Information Systems Frontiers
https://doi.org/10.1007/s10796-023-10371-z

# Transfer Learning Enabled Bearing Fault Detection Methods Based on Image Representations of Single-Dimensional Signals

**Bilgin Umut Deveci[1] · Mert Celtikoglu[2] · Ozlem Albayrak[1] · Perin Unal[1] · Pinar Kirci[2]**

Accepted: 5 January 2023
© The Author(s) 2023

**Abstract**
Bearings are vital components in rotating machinery. Undetected bearing faults may result not only in financial loss, but also in the loss of lives. Hence, there exists an abundance of studies working on the early detection of bearing faults. The rising use of deep learning in recent years increased the number of imaging types/neural network architectures used for bearing fault classification, making it challenging to choose the most suitable 2-D imaging method and neural network. This study aims to address this challenge, by sharing the results of the training of eighteen imaging methods with four different networks using the same vibration data and training metrics. To further strengthen the results, the validation dataset size was taken as five times the training dataset size. The best results obtained is 99.89% accuracy by using Scattergram Filter Bank 1 as the image input, and ResNet-50 as the network for training. Prior to our work, Scattergram images have never been used for bearing fault classification. Ten out of 72 methods used in this work resulted in accuracies higher than 99.5%.

**Keywords** Rolling bearings fault determination · Transfer learning · Defect detection · CNN · Deep learning · GoogLeNet · ResNet-50 · SqueezeNet · Inception-ResNet-v2 · Signal processing · Time–frequency images · Scattergram

## 1 Introduction

Applications of deep learning and transfer learning are increasingly employed to solve important problems, and fault determination/classification of rolling bearings is no exception (Jiao et al., 2020; Zhang et al., 2019a). Bearings are essential elements of many industrial machines (Abdelkader et al., 2018). The widespread and frequent use of bearings makes them susceptible to deterioration, which may result in machine failures. Bearings are operational for extended periods in complex workload environments, and heavy workloads may result in undesirable deteriorations in the rolling bearings. Due to the key importance of bearing faults and the increasing use of deep learning methods and convolutional networks (CNN) on such problems, many studies conducted by both industry and academia aimed to classify defects (faults) in bearings (Han et al., 2020; Hendriks, 2021; Neupane & Seok, 2020; Niyongabo et al., 2022; Shao et al., 2018; Wang et al., 2019; Xu et al., 2021; Zhang et al., 2019a, 2020a, 2020b, 2020c). The use of these CNN techniques to detect a bearing fault has become an effective approach for the early diagnosis of a defect, potentially preventing catastrophic failure and reducing operating costs (Neupane & Seok, 2020).

To classify rolling bearing faults, this study proposes a method to determine the best composition of convolutional neural network (CNN) architecture and imaging method with the highest accuracy. To achieve this, we applied CNN and transfer learning, and followed the methodology defined in Section 3.

CNN's are a type of neural network composed of many layers, such as pooling layers, fully connected layers, and normalization layers. In other words, CNN's are known as a type of feature map. Thus, their properties always remain the same

✉ Bilgin Umut Deveci
deveci@teknopar.com.tr

Mert Celtikoglu
mrtceltikoglu@gmail.com

Ozlem Albayrak
albayrak@teknopar.com.tr

Perin Unal
punal@teknopar.com.tr

Pinar Kirci
pinarkirci@uludag.edu.tr

[1] Department of Research and Development, TEKNOPAR, Ankara, Turkey

[2] Department of Computer Engineering, Uludag University, Bursa, Turkey

Published online: 21 February 2023

within the same space. Adjusting the input properties within itself doesn't affect the properties (Naranjo-Torres et al., 2020).

The proposed method consists of four different open-source learning algorithms: GoogLeNet, ResNet-50, SqueezeNet, and Inception-Resnet v2. These algorithms were trained with millions of images to classify 1000 objects. Building an entire CNN architecture takes much time and effort. It is known that GoogLeNet, ResNet-50, SqueezeNet, and Inception-Resnet v2 perform very well on images, and are fine-tuned, therefore, they can be used directly with two-dimensional images, saving considerable time in the CNN design process. Transfer learning in CNN is defined as retraining previously trained neural networks on a new target. The study aims to detect bearing errors using these methods, using 2-D time–frequency images.

