolutional neural networks. Int J Sci Res Dev
16. Shinde SS, Autee RM, Bhosale VK (2013) Sign language recognition for deaf & dumb. Int J Adv Res Comput Sci Softw Eng

Explainable Artificial Intelligence and Deep Transfer Learning for Skin Disease Diagnosis



James Mayanja, Enoch Hall Asanda, Joshua Mwesigwa, Pius Tumwebaze, and Ggaliwango Marvin

Abstract Accurate diagnosis of skin diseases, including skin cancers, is challenging and often leads to misdiagnosis and incorrect treatments. Whereas misdiagnoses have far-reaching consequences for both the patients and the healthcare system, resulting in delayed interventions, disease progression, and compromised outcomes, diagnostic precision is extremely crucial for improved prognosis and disease management. Unfortunately, the complexity of skin diseases with their diverse symptoms and subjective interpretation adds to the diagnostic challenges faced by dermatologists. It complicates accurate identification of essential features to prevent unnecessary interventions and ensure appropriate treatment, saving healthcare resources. To address misdiagnosis, we apply intelligent approaches to dermatology particularly integration of artificial intelligence (AI) and machine learning (ML) using comprehensive datasets to enhance diagnostic accuracy. The AI models are trained on diverse skin disease data to provide reliable diagnostic support using explainable AI (XAI) techniques tailored for skin disease diagnosis. We provide transparency and interpretability to clinicians to enable them understand AI-driven diagnostics, promote trust and collaboration with AI diagnostic tools. This facilitates informed decision-making and improves the overall diagnostic process where accurate diagnosis is crucial given the prevalence of skin diseases. Therefore, this work presents advancements in ML and XAI to mitigate errors and improve dermatology diagnostic outcomes. This collaboration between dermatologists and technology experts

J. Mayanja · E. H. Asanda · J. Mwesigwa · P. Tumwebaze · G. Marvin (✉)
Makerere University, 7062 University Road, Kampala, Uganda
e-mail: ggaliwango.marvin@mak.ac.ug

J. Mayanja
e-mail: james.mayanja@students.mak.ac.ug

E. H. Asanda
e-mail: enoch.hall.asanda@students.mak.ac.ug

J. Mwesigwa
e-mail: joshua.mwesigwa@students.mak.ac.ug

P. Tumwebaze
e-mail: pius.tumwebaze@students.mak.ac.ug

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2023

S. Shakya et al. (eds.), *Fourth International Conference on Image Processing and Capsule Networks*, Lecture Notes in Networks and Systems 798,
https://doi.org/10.1007/978-981-99-7093-3_47

demonstrates the need and potential to enhance accuracy, efficiency, and effectiveness in skin disease diagnosis, leading to better healthcare delivery and patient well-being.

Keywords Deep transfer learning (DTL) · Artificial intelligence (AI) · Explainable artificial intelligence (XAI) · Convolutional neural networks (CNN) · Skin disease diagnosis

1 Introduction

1.1 Background and Motivation

Dermatological disorders are some of the most widespread diseases in the world [1]. Dermatologists may draw wrong or at least inaccurate conclusions at the beginning of the process of diagnosis without doing a deeper inspection with laboratory tests. Diagnostic errors have been seen as a big cause of preventable harm in United States health care, and, on the other hand, recent estimates of average misdiagnosis-related harms may vary widely.

In the year 2015, the National Academy of Medicine discovered that misdiagnosis is the third biggest cause of death among victims which is the cause for most of the paid medical misbehavior, claims Abbasi [2]. Total estimates adding total autopsy detected error rates as well as the sum of hospital deaths point to a possible 40,000–80,000 misdiagnosis-related deaths in United States hospitals yearly. According to [3], approximately $750 billion dollars is misused annually as a result of needless expenses on medications, procedures, work time lost, among others as a result of diagnostic errors. Primary care doctors are crucial in the delivery of timely as well as accurate diagnosis including the choosing of proper treatment. Failure to assess patient conditions can result in catastrophic results. Misdiagnosis can exact an emotional cost, in addition to the financial costs. The diagnosis process is therefore a vital, collaborative, and complex activity that includes information gathering as well as clinical reasoning to figure out a victim's health issue.

This research on explainable artificial intelligence (XAI) in skin disease diagnosis contributes to the community: through enabling non-technical individuals, such as end users, to understand how AI systems function and answer questions about their behavior, academia: Software developers and machine learning practitioners can utilize explanations to ensure that AI models and projects meet requirements, and benefits afflicted individuals: Through employing AI, healthcare experts can quickly and accurately determine the causes of specific skin diseases and recommend appropriate treatments. By harnessing the power of explainability and interpretable machine learning techniques, we have built a model that not only ensures accurate and timely diagnoses but also fosters unwavering trust in the capabilities of AI. With a profound focus on improving patient outcomes, this work in XAI ushers in a new era of trust, transparency, and transformative impact in the field of dermatology.

1.2 Research Contribution

Explainable artificial intelligence (XAI) in skin disease diagnosis contributes towards enabling non-technical individuals, such as end users, to understand how AI systems function and answer questions about their behavior, academia: software developers and Machine learning practitioners can utilize explanations to ensure that AI models and projects meet requirements, and benefits afflicted individuals: through employing AI, healthcare experts can quickly and accurately determine the causes of specific skin diseases and recommend appropriate treatments.

By harnessing the power of explainability and interpretable machine learning techniques, a model was built that not only ensures accurate and timely diagnoses but also fosters unwavering trust in the capabilities of AI. With a profound focus on improving patient outcomes, this work in XAI ushers in a new era of trust, transparency, and transformative impact in the field of dermatology.

2 Existing Works

Skin conditions have a profound impact on a significant portion of the global population, with approximately 1.9 billion people affected [4]. However, due to the limited availability of dermatologists, many cases are handled by physicians with lower diagnostic accuracy. The consequences are severe, as skin cancer treatment costs surpass a billion dollars, and millions of individuals suffer from side effects such as hair loss, infections, itchiness, and burns. Early detection of cancer has been identified as a crucial factor in reducing the prevalence of such cases. Malignant illnesses like pyogenic granulomas and melanoma, as well as non-carcinogenic conditions like dermatofibroma and melanocytic nevi, pose significant harm to the skin and can spread if left untreated [5, 6]. Addressing this issue, recent research [7] emphasizes the importance of early skin disease diagnosis to alleviate the burden on individuals and society as a whole. Scholarly literature by [8] scrutinized the various non-invasive approaches of evaluating the visual features of the skin. Ophthalmic recordings were built on the experiments of nonionizing electromagnetic radiation with the skin. In their research, low frequency radiation was what they used and they examined skin disorders using over 8 optical technologies some of which included: two-photon microscopy, in vivo confocal microscopy, and the diffuse reflectance spectroscopy (DRS).

More research carried out by [9] indicates that the diagnosis of skin diseases is now being done on a nanoscale using nanoparticles (Janus nanoparticles) owing to the advancement in technology. Nanotechnology is an emerging scientific discipline that hones tools and devices of sizes less than 100 nm, with peculiar molecular uses. These particles are nanocapsules with impressive applications in cosmetics and drug delivery. They have compartments affiliated with an aqueous compartment bound

by a layer of natural oils [9]. Several other methods of skin diagnosis have been discussed by [10], many of which include:

1. the use of compact digital cameras for snap photography
2. fiber-optic microscopy for skin surface imaging
3. computerized image analysis of clinical photos.