In order to overcome the difficulties experienced with all these engineering and signal processing experiences, Transfer Learning (TL) is widely used (Neupane & Seok, 2020). The TL method is currently seen in many machine learning applications, such as image recognition and text recognition (Pan & Yang, 2009). Its development was accelerated due to its application in such a widespread learning method, which uses existing knowledge, and also forms the source in the data processing area. For this reason, the TL method is commonly used in the processing of time–frequency images, as in the current study.

Vibration signals are expressed in highly descriptive two-dimensional images. In this study, hierarchical representations of vibration signals were created for use in deep learning algorithms. These various created vibration signals contain a large number of frequency and time–frequency images. The major advantage of CNNs on these images is that they can automatically detect hierarchical patterns and make decisions. In transfer learning method, CNN does not require expertise in signal processing and engineering, as it performs the operations automatically (Voulodimos et al., 2018).

This study aims to compare the most frequently used time–frequency images and convolutional neural networks used in bearing fault detection. Four different CNN architectures are selected based on their computational cost and performance. The motivations are to give insight into which time–frequency imaging methods visualize vibration features more distinctly, which CNN architecture distinguishes between different faults, as well as explore the replicability of the approach in industrial/real-world applications. With 72 different methods explored, this paper is one of the most comprehensive studies made on bearing fault classification problem. Another contribution of this study is using time–frequency imaging methods that are rarely/never used in the literature, and one of these imaging methods (Scattergram with Filter Bank 1) obtained the best accuracy out of 18 imaging methods.

The organization of the remaining of this paper is as follows: Section 2 provides a background of the study, and Section 3 presents the methodology used. Section 4, the results, summarizes the findings. In Section 5, discussion, the results are examined and discussed, and Section 6 concludes the paper, and presents possible future research recommendations.

## 2 Background

Failures of machines caused by faults in their major components have a significant impact on operational safety and system dependability in industrial production systems. Bearings are critical components of motors; it has been demonstrated that rolling bearing failures account for 30% of all failures in rotating machinery (Cerrada et al., 2018). Hence, several public datasets were published for addressing this problem. Deep learning-enabled bearing fault detection is a concept that is used increasingly frequently in recent years (Hakim et al., 2022). The input to the CNNs varies domains in such as time, frequency, and time–frequency. More in-depth reviews on the use of deep learning for bearing fault diagnostics can be found in Hakim et al. (2022); Tama et al., 2022; Cen et al., 2022; Liu et al., 2022). In Table 1, a general comparison of selected works and their important features are listed:

## 3 Overview of CNN Architectures

When detecting a fault in a bearing, sensors are placed in different parts of the machine to collect operating data, and these sensors process and determine the working procedure (Yin et al., 2014). The performance of the fault detection methods is determined by the quality of the signals collected, the effectiveness of signal processing, and feature extraction techniques (Neupane & Seok, 2020). Before the development of Industry 4.0, the maintenance of these types of machinery was performed only after a machine fault occurred. This type of post-failure maintenance approach often led to serious machine failures, resulting in material and non-material losses (Peng et al., 2002). For all these reasons, it is important for machine owners to monitor the machine's operating status and bearing health status to minimize losses. For detecting and diagnosing the faults in machinery and bearings, many signal processing approaches, machine learning (ML)-based approaches, and deep learning (DL)-based approaches have been proposed and implemented (Sit et al., 2020).

In the remaining part of this section, the CNN architectures and the imaging methods used in this study are summarized. These CNN architectures were selected according to the comparison in Fig. 1.

Figure 1 shows that SqueezeNet has a low prediction time, hence is less resource intensive and it has an overall

**Table 1** Comparison of the important features of the studies

| Study | General Information | | | |
|---|---|---|---|---|
| | Input | Dataset | Architecture | Accuracy Reached |
| Niyongabo, Zhang et al. (2022) | Time–frequency (CWT) | CWRU Bearing | CNN | 98.57% |
| Han, Tian, et al. (2020) | Time signal, Time Frequency | CWRU Bearing | LeNet-5 | 93.7% |
| Kumar, Anil, et al. (2020) | Time–Frequency | Own Dataset (Collected) | CNN | 96.8% |
| Zhang, Ying, et al. (2020a) | Time signal, Time- Frequency | Own Dataset (Collected) | CNN | 97.81% |
| Dhar, Priyadarshiny, et al. (2021) | Time–frequency | PCG Bearing database | AlexNet | 98.00% |
| Cao, Guannan, et al. (2020) | Time–frequency | CWRU Bearing | GoogLeNet | 99.5% |
| Neupane, Dhiraj, et al. (2020) | Time–frequency | CWRU Bearing | ResNet-50 | 99.68%—99.95% |
| S. Zhang, F. Ye, et al. (2019b) | Time signal | CWRU Bearing | CNN | 98.06% |
| Duan, Jian, et al. (2021) | Time–frequency | CWRU Bearing | CNN | 96.05% |
| J. Zhang, Y. Sun, L. Guo, et al. (2020c) | Time–frequency | Own Dataset (Collected) | DFCNN | 99.80% |
| Wen, Long, et al. (2019) | Time signal to RGB | CWRU Bearing | VGG-19 | 99.175% |
| Grover, Chhaya, and Neelam Turk (2022) | Bispectrum images | CWRU Bearing | ResNet-50 | 99.85% |
| Wei, Hao, et al. (2021) | Time–frequency (CWT) | CWRU Bearing | Residual Extreme Learning Machine (ResELM) | 99.90% |
| Liu, Chenyu, et al. (2021) | Time signal | CWRU Bearing | SE-ResNet-26 | 99.30% |
| Lee, Chun-Yao, and Truong-An Le. (2021) | Frequency (Persistence) | CWRU Bearing | ResNet structure (RCNN) | 99.625% |
| Li, Mingyong, et al. (2019b) | Time signal | CWRU Bearing | Sequential CNN | 96%-98% |
| Sun, Guodong, et al. (2021) | Time–frequency (MSSST) | CWRU Bearing | Faster Dictionary Learning | 99.08% |
| Sun, Guodong, et al. (2020) | Time–frequency (STMSST) | CWRU Bearing | LeNet-5 | 99.83% |
| Yuan, Laohu, et al. (2020) | Time–frequency (CWT) | CWRU Bearing | CNN-SVM | 98.75% |
| Minervini, Marcello, et al. (2021) | Time–frequency (spectrogram) | CWRU Bearing | CommandNet | 91.3% |
| Deveci, Çeltikoglu, et. al. (2021) | Time–frequency (spectrogram) | CWRU Bearing | ResNet-50 | 99.27% |
| Zhang, Huang, et al. (2022) | Time–frequency (spectrogram) | Machine Fault Prevention Technology (MFPT) Dataset | CNN | 99.63% |
| Yu, Y., et al. (2022) | Time–frequency (CWT, FFT) | MFPT Dataset | DASENet | 98.26% |
| Yu, Y., et al. (2022) | Time–frequency (CWT, FFT) | MFPT Dataset | MIMTNer | 97.80% |
| Yu, Y., et al. (2022) | Time–frequency (CWT, FFT) | MFPT Dataset | ResNet 18 | 96.68% |
| Yu, Y., et al. (2022) | Time–frequency (CWT, FFT) | MFPT Dataset | TCNN (FDS) | 93.86% |
| Yu, Y., et al. (2022) | Time–frequency (CWT, FFT) | MFPT Dataset | FTNN | 93.61% |
| Yu, Y., et al. (2022) | Time–frequency (CWT, FFT) | MFPT Dataset | SVM (FDS) | 90.44% |
| Wang, Zongyao, et al. (2022) | Time–frequency (CWT) | MFPT Dataset | GoogLeNet | 98.412% |
| Wang, Zongyao, et al. (2022) | Time–frequency (CWT) | MFPT Dataset | ResNet50 | 92.318% |
| Wang, Zongyao, et al. (2022) | Time–frequency (CWT) | MFPT Dataset | SqueezeNet | 97.51% |
| Wang, Zongyao, et al. (2022) | Time–frequency (CWT) | MFPT Dataset | IGoogLeNet | 99.4% |