Reference [11] exploited CNN methods to implement a novel strategy. They employed CNN approaches which produced advanced classification models, rather than neural networks to categorize the data. 379 test samples and 900 training samples were employed in the approach. The samples fell into the benign and malignant categories. The image size was downscaled, and it was converted to grayscale as part of the preprocessing. Following that, the system made use of AlexNet. After feature reduction using LDA approaches, 4 alternative classifiers and metrics were employed, with KNN providing the best specificity (99.9%) and precision (86%) of all the classifiers. Reference [12] leveraged pretrained networks in two different ways: as a feature extractor and as a transfer learning paradigm. The dataset, PH2, was obtained from Hospital Pedro Hispano in Portugal's dermatology department. There are 200 images used in total. The required preprocessing was done, because the preceding networks require inputs of various sizes. For feature extraction and transfer learning, AlexNet and VGG16 were separately employed. As a transfer learning model, VGG16 produced the greatest results out of all of these techniques: a precision of 97.5% and a specificity of 96.87%. Reference [13] demonstrated an automated method for identifying melanoma from skin cancer ultimately improving precision while utilizing a minimal number of features. The dataset was divided, and features of interest were selected. According to test results obtained, which were based on 200 pictures, just six criteria were effective for identifying melanoma and precision of 86.67% was achieved [5, 6]. Reference [14] proposed a novel system to categorize skin disorders. Two layers of the system were put into place. A 112-lesion dataset separated into three subsets was used to train the binary classifiers. The three independently trained models were compared using the LOO classification process, a classification cross-validation technique used in machine learning as well as statistics to assess the performance of an artificial intelligence classification model, to pick the best model. The best outcomes came from a model with four multinomial kernels. This approach achieved an $F1$-score of 85%.

While significant advancements have been made in leveraging highly advanced artificial intelligence algorithms for machine learning, there is a critical need to comprehend the reasoning behind model predictions, decisions, and internal representations, particularly in crucial domains such as health care. This understanding is essential for ensuring the transparency and interpretability of the applied algorithms, particularly in the context of skin disease diagnosis. By unraveling the rationale behind the predictions and decisions made by the machine learning models, one can enhance the trust and reliability of the diagnostic process. This allows healthcare professionals to make informed decisions based on the explanations provided by the models, ultimately improving the accuracy and effectiveness of skin disease diagnosis.

3 Computational Problem and Solution

3.1 Problem Definition

Dermatological problems are a major worldwide health burden. Due to the time-consuming nature of the process, dermatologists confront difficulties in providing reliable diagnoses. Current skin condition diagnosis methods are frequently difficult, expensive, and unavailable in low-resource settings, and, if they are, they are traditional methods.

Therefore, this research aims to address the problem of accurate diagnosis of skin disorders using AI techniques. Misdiagnosis can have severe consequences for patient health and strain the healthcare system. A more efficient and accessible diagnostic approach to improve patient outcomes would be expedient.

The current state of dermatological diagnosis, particularly in low-resource communities, involves extensive tests and analysis by dermatologists, which are traditional methods. However, these methods have limitations and can vary in results and time. AI and XAI present an opportunity to enhance the diagnostic process and improve accuracy. Despite advances, dermatological diagnosis remains difficult. Furthermore, the complexity and cost of current diagnostic procedures create impediments to accurate diagnosis.

4 Research Methodology

Deep learning has improved and experienced exponential evolution in recent years while playing a tremendous role in image classification in various research domains. Recent research shows that convolutional neural networks recently have achieved high performance in detecting numerous illnesses including coronary artery disease, malaria etc. A dataset was used for validating as well as training the model. Data were obtained from DermNet the largest dermatology source. The dataset includes images of 5 classes of skin diseases including melanoma, eczema, rosacea, psoriasis, acne, and seborrheic keratoses, https://dermnetnz.org/image-library. In recent research, researchers describe in detail the importance of data preprocessing in convolutional neural networks; hence, image data shall be passed through a preprocessing pipeline which comprises the following process: resizing, normalization, and image augmentation.

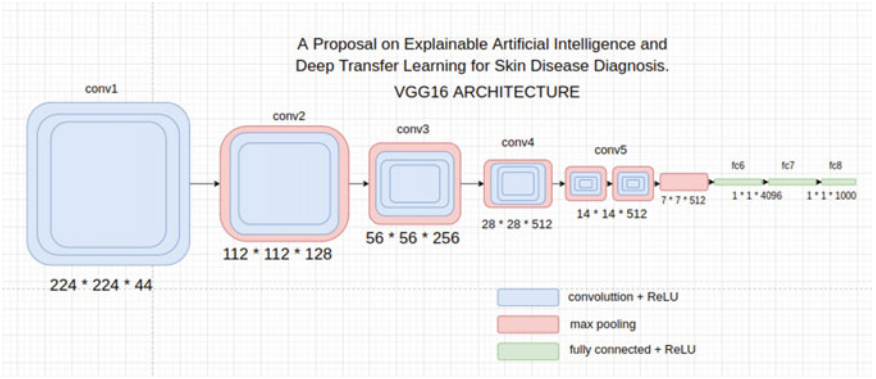


Fig. 1 Image of the VGG16 model architecture

4.1 Model Architecture Methodology

4.1.1 Data Sources

The data were obtained from DermNet, and it was comprised of 5 classes for the case study such as acne, eczema, rosacea, melanoma, and keratoses. The majority of the image data will be used for training, while a smaller portion will be reserved for testing. Figures 1, 2, 3, and 4 provide a sample representation of the classes used to train and test the mode. The split ratio in the training and testing data was 8:2.

4.2 Model Implementation

4.2.1 Image Augmentation

Data augmentation is a technique that entailed producing different data points from the original data with the intent of expanding the dataset domain. This is usually done by making small modifications to the data, such as adding minor translations, flipping, rotations, or noise. Research has shown that the use of augmented data can improve the results of these models by creating new and diverse data for training [15].

Resizing: In this research, the VGG16 model was used because of its fewer layers compared to VGG19 model hence less computational power, so the dataset must be resized to match the input configuration of the pretrained model. In this case, all images in the dataset were resized to 224 $\times$ 224, because this is the configuration used by the VGG16 network.

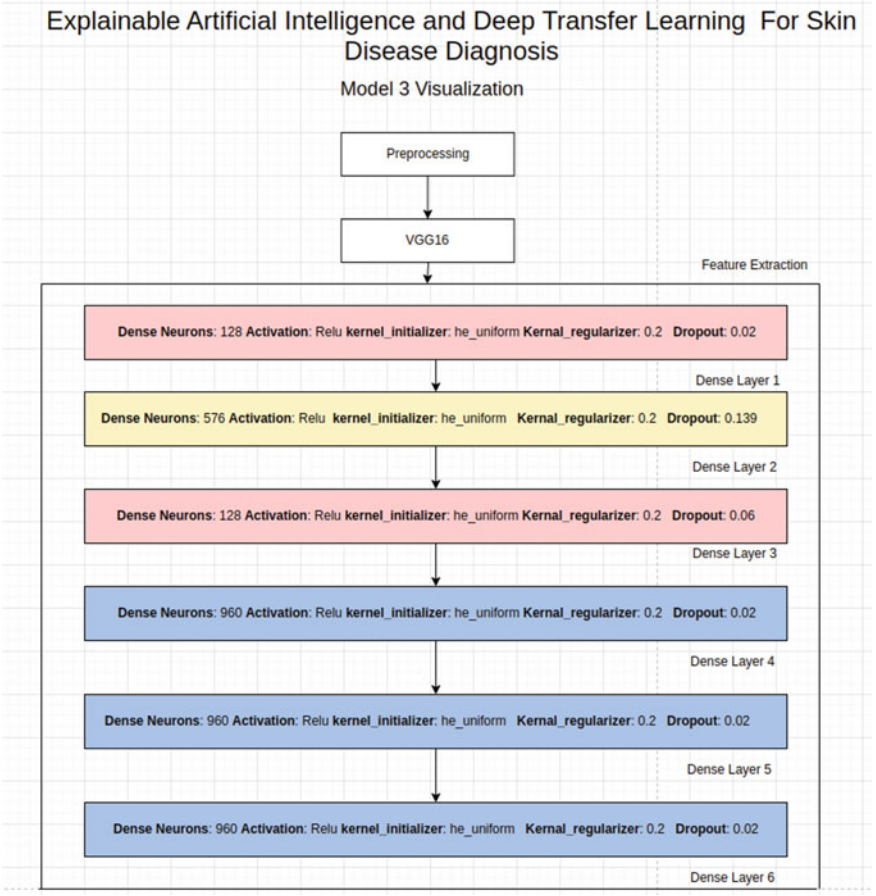


Fig. 2 Image of model visualization



Fig. 3 Sample images of skin diseases

Pixel Normalization: Pixel normalization is a way of normalizing pixel values in an image. The whole purpose is to transform the images such that they are suitable for training the model. After image data have been normalized, each rescaled image has values in a range from 0 to 255 (Table 1).

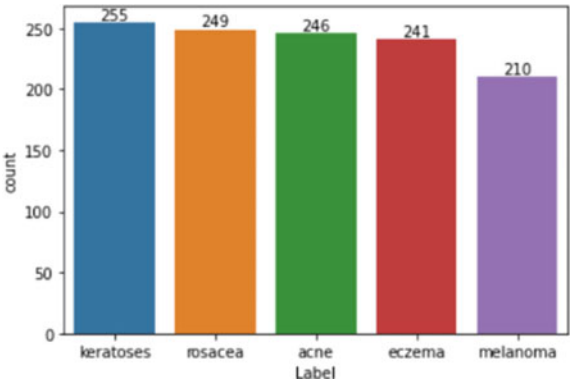


Fig. 4 Values used in data augmentation

Table 1 Table of values used in data augmentation

Process	Value
Rescale	1./255
Horizontal_flip	True
zoom_range	0.1
shear_range	0.1
Preprocessing_function	Preprocess_input

$$\text{normalized_pixel} = (\text{pixel_value} - \text{mean})/\text{standard_deviation}$$

4.3 Model Training and Evaluation

To select the best diagnosis model, the selected model is evaluated using benchmarks: recall, accuracy, precision, ROC curve, and $F1$-score. In the case of skin diagnosis, there are $Z = 5$ classes, resulting in a 5×5 matrix. The training time in this case was 18 h for one model. Each model was trained on i5 Intel processor, 12 GB RAM, 1 GB GPU with a Keras 2.6.0 library (Fig. 5).

Precision representing the portion of image samples that were correctly classified, and accuracy representing the allotment of the summation of correct model predictions (Fig. 6).

Recall represents the ratio of images which were accurately classified to the summation of images in the dataset. In this research, prioritize to optimize recall because the implications of misclassifying images could be catastrophic if the model was deployed in a working health facility. Finally, $F1$-score representing the classification integration of accuracy of recall and precision. The following is the formula for $F1$-score (Table 2).

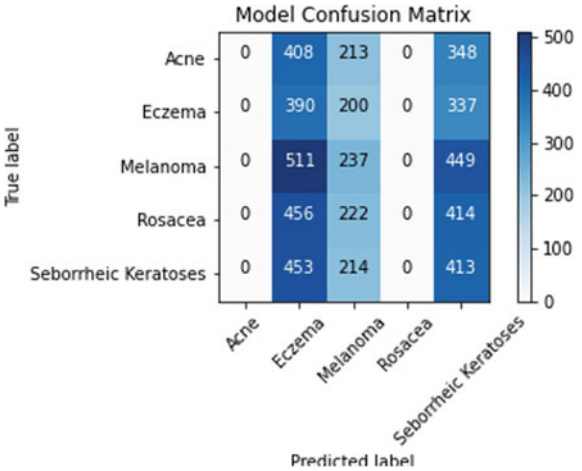


Fig. 5 Confusion matrix for the model considered

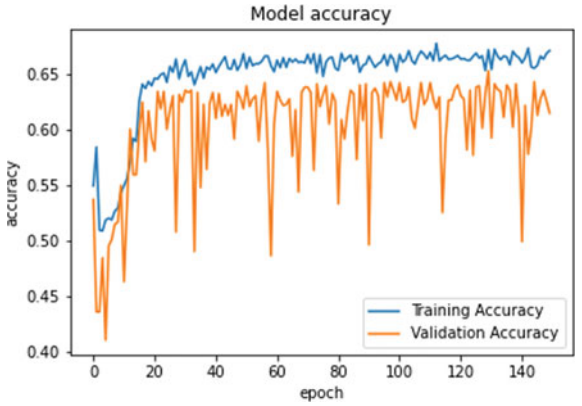


Fig. 6 An image of the model accuracy of the third model

$$F1 = (\text{recall} - 1 + \text{precision} - 1)/2$$
$$= 2(\text{precision} * \text{recall} + \text{recall})$$

Table 2 Table with a summary of the classification report for each of the trained neural networks

Model	Trainable parameters	Precision (%)	Recall (%)	Val. accuracy (%)	Accuracy (%)	$F1$ score
Model 1	27,309,509	23.59	21.394	35.525	37	0.2195
Model 2	12,598,085	51.34	48.64	65.025	64.654	0.4943
Model 3	5,333,317	68.904	65.45	65.925	67.654	0.6632

5 Explainable Artificial Intelligence

In this skin disease diagnosis system, the power of saliency maps was harnessed; SmoothGrad and Grad-CAM techniques were used to enhance the interpretability, reliability, and trustworthiness of deep learning model's predictions. Saliency maps, in particular, played a crucial role in providing visual explanations by highlighting the important regions in the input images that contributed most to the model's diagnoses.

The Grad-CAM (gradient-weighted class activation mapping) method, invented by Selvaraju and coauthors [14], is a powerful technique that has been applied in various domains, including skin disease diagnosis. In this skin disease diagnosis system, Grad-CAM was utilized to determine the important regions in input images that contribute the most to the prediction and classification of skin diseases. By analyzing the classification score slopes relative to the final convolutional feature map, Grad-CAM helped identify the areas in the image that have the greatest impact on the predicted diagnosis. These areas, where the gradient is significant, play a crucial role in the network's decision-making process for skin disease classification (Figs. 7 and 8).



Fig. 7 Sample images on acne and eczema

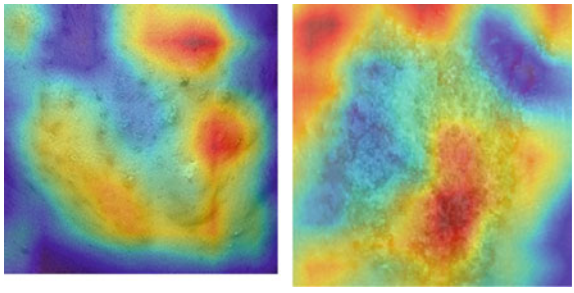


Fig. 8 Heatmaps of acne and eczema indicating important regions

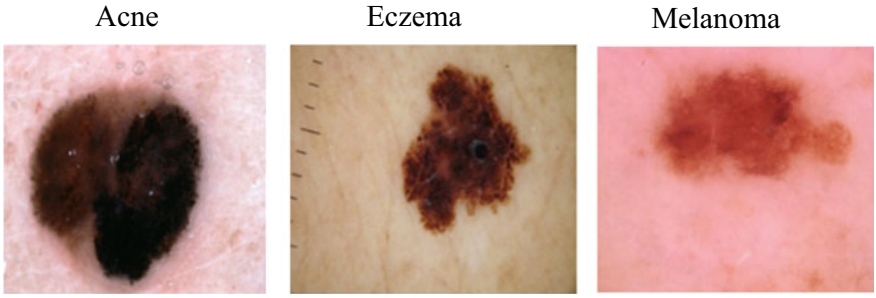


Fig. 9 Sample images on acne, eczema, and melanoma

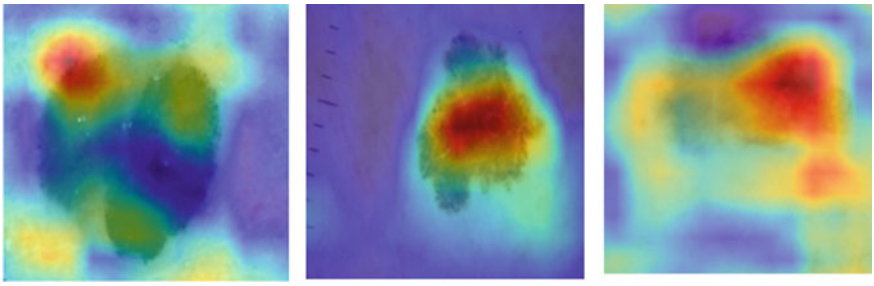


Fig. 10 Visual explanations with added noise on the images on acne, eczema, and melanoma

5.1 SmoothGrad

SmoothGrad was crucial in the skin disease diagnosis system by examining the model's sensitivity to small perturbations in the input image. It involved generating multiple copies of an input image with added noise to better understand how the model's output changes with small perturbations (Figs. 9 and 10).