better prediction accuracy compared to AlexNet. GoogLeNet is a network that uses Inception modules. ResNet50 is a pretrained network that uses residual architecture and performs quite well in its training time. Since a residual network is already being trained, ResNet18 was not used in this study. Inception-ResNet-v2 is a network that is resource intensive but has the second most accuracy (~ 80%) in pretrained networks that are available. To be able to compare networks that require multiple levels of resources was one of the objectives of this study. Networks like SqueezeNet and GoogLeNet can be trained and used in single-board computers such as Raspberry Pi, ResNet50 can be trained/used in modern laptops and Inception-ResNet-v2 can be used in workstations/servers. Hence, based on the needs of the project, different CNNs can be selected and used. Using different CNNs in this project also gives insight about into the performance

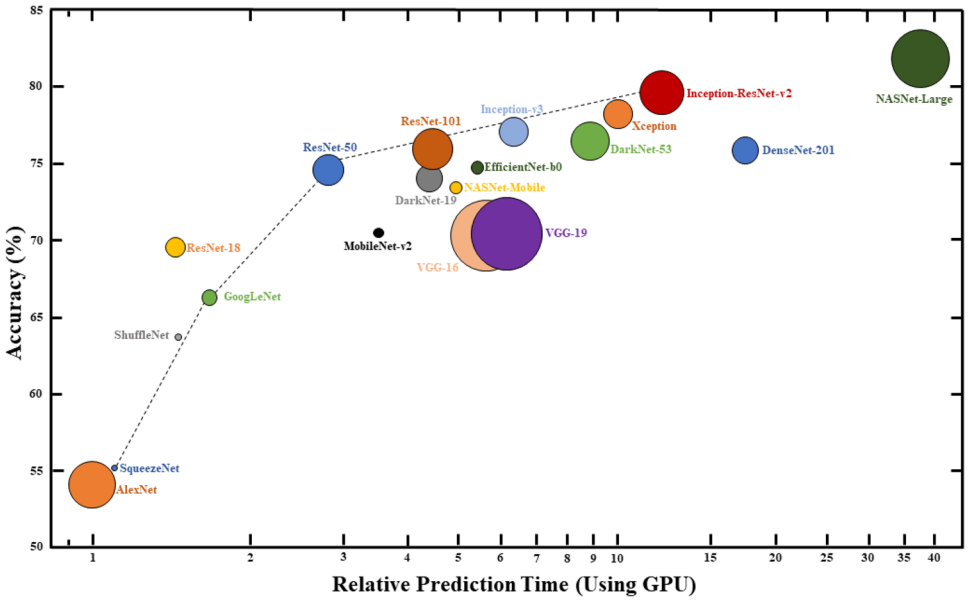



**Fig. 1** Comparison of pretrained networks – prediction time/accuracy (MathWorks Inc, 2022)

and reliability of different time–frequency imaging methods in bearing fault classification. If the imaging method separates the vibration signals clearly, the performance of the networks would not differ greatly.

## 3.1 CNN Architectures

### 3.1.1 GoogLeNet

GoogLeNet architecture was designed as a powerhouse with available calculational efficiency compared to its predecessors and similar contemporary networks (Cao et al., 2020).

One of the methods by which the GoogLeNet network achieves efficiency is through minimization of the input image, whilst simultaneously retaining important spatial information.

The first convolution layer in Fig. 2 uses a filter (patch) size of $7 \times 7$. This layer's primary purpose is to immediately minimize the input image without loss of spatial information, by utilizing large filter sizes.

The input image size (height and width) is reduced by a factor of four at the second convolution layer, and a factor of eight before reaching the first inception module, and, more feature maps are created.

The second conv layer has a depth of two and leverages the $1 \times 1$ conv block, which is the effect of dimensionality reduction. Dimensionality reduction through $1 \times 1$ conv block allows the decrease of calculational load by lessening the layers' number of operations.

The GoogLeNet architecture contains nine inception modules, as depicted in Fig. 3. There are two max-pooling layers between some of the inception modules, the purpose of which is to downsample the input. This is achieved through the reduction of the height and width of the input data.