5.2 Saliency Map

By calculating gradients with respect to the input image, saliency maps helped identify the pixels or regions that had the greatest influence on the model's predictions. These maps provided a visual representation of the model's attention and focus, highlighting the specific areas of the skin image that were most informative for the diagnosis.

In conjunction with SmoothGrad and Grad-CAM, saliency maps augmented the understanding of the decision-making process in skin disease diagnosis. SmoothGrad allowed us to analyze the model's sensitivity to small perturbations in the input images, ensuring the stability and robustness of the predictions. Grad-CAM provided

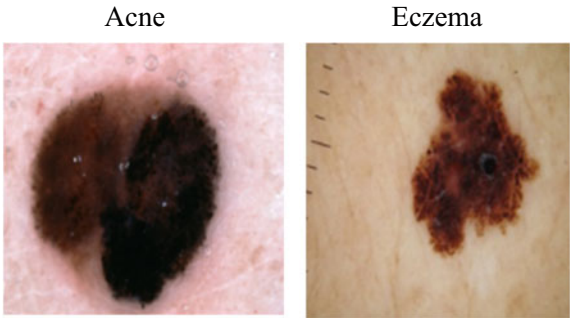


Fig. 11 Sample images of acne and eczema

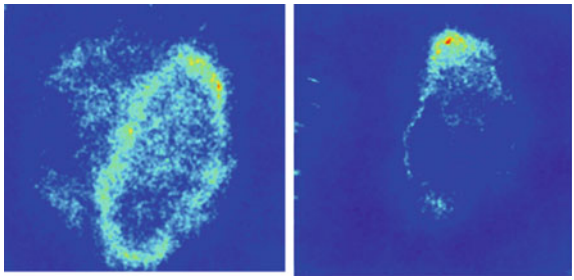


Fig. 12 Region of focus on acne and eczema using saliency maps

insights into the important regions of the image that influenced the model's decision by analyzing the gradients and convolutional feature maps.

By incorporating saliency maps into this system, dermatologists and healthcare professionals were offered a transparent and interpretable view of the model's diagnoses. The highlighted regions in the saliency maps provided visual explanations and enabled them to validate the model's predictions, fostering trust and confidence in the system (Figs. 11 and 12).

6 Limitations and Conclusion

6.1 Limitations and Recommendations

The model was only trained on images of people with a fairly toned skin mainly due to the fact that majority of the images in the original dataset were images of fairly toned skin infected people, and this raised questions on how the model would perform when required to diagnose people with a dark-skinned tone. The dataset's exclusive focus on fairly skinned people posed several challenges and drawbacks:

Lack of Diversity: By collecting data exclusively from fairly skinned individuals, the vast diversity present in human skin tones was not captured. This limitation led to

biased and inaccurate predictions when the model encountered people with different skin tones.

Generalizability Issues: Since the dataset is not representative of the entire population's skin tones, the trained model struggled to generalize its predictions to individuals with darker or lighter skin tones. This led to inaccuracies and reduced performance in real-world scenarios.

Biased Results: The model's training on a limited dataset perpetuated biases present in the data. It led to unequal or unfair treatment for individuals with darker skin tones when it came to medical assessments or diagnoses.

Lack of Inclusivity: By not considering individuals with different skin tones, critical information that could impact healthcare outcomes and diagnostics for a more diverse population was missed out on.

Limited Scope: Inaccessibility to data from hospitals due to the lack of institutional review board (IRB) approval restricted the dataset's scope to a specific subset of the population. This narrow scope hindered the model's ability to handle diverse medical conditions and scenarios.

Ethical Concerns: The lack of IRB approval raised ethical concerns about data privacy, patient consent, and ensuring that data collection and usage comply with ethical standards.

6.2 Conclusion

The integration of explainable artificial intelligence (XAI) has the potential to revolutionize the field of skin disease diagnosis by harnessing the power of imagery data. XAI techniques enable healthcare dermatologists to gain deeper insights into the decision-making process of AI models, providing them with a transparent and interpretable understanding of how diagnoses are made. Through analyzing and interpreting the important features and regions within the images, XAI empowers dermatologists and healthcare practitioners to make more accurate and informed diagnostic decisions. This not only enhances the accuracy and reliability of skin disease diagnosis but also builds trust and confidence in the AI-driven diagnostic systems. As a part of future work and research in XAI, one can expect significant improvements in the early detection, personalized treatment, and overall management of skin diseases, ultimately leading to better healthcare outcomes for patients.

Collection of imagery data specifically tailored for XAI in skin disease diagnosis from various health institutions shall be done. This shall ensure that the models are diverse, representative, and well-annotated for robust training of AI models. This contribution shall not only enhance the reliability of the diagnostic system but also promote the advancement of the field as a whole by expanding the available resources for researchers and practitioners. Additionally, the collection of imagery data enables

us to address specific challenges and gaps in the current datasets, such as the inclusion of underrepresented skin types or rare skin conditions. By undertaking this task, one can further improve the performance and generalizability of the XAI-based diagnostic approach, ultimately leading to more effective and accessible healthcare solutions for individuals affected by skin diseases.

References

1. Hellen N, Marvin G (2021) Interpretable feature learning framework for smoking behavior detection. arXiv preprint arXiv:2112.08178
2. Abbasi J (2016) A headline-grabbing study brings attention back to medical errors. JAMA 316(7):698–700
3. McGinnis JM, Stuckhardt L, Saunders R, Smith M et al (2023) Best care at lower cost: the path to continuously learning about health care in America
4. Liu Y, Jain A, Eng C, Way DH, Lee K, Bui P, Kanada K, de Oliveira Marinho G, Gallegos J, Gabriele S et al (2020) A deep learning system for differential diagnosis of skin diseases. Nat Med 26(6):900–908
5. Gulati S, Bhogal RK (2019) Detection of malignant melanoma using deep learning. In: International conference on advances in computing and data sciences. Springer, pp 312–325
6. Mustafa S, Kimura A (2018) A SVM-based diagnosis of melanoma using only useful image features. In: 2018 international workshop on advanced image technology (IWAIT). IEEE, pp 1–4
7. Maniraju M, Adithya R, Srilekha G (2022) Recognition of type of skin disease using CNN. In 2022 first international conference on artificial intelligence trends and pattern recognition (ICAITPR). IEEE, pp 1–4
8. Kollias N, Stamatas GN (2002) Optical non-invasive approaches to diagnosis of skin diseases. J Invest Dermatol Symp Proc 7:64–75
9. Kemel K, Deniset-Besseau A, Baillet-Guffroy A, Faivre V, Dazzi A, Laugel C (2020) Nanoscale investigation of human skin and study of skin penetration of janus nanoparticles. Int J Pharm 579:119193
10. Serup J, Jemec GB, Grove GL (2006) Handbook of non-invasive methods and the skin. CRC Press
11. Majtner T, Yildirim-Yayilgan S, Hardeberg JY (2019) Optimised deep learning features for improved melanoma detection. Multimed Tools Appl 78(9):11883–11903
12. Mete M, Sirakov NM, Griffin J, Menter A (2016) A novel classification system for dysplastic nevus and malignant melanoma. In 2016 IEEE international conference on image processing (ICIP). IEEE, pp 3414–3418
13. Perez L, Wang J (2017) The effectiveness of data augmentation in image classification using deep learning. arXiv preprint arXiv:1712.04621
14. Selvaraju RR, Cogswell M, Das A, Vedantam R, Parikh D, Batra D (2017) Grad-cam: visual explanations from deep networks via gradient-based localization. In: Proceedings of the IEEE international conference on computer vision, pp 618–626
15. Marvin G, Alam MGR (2022) Explainable augmented intelligence and deep transfer learning for pediatric pulmonary health evaluation. In: 2022 international conference on innovations in science, engineering and technology (ICISET)

A Comparison Study Between Otsu's Thresholding, Fuzzy *C*-Means, and *K*-Means for Breast Tumor Segmentation in Mammograms



Moustapha Mohamed Saleck, Nagi Ould Taleb, Mohamed El Moustapha El Arby Chrif, and El Benany Mohamed Mahmoud

Abstract Currently, the technique of digital mammography is considered the first screening tool used for detecting breast cancer in early stages. The shape and margin of masses in mammograms play a crucial role in the classification of benign and malignant tumors. Several machine learning (ML) algorithms have demonstrated high performance in breast mass edge segmentation from mammograms. The main objective of this paper is to present a comparative study of three algorithms of ML where: *C*-means clustering, *K*-means clustering, and Otsu's thresholding for automatically detecting the lesion edges on mammogram images. Two groups ($C = 2$) are assigned to clustering as output for each method. In order to achieve high accuracy in contour extraction, only relevant pixels are selected as inputs. This study will highlight the advantages and disadvantages of these algorithms, which are expected to improve the detection of breast masses contours in mammographic images. The mini-MIAS database is used to evaluate the obtained results which show a higher performance in terms of segmentation accuracy when using fuzzy *C*-means algorithm. However, a faster lesion extraction process can be achieved by utilizing the *K*-means clustering and Otsu's threshold.

Keywords Breast lesion segmentation · Mammograms · K-means clustering · Fuzzy *C*-means clustering · Otsu's thresholding

M. Mohamed Saleck (✉)
LAROSERI Laboratory, Department of Computer Sciences, Faculty of Sciences, University of Chouaib Doukkali, El Jadida, Morocco
e-mail: saleck.moustapha@gmail.com

N. Ould Taleb
Computer Sciences Department, Higher Institute of Digital, Nouakchott, Mauritania
e-mail: nagi.taleb-aly@supnum.mr

M. El Moustapha El Arby Chrif · E. Benany Mohamed Mahmoud
Department of Mathematics and Computer Science, University of Nouakchott AL Aasriya, Nouakchott, Mauritania

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2023

S. Shakya et al. (eds.), *Fourth International Conference on Image Processing and Capsule Networks*, Lecture Notes in Networks and Systems 798,
https://doi.org/10.1007/978-981-99-7093-3_48

1 Introduction

Recent data from the GLOBOCAN database in 2020 [1] reported over 2,300,000 new instances of breast cancer and 685,000 related deaths globally. Multiple studies emphasize that early detection and treatment significantly lower mortality rates and enhance patient survival [2]. Over the past decade, diverse medical imaging techniques for breast examination have emerged, including mammography, thermography, breast ultrasound, MRI, PET, optical imaging, CT, scintimammography, and electrical impedance-based imaging [3]. Digital mammography remains the most prevalent tool in breast screening centers.

Despite the crucial role of digital mammography in early breast disease diagnosis, visual assessment by radiologists often yields unsatisfactory outcomes [4]. To address this, computer-aided diagnosis tools are essential to aid doctors in interpreting mammograms and making informed decisions [5]. The segmentation process, which entails isolating tumor boundaries from image details, is pivotal for CADx systems serving as secondary readers alongside doctors. Various segmentation techniques have been proposed for lesion contour extraction from digital mammograms.

Researchers have introduced diverse strategies for mammographic image segmentation. Xie et al. [6] introduced an innovative approach that involves employing a pulse-coupled neural network (PCNN) algorithm combined with the level set method to establish an initial contour. Subsequently, an enhanced level set evolution is utilized to segment the images and attain the final outcomes. Berber et al. [7] devised a fresh breast mass contour segmentation (BMCS) technique for digital mammograms. This method harnesses the classical seed region growing method to enhance mass boundaries within a specified region of interest (ROI). The approach's distinguishing feature lies in its adaptability to fine-tune threshold values and incorporate suitable stopping conditions, effectively addressing under and over segmentation challenges. Ittannavar et al. [8] put forth a novel multiobjective optimization method for extracting breast masses from mammograms. Their procedure involves image denoising and enhancement, followed by the implementation of an electromagnetism optimization algorithm to segment both noncancerous and cancerous regions in mammograms. Sadad et al. [9] employ a cascaded approach, utilizing the fuzzy *C*-means algorithm in combination with region growing, to segment breast tumors in mammographic images. They augment this by incorporating texture-based features such as LBP-GLCM, LPQ, and hybrid attributes to capture the distinct characteristics of segmented tumors. Thawkar et al. [10] amalgamate three techniques: Otsu's global thresholding, morphological operations, and watershed transformation to create an automated method for detecting suspicious masses within mammograms.

Chattaraj et al. [11] propose a novel algorithm based on marker-controlled watershed principles for breast lesion segmentation. Their work leverages catchment basins on the topographic surface of mammograms. Makandar et al. [12] engineer an efficient system capable of detecting abnormalities in mammograms, utilizing Otsu's thresholding and morphological operations. Hu et al. [13] devise an innovative

methodology for spotting suspicious lesions in mammograms. This approach integrates adaptive global and local thresholding segmentation, effectively segmenting multiresolution sub-images extracted from the original mammogram. Xu et al. [14] develop an original model adaptive receptive field network (ARF-Net) for segmenting breast masses across mammographic and ultrasound images. Li et al. [15] introduce an end-to-end deep learning framework to predict lesion segmentation and diagnosis outcomes. This approach employs dual convolutional neural networks. Li et al. [16] propose a model named region-based graph convolution coupled with the atrous spatial pyramid pooling network (RGC-ASPP-Net) for mammogram mass segmentation. Kanungo et al. [17] suggest a hybrid segmentation approach for detecting suspicious breast masses in mammograms. Their strategy involves the utilization of fuzzy *C*-means, *K*-means, fuzzy possibilistic *C*-means (FPCM), genetic algorithm, and ant colony optimization. Roy et al. [18] present a combined approach, integrating convolutional neural networks (CNN) with connected component analysis (CCA) for preprocessing-free breast mass detection. This study involves comparing the outcomes of *K*-means and fuzzy *C*-means algorithms.

This paper focuses on comparing the segmentation performance of three suggested methods: the fuzzy *C*-means algorithm, *K*-means algorithm, and Otsu's thresholding. The underlying concept of these methods involves adapting a global threshold to identify pertinent pixels, which are then used as input for *C*-means, *K*-means, and Otsu's thresholding, leading to the generation of two clusters as outputs, with one cluster representing the lesion.

The subsequent sections of this paper are structured as follows:

Section 2 delves into the description of materials and methods employed in this study. Section 3 presents the experimental results and a comprehensive comparison of the proposed methods. And, finally, Sect. 4 concludes the paper by summarizing the findings and insights obtained.

2 Materials and Method

In this section, the details of proposed methods are explained. Figure 1 shows graphically the different steps of the suggested system.