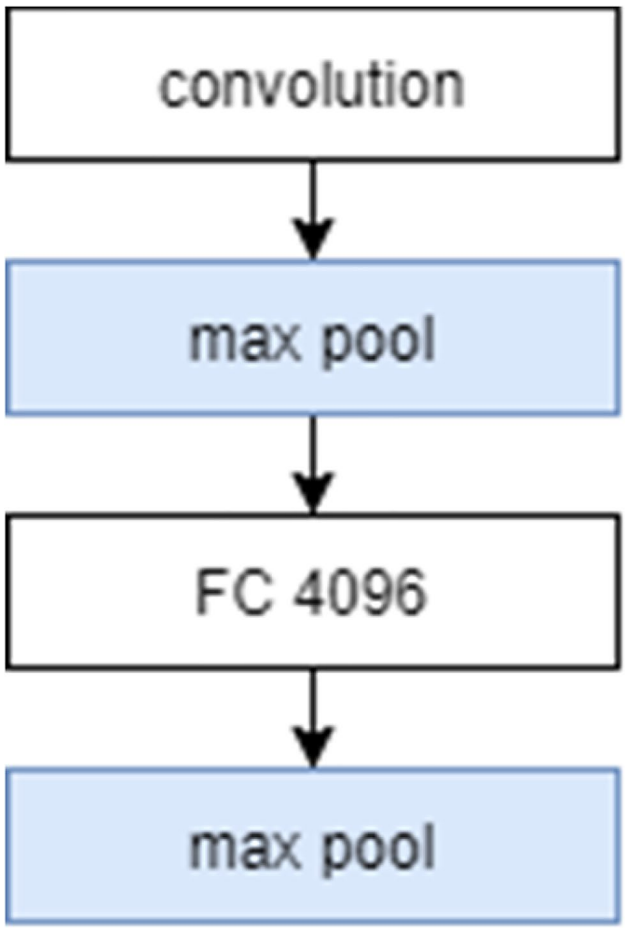



**Fig. 2** Convolution Layers (Cao et al., 2020)

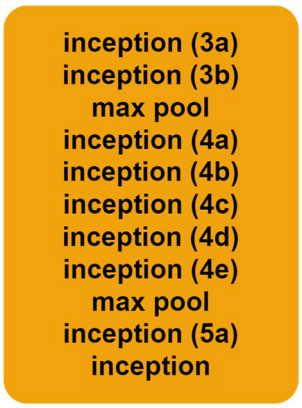



**Fig. 3** Partition of GoogLeNet (Cao et al., 2020)

by researchers at Microsoft Research, introduced a new architecture called Residual Network (He et al., 2016). ResNet-50 has 177 layers in total, and its architecture is given in Fig. 5 (Zagoruyko & Komodakis, 2016).

Figure 5 shows that, unlike GoogLeNet, it has no inception layers, and has a highway structure. This structure helps ResNet-50 extract features that are relatively deeper and more abstract. Hence, ResNet-50 has a good performance on image localization (Khankari et al., 2022). Since localization of certain patterns on time–frequency images is the most important goal in bearing fault classification, ResNet-50 was used in this study. ResNet-50 has 77 convolution layers (Peng et al., 2002).

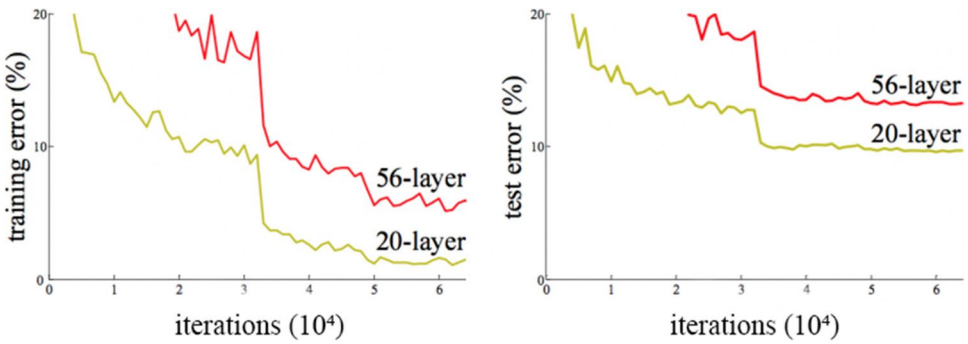



**Fig. 4** The effect of layer numbers in training error (Zagoruyko & Komodakis, 2016)

Reducing the input size between the inception module is another effective method of reducing the network's computational load. Transfer learning allows the utilization of a GoogLeNet network trained on ImageNet without the user's need to implement or train the network themselves]. A bottleneck of using GoogLeNet in bearing fault classification is that GoogLeNet drastically reduces the feature space between layers to reduce computational costs. This may lead to the loss of features that might have been useful in this application (Khan et al., 2020).