2.1 Image Preprocessing

The presence of artifacts or noises on mammograms could affect the accuracy of breast mass segmentation, which can lead to unsatisfactory results in computer-aided diagnosis systems (CADs). Mammograms preprocessing is the step taken to simplify the recognition of abnormalities, minimize noise, and preserve meaningful information. The median filter with a 3×3 neighborhood window is used in this

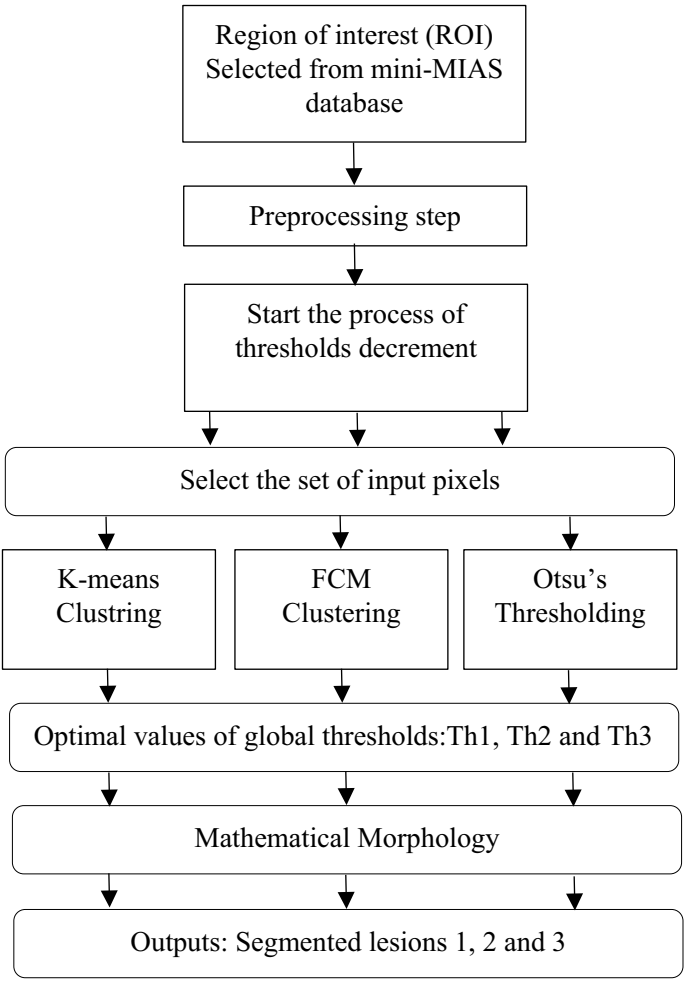


Fig. 1 Block diagram of the suggested methods

work to eliminate any expected noise from the region of interest (ROI) in original mammograms.

2.2 Fuzzy C-means Algorithm

Fuzzy *C*-means (FCM) algorithm was first presented by Dunn in 1973 [19] and then improved by Bezdek in 1983 [20]. FCM algorithm is a technique of clustering which allows one pixel of image to belong to two or more clusters with a membership degree $u_{ik} \in [0, 1]$. This algorithm is based on optimizing the objective function (1):

$$J = \sum_{i=1}^{c} \sum_{k=1}^{n} u_{ik}^{m} d^2(x_k, v_i) \tag{1}$$

$$0 \leq u_{ik} \leq 1 \quad \forall i \in [1, c], \quad k \in [1, n]$$

$$\sum_{k=1}^{c} u_{ik} \; \forall i \quad k \in [1, n]$$

$$0 < \sum_{k=1}^{n} u_{ik} < n \quad \forall i \in [1, c]$$

$$u_{ik} = \frac{(d(x_k, v_i))^{\frac{2}{m-1}}}{\sum_{j=1}^{n} (d(x_k, v_i))^{\frac{2}{m-1}}} \tag{2}$$

$$v_i = \frac{\sum_{k=1}^{n} u_{ik}^{m} x_k}{\sum_{k=1}^{n} u_{ik}^{m}} \tag{3}$$

where each cluster is described by its centroid $c_{i,}$, m is the fuzzy index often taken $m = 2$, n represents the number of pixels in image, u_{ik} is the membership degree pixel x_k to the centroid c_i, the Euclidean distance between $x_{k,}$ and c_i is calculated by $d_{ik} = x_k - c_i$. The elements values of membership matrix $U = [u_{ik}]$ are ranging in [0, 1]. The objective function J (1) is optimized by updating membership matrix (2) and the cluster centroid (3).

2.3 K-means Algorithm

The *K*-means clustering is one of the simplest unsupervised learning algorithms that can be applied to solve the clustering segmentation problem [21]. *K*-means algorithm allows to classify the set of input pixels from given image into multiple clusters based on their inherent distance from each other.

Given a set of n pixels $\{x_1, x_2, \ldots, x_n\}$, *K*-means algorithm partitions the pixels into k cluster $\{C_1, C_2, \ldots, C_k\}(k < n)$ so as to minimize the distance within the cluster (4).

$$V = \sum_{i=1}^{k} \sum_{x_i \in s_j} ||x_j - v_i||^2 \tag{4}$$

$$v_i = \frac{\sum_{j=1}^{k} x_j^i}{n_i} \tag{5}$$

where k is the number of cluster and C_i is the number of pixels in ith cluster, $||x_j - v_i||^2$ is the Euclidean distance between x_i and v_i, v_i is the mean of ith in C_i during each iteration.

2.4 Otsu's Thresholding

Otsu's thresholding technique is one of the most widely used algorithms in image segmentation. The effectiveness of this method lies in its ability to select automatically the threshold. Otsu's method also works on the normalized histogram for discriminating the interclass threshold. This method starts by calculating the global mean intensity (u_1, u_2), the class probabilities (q_1, q_2), the individual class variances (σ_1^2, σ_2^2), and finally the weighted within-class variance (σ_ω^2).

$$\sigma_\omega^2(k) = q(t)\sigma_1^2(t) + q_2(t)\sigma_2^2(t) \tag{6}$$

$$q_1(t) = \sum_{i=0}^{t} P_i \quad \text{and} \quad P_2(t) = \sum_{i=t+1}^{l} P_i \tag{7}$$

$$u_1(t) = \sum_{i=0}^{t} \frac{i\,P_i}{P_1(t)} \quad \text{and} \quad u_2(t) = \sum_{i=t+1}^{l} \frac{i\,P_i}{P_2(t)} \tag{8}$$

$$\sigma_1^2(t) = \sum_{i=0}^{t} [i - u_1(t)]^2 \frac{i\,P_i}{P_1(t)} \tag{9}$$

$$\sigma_2^2(t) = \sum_{i=t+1}^{l} [i - u_2(t)]^2 \frac{i\,P_i}{P_2(t)} \tag{10}$$

Otsu's method needs to run the full range of $t \in [1, 256]$ and choose the value of t which minimizes the weighted within-class variance (6).

2.5 Breast Mass Segmentation

Breast mass segmentation from the region of interest (ROI) is a challenging task in the development of computer-aided diagnosis systems (CADs) for the detection of breast cancer in mammograms. Our purpose behind this research is to draw comparison between three methods for automatic detection of the boundaries of breast mass from the ROI. These methods are based on Otsu's thresholding, fuzzy C-means algorithm, and K-means algorithm. The main idea of these methods is to apply each algorithm in a suitable range of pixels (inputs) where the first limit is the maximum

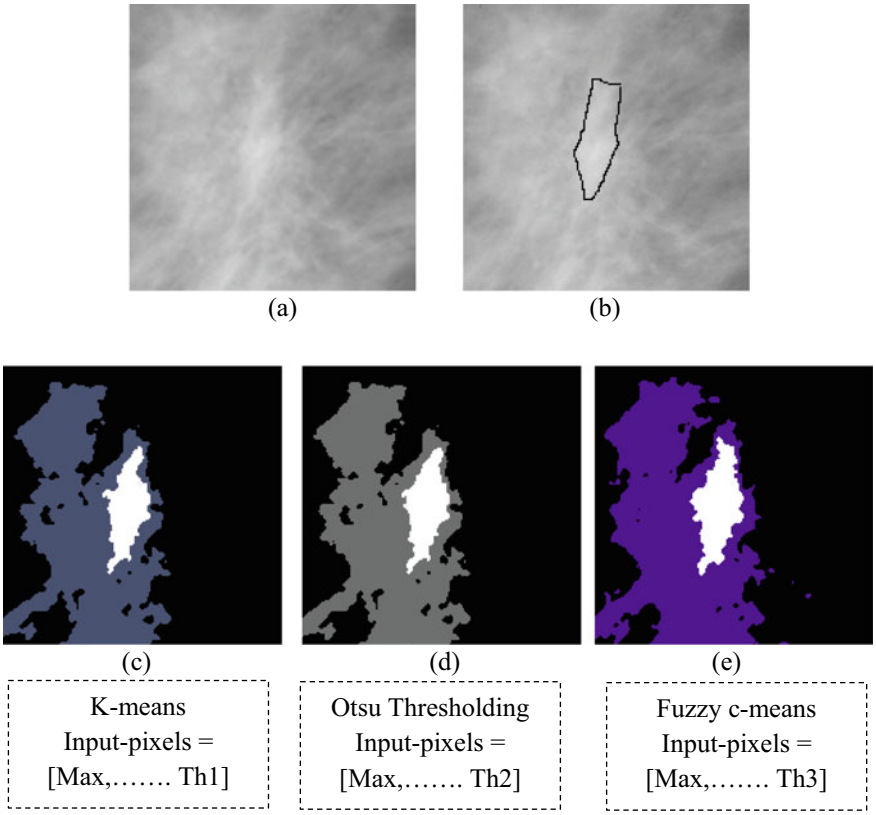


Fig. 2 Segmentation results for mdb_145 from MIAS. **a** Original ROI. **b** Lesion contour denoted by radiologist. **c** Mass segmented using *K*-means. **d** Mass segmented using Otsu's thresholding. **e** Mass segmented using FCM