### 3.1.2 ResNet-50

In recent studies in the ImageNet field, each architecture uses more layers in a deep neural network to decrease the error rate (He et al., 2016). The number of layered architectures is therefore important. However, when the number of layers is increased, there is a problem in deep learning associated with the 'Vanishing/Exploding gradient'. This causes the gradient to become either 0, or too large. Thus, the training and test error rate increases with the number of layers (Liang, 2020) (Fig. 4).

When the graphs above are examined, it is noticed that the CNN architecture, with 56 layers, has more error repair than the 20-layer CNN architecture, in both training and test datasets (Zagoruyko & Komodakis, 2016). For this reason, there are considerable benefits in increasing layers in the ResNet architecture. ResNet, proposed in 2015

Again, to apply transfer learning with ResNet-50, the last fully connected layer (hidden in the softmax layer in Fig. 5) is changed to output ten machine health states (Shao et al., 2018).

ResNet-50 won the ILSVRC'14 competition (Zagoruyko & Komodakis, 2016). The top-5 error of various CNNs in this competition are given in Table 2.

### 3.1.3 SqueezeNet

SqueezeNet is a deep neural network for computer vision, released in 2016. SqueezeNet was developed by researchers at DeepScale, the University of California, Berkeley, and Stanford University. The aim was to create a smaller neural network with fewer parameters, which n more easily fit into computer memory, and this allows easier transmission over a computer network (Gaikwad & El-Sharkawy, 2018).

SqueezeNet Architecture**:**

Layers breakdown (Gaikwad & El-Sharkawy, 2018)

- layer 1: regular convolution layer
- layer 2–9: fire module (squeeze + expand layer)
- layer 10: regular convolution layer
- layer 11: softmax layer

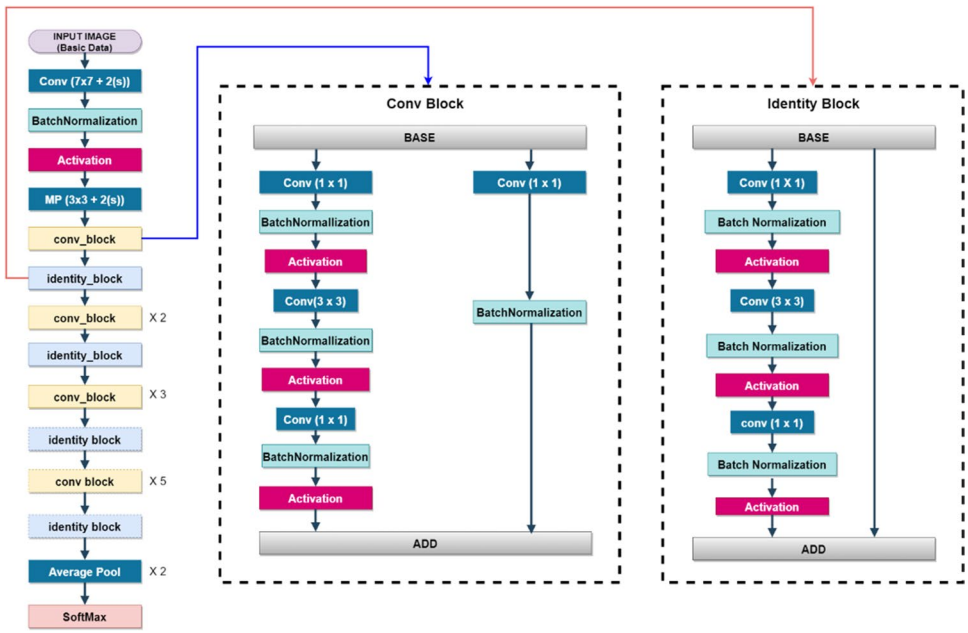



**Fig. 5** ResNet-50's architecture (Shao et al., 2018)

**Table 2** ResNet-50 error results (Zagoruyko & Komodakis, 2016)

| Method | Top-5 Error (Test Results) |
|---|---|
| VGG (Simonyan & Zisserman, 2014) (ILSVRC'14) | 7.32 |
| GoogLeNet (Szegedy et al., 2015) (ILSVRC'14) | 6.66 |
| VGG (Simonyan & Zisserman, 2014) (v5) (ILSVRC'14) | 6.8 |
| PreLU-Net (He et al., 2015)(ILSVRC'14) | 4.94 |
| BN-inception (Ioffe et al., 2015) | 4.82 |
| ResNet (ILSVRC'14) | 3.57 |

Architecture specifications (Gaikwad & El-Sharkawy, 2018)

- gradually increase the number of filters per fire module
- max-pooling with a stride of 2 after layers 1, 4, 8
- average-pooling after layer 10
- delayed downsampling with pooling layers

Surprisingly, this architecture lacks fully connected layers (Fig. 6). The difference of SqueezeNet is that typically, in a network like VGG, the later fully connected layers learn the relationships between the earlier higher-level features of a CNN and the classes that the network is aiming to identify. That is, the fully connected layers are those that learn that noses and ears make up a face, and wheels and lights indicate cars. However, in this architecture, this extra learning step is embedded within the transformations between various "fire modules". The authors of Gaikwad and El-Sharkawy (2018) derived inspiration for this idea from the Network in Network (NiN) architecture. Their SqueezeNet architecture was able to achieve a 50X reduction in model size compared to AlexNet while meeting or exceeding the top-1 and top-5 accuracy of AlexNet (Gaikwad & El-Sharkawy, 2018).

### 3.1.4 Inception ResNet-v2

Convolutional networks have been central to the largest advances in image recognition performance in recent years, and the achieved architecture has been shown to perform very effectively at a relatively low computational cost. The introduction of residual connections in conjunction with a more traditional architecture recently yielded state-of-the-art performance in the 2015 ILSVRC challenge; its performance was similar to the latest generation Inception-v4 network. There exists several variations of CNNs with Inception modules. To sum up, Inception-ResNet-v1 is a version of inception that has a similar computational cost to Inception-v3. Inception-ResNet-v2 is a more costly version of Inception with significantly improved recognition performance. Inception-v4 has a similar identification performance to Inception-ResNet-v2. But Inception-v4 is a purer version of Inception without residual connections. Inception-v4 has more computational cost compared to Inception-ResNet-v2 which is caused by the lack of residual connections. Inception-ResNet-v2 provides clear empirical evidence that training with residual connections significantly accelerates the training of Inception networks, and there is also some evidence of residual Inception networks slightly outperforming similarly expensive Inception networks