value of intensity (Max) and the second is an optimal threshold (Th) which is able to eliminate all pixels that can affect the accuracy of segmentation. The number of clusters is chosen as $C = 2$. Morphological operations are applied in the final stage so as to remove imperfections from the segmented image (Fig. 2).

To find out the optimal threshold that permit to obtain the appropriate set of pixels for lesion segmentation, we have adopted the technique proposed in [22].

3 Experimental Results and Comparison

In order to assess the efficacy of the presented techniques, the experiment utilizes the mini-MIAS database. This database encompasses 322 mammograms, with 21 of these images employed for testing purposes. Each image from the MIAS was

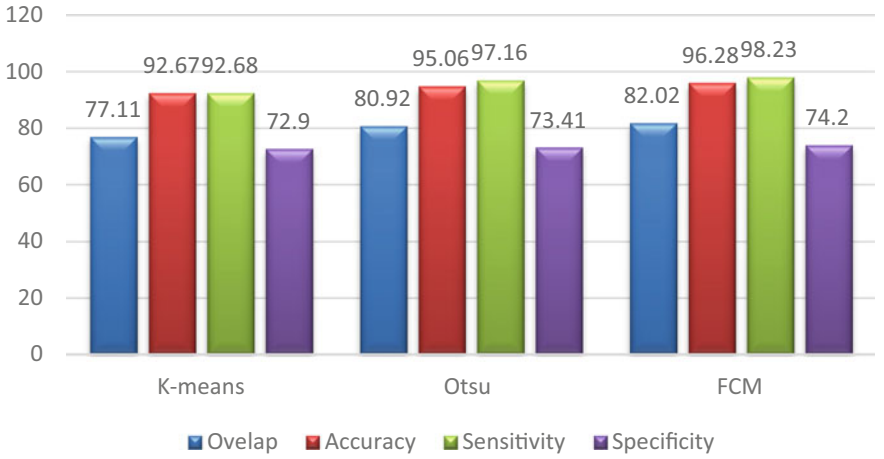


Fig. 3 Comparison of result

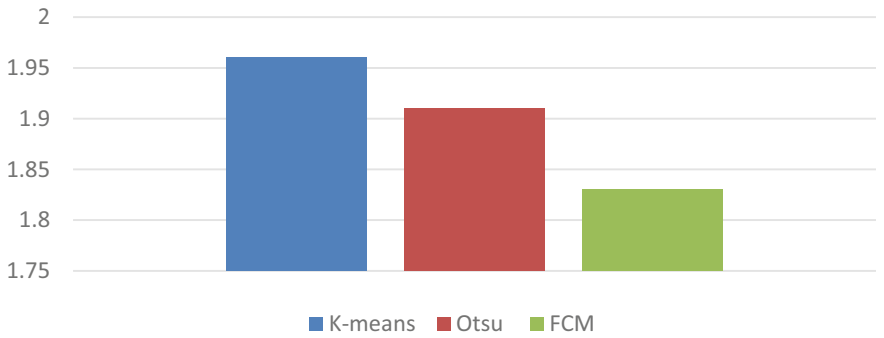


Fig. 4 The time complexity of different methods

digitized at 50 μm, where each pixel was represented by 8 bit. The size of the ROI taken for this work is 174 × 174. The performance evaluation of the proposed methodologies *K*-means, *C*-means, and Otsu's threshold is depicted in Fig. 3 (Fig. 4).

4 Conclusion

Breast cancer currently ranks among the prominent health challenges, emphasizing the significance of early detection to elevate patient survival rates. Mammogram-based breast screening is the prevailing technique in developed nations for early diagnosis. However, a robust and effective approach to extracting breast masses from the region of interest (ROI) within mammogram images remains an intricate

undertaking. This paper addresses this challenge by offering a comparative analysis of three proposed breast tumor segmentation methods applied to the ROI. These methods hinge on *K*-means clustering, *C*-means clustering, and Otsu's thresholding. The study presents insights into the efficiency and time required for completing the segmentation process for each method.

References

1. Sung H, Ferlay J, Siegel R et al (2021) Global cancer statistics 2020: GLOBOCAN. CA Cancer J Clin 71:209–249. https://doi.org/10.3322/CAAC.21660
2. Mandelblatt JS et al (2009) Effects of mammography screening under different screening schedules: model estimates of potential benefits and harms. Ann Intern Med. https://doi.org/10.1059/0003-4819-151-10-200911170-00010
3. Sree SV, Ng EY-K, Acharya RU, Faust O (2011) Breast imaging: a survey. World J Clin Oncol 2(4):171–178. https://doi.org/10.5306/wjco.v2.i4.171
4. Gandomkar Z, Mello-Thoms C (2019) Visual search in breast imaging. Br J Radiol 92(1102). https://doi.org/10.1259/bjr.20190057
5. Zebari DA et al (2021) Systematic review of computing approaches for breast cancer detection based computer aided diagnosis using mammogram images. Appl Artif Intell 35(15):2157–2203. https://doi.org/10.1080/08839514.2021.2001177
6. Xie W, Li Y, Ma Y (2016) PCNN-based level set method of automatic mammographic image segmentation. Optik (Stuttg) 127(4):1644–1650. https://doi.org/10.1016/j.ijleo.2015.09.250
7. Berber T, Alpkocak A, Balci P, Dicle O (2013) Breast mass contour segmentation algorithm in digital mammograms. Comput Methods Programs Biomed 110(2):150–159. https://doi.org/10.1016/j.cmpb.2012.11.003
8. Ittannavar SS, Havaldar RH (2022) Segmentation of breast masses in mammogram image using multilevel multiobjective electromagnetism-like optimization algorithm. Biomed Res Int 2022:8576768. https://doi.org/10.1155/2022/8576768
9. Sadad T, Saba T, Hussain A (2018) Fuzzy C-means and region growing based classification of tumor from mammograms using hybrid texture feature. J Comput Sci 29. https://doi.org/10.1016/j.jocs.2018.09.015
10. Thawkar S, Ingolikar R (2017) Automatic detection and classification of masses in digital mammograms. Int J Intell Eng Syst 10(1):65–74. https://doi.org/10.22266/ijies2017.0228.08
11. Chattaraj A, Das A, Bhattacharya M (2017) Mammographic image segmentation by marker controlled watershed algorithm. In: Proceedings—2017 IEEE international conference on bioinformatics and biomedicine, BIBM 2017, pp 1000–1003. https://doi.org/10.1109/BIBM.2017.8217793
12. Makandar A, Halali (2015) Threshold based segmentation technique for mass detection in mammography. https://doi.org/10.17706/jcp.11.6.472-478
13. Hu K, Gao X, Li F (2011) Detection of suspicious lesions by adaptive thresholding based on multiresolution analysis in mammograms. IEEE Trans Instrum Meas 60(2):462–472. https://doi.org/10.1109/TIM.2010.2051060
14. Xu C, Qi Y (2022) ARF-Net: an adaptive receptive field network for breast mass segmentation in whole mammograms and ultrasound image. Biomed Signal Process Control 71:103178. https://doi.org/10.1016/j.bspc.2021.103178
15. Li H, Chen D et al (2022) Dual convolutional neural networks for breast mass segmentation and diagnosis in mammography. EEE Trans Med Imaging 41(1):3–13
16. Li Z, Deng Z, Liao J (2022) Multi-stream information-based neural network for mammogram mass segmentation. Artificial neural networks and machine learning—ICANN 2022, pp 267–278. 978-3-031-15919-0

17. Kanungo GK, Singh N, Dash J, Mishra A (2015) Mammogram image segmentation using hybridization of fuzzy clustering and optimization algorithms. Intelligent computing, communication and devices, pp 403–413
18. Dunn JC (1973) A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters. J Cybern 3(3):32–57. https://doi.org/10.1080/01969727308546046
19. Roy A, Singh BK, Banchhor SK (2022) Segmentation of malignant tumours in mammogram images: a hybrid approach using convolutional neural networks and connected component analysis. Expert Syst 39(1):e12826
20. Bezdek JC (1981) Pattern recognition with fuzzy objective function algorithms, 1st edn. Springer, New York, NY
21. Hartigan JA, Wong MA (1979) A K-means clustering algorithm. J R Stat Soc Ser C (Appl Stat) 28(1):100–108 (9 p.)
22. Saleck MM, El Moutaouakkil A, Moucouf M, Bouchaib M, Samira H, Zineb J (2018) Breast mass segmentation using a semi-automatic procedure based on fuzzy C-means clustering. Telkomnika (Telecommun Comput Electron Control 16(2):665–672. https://doi.org/10.12928/TELKOMNIKA.v16i2.6193

Author Index

A
Abhishek, S., 511
Aditya Shreeram, 115
Afridi, Sheikh Shemanto, 239
Afruza Begum, 647
Akhendra Kumar, P., 253
Akila, G., 563
Anjali, T., 511
Ankita Kumari, 551
Antony Freeda Rani, M., 323
Anvay Deshpande, 551
Arafat Rahman, S. M., 239
Arif, Abu Shamim Mohammad, 523
Asanda, Enoch Hall, 711
Ashfan Khan, 551
Ashwini Sanjay Gaikwad, 389
Asok Bandyopadhyay, 411
Attar, V. Z., 197
Azeroual, A., 157

B
Babasaheb R. Patil, 57
Babeetha, S., 435
Babitha, M. S., 463
Baby Kalpana, Y., 73
Balangcod, Ashlyn Kim D., 131
Barnali Pal, 411
Belhoussine Drissi, T., 39
Belousov, Ilya, 449
Benany Mohamed Mahmoud, El, 725
Bhakti Sarag, 281
Bivas Dam, 411
Bomken Kamdak, 339

C
Çetinkaya, Bedrettin, 269
Chandrasekar Venkatachalam, 213
Chashu Agrawal, 665
Chirag Bhandari, 665
Çimtay, Yücel, 269
Cota Navin Gupta, 115

D
Dhafer, W., 157
Dharini, S., 145
Dhiyanesh, B., 73
Diana A Andrushia, 475
Diana Andushia, A., 463
Dipak Chandra Patranabis, 411

E
El Moustapha El Arby Chrif, Mohamed, 725
El Ouahabi, Y., 157
El yousfi Alaoui, M. H., 157

G
Gayathri, R., 563
Genemo, Musa, 225
Gilbert, Yiga, 181
Gom Taye, 339
Günay, Fatma Nazli, 269

H
Haque, Samia Shameem, 297

© The Editor(s) (if applicable) and The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2023

S. Shakya et al. (eds.), *Fourth International Conference on Image Processing and Capsule Networks*, Lecture Notes in Networks and Systems 798,
https://doi.org/10.1007/978-981-99-7093-3

Hareesh, P. M., 373
Herald Anantha Rufus, N., 323
Himanka Sekhar Mondal, 411

J
Jay B. Simha, 581
Jennifer, Sanjeda Sara, 239, 297

K
Kalungi, Joshua Edward, 181
Kalyani, G., 425
Kaveri Pawar, 89
Kawsar, Al, 297
Kayanja, Emmy William, 181
Ketan Bodhe, 611
Kinjal Gautam, 1
Kinjal Patel, 1
Konumuri Kalyan Suhas, 425
Kyagaba, Jonah Mubuuke, 181

L
Lakshmi Harika Palivela, 647
Lasya, M., 169
Lokesh Gopinath, 629

M
Mali, Prabhat, 697
Manasa, A., 309
Mandava Venkata Sri Sai Surya, 425
Manivannan, K., 213
Marvin, Ggaliwango, 181, 711
Massar, H., 39
Mayanja, James, 711
Mayur Chauhan, 1
Mayuri Kundu, 581
Medha Wyawahare, 551, 665
Mehathab, A., 463
Milind Shah, 1
Miyara, M., 39
Mohamed Saleck, Moustapha, 725
Mohammad Tahir, 611
Monish Mohanty, 491
Muddayya Swami, 197
Mwesigwa, Joshua, 711

N
Nikunj Bhavsar, 1
Noman Al, Abdullah, 523
Nsiri, B., 39, 157

O
Ould Taleb, Nagi, 725

P
Pabico, Jaderick P., 131
Panday, Sanjeeb Prasad, 697
Peeta Basa Pati, 491
Pradnya N. Gokhale, 57
Pranav Dalve, 665
Pranay Meshram, 611
Prathamesh Patil, 89
Pratham Ghatkar, 89
Prathibha Prakash, 511
Premanand, V., 647
Priyanka Acharya, 551
Priyanka Sakpal, 89
Puranam Revanth Kumar, 253

R
Rachit Verma, 597
Radha, R., 73
Radhesyam Vaddi, 169
Rajesh Kumar Jha, 253
Rajesh Saturi, 539
Rajkumar, S., 73
Rakhi Bhardwaj, 281
Ravi, K., 677
Redwan Ahmed, Md., 239
Renuka Devi, S. M., 309
Reza, Ahmed Wasif, 239, 297
Rima, Sharmin Akther, 297
Rogachev, Aleksey, 449
Rogachev, Dmitry, 449
Ruhan Bevi, A., 629
Rushikesh Sanap, 197

S
Sagar Lahade, 353
Sahil Shendurkar, 281
Sai Chandu, 677
Sajid Anam Ifti, Md., 239
Sajitha, P., 475
Sameer Joshi, 57
Sanjay Jain, 145
Saraswathi, P., 73
Satish, S., 323
Selva Balan, M., 551
Senthil Rama, R., 323
Shabeer, S. K., 169
Shajulin Benedict, 597
Shakya, Aman, 697

Shanmugavalli Venkatachalam, 213
Shantanu Chakraborty, 197
Shashank Bhosagi, 89
Shinu Abhi, 581
Shreya Daga, 665
Shubham Waghule, 281
Sivaiah Bellamkonda, 373
Soulaymani, A., 157
Sridhar, S. S., 435
Srikanth Reddy, G., 539
Srinivas Goud, D., 539
Sundeep Vijayaraghavan, 511
Suni, S. S., 475
Suresh, S., 73
Swati Shinde, 353

T
Tanmayee Samantaray, 115
Tanmay Kadam, 281
Tumwebaze, Pius, 711
Turriate-Guzman, Adriana Margarita, 27
Tushar Barai, 611

U
Utpal Bhattacharjee, 339

V
Vaishnavi V. Kulkarni, 581
Varun Vinod Kulkarni, 491
Venkatesh, N., 539
Venkateshwara Rao, M., 539
Vijaya B. Musande, 389
Villar-Quispe, Sofía Alessandra, 27
Vishal, S., 491
Vishwanath R. Hulipalled, 581

Y
Yılmaz, Gökçe Nur, 269

Z
Zameer Gulzar, 677
Zannat, Meftahul, 297