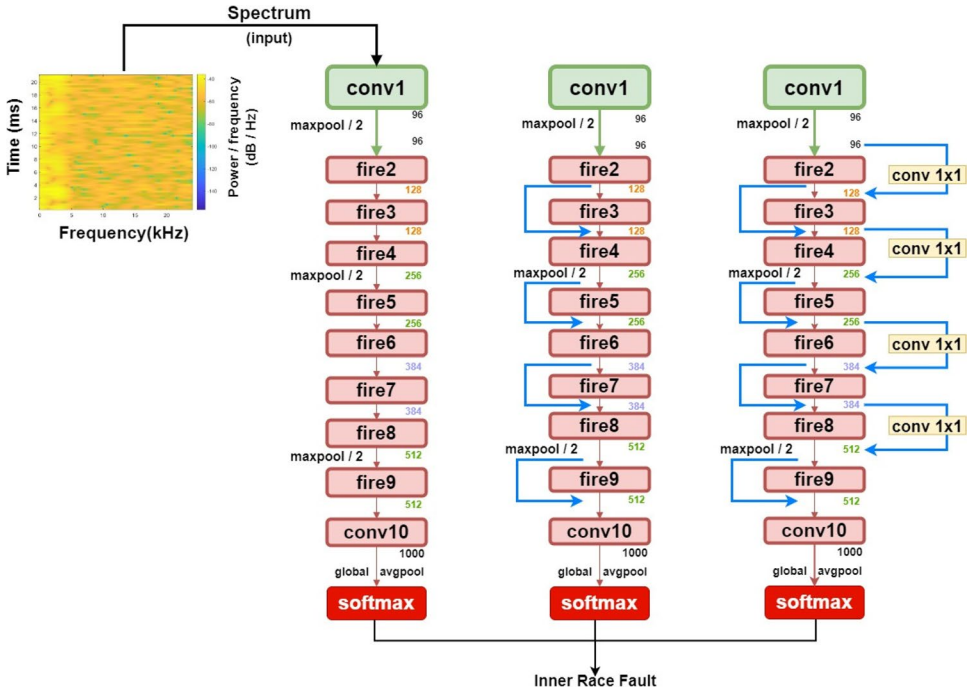



**Fig. 6** SqueezeNet (Left), SqueezeNet with simple bypass (Middle), SqueezeNet with complex bypass (Right) (Gaikwad & El-Sharkawy, 2018)

without residual connections. Inception-ResNet-v2 is a convolutional neural network that is trained on more than a million images from the ImageNet database.

At the top of the second Inception-ResNet-v2 in Fig. 7, the full network in its expanded form can be seen. This network is considerably deeper than the previous Inception V3. Figure 7 is an easier-to-read version of the same network where the repeated residual blocks have been compressed. The simplified inception blocks on Fig. 7 contain fewer parallel towers than the previous Inception-v3 (Baldassarre et al., 2017).

In addition, VGG-16/19 has a depth of 23/26 layers, compared to Inception ResNet-v2. VGG-16/19 performs worse accuracy overall and has more computational cost than Inception-ResNet-v2.

To detect bearing faults, eighteen different imaging methods were used in this study. The following (Section 3.1 and Section 3.2) is a brief introduction to all the methods used and their mathematical backgrounds. The visualization of these imaging methods is also shown.

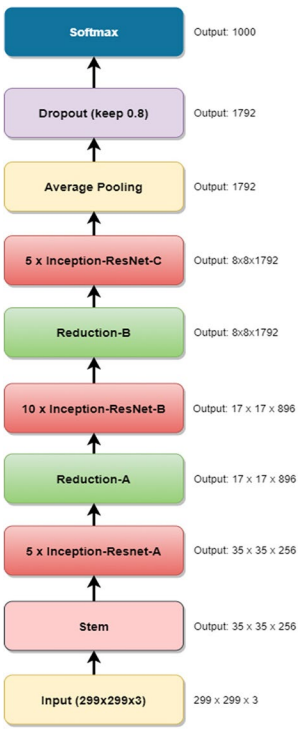



**Fig. 7** Inception ResNet-v2's architecture (Szegedy et al., 2017)

## 3.2 Time–Frequency Methods

Table 3 represents the order of time–frequency images in Section 3.2 of this study. After representing the mathematical formulae of each method, a visualization of the results was a need for easier replication. A unique contribution of this study is using vibration data regardless of the load that is on the motor. There are 10 labels to be predicted in this study, that contains normal and faulty signals with differing sizes, disregarding motor loads. Thus, one image from each label was represented on figures in the order specified. These figures also help to give an idea on which imaging methods output more detailed and continuous images, ensuring better results. These images were used as inputs to CNNs.

**Table 3** The order of time–frequency images used in this section

| | | |
|---|---|---|
| Normal Load 2 | Ball Fault 007 Load 0 | Ball Fault 014 Load 1 |
| Ball Fault 021 Load 3 | Inner Race Fault 007 Load 0 | Inner Race Fault 014 Load 1 |
| Inner Race Fault 021 Load 3 | Outer Race Fault 007 Load 0 | |
| Outer Race Fault 014 Load 1 | Outer Race Fault 021 Load 3 | |

#### 3.2.1 Spectrogram

A spectrogram provides a visual representation of the strength or amplitude height of a signal over time at various frequencies, which are presented in a given waveform. It is possible not only to identify whether there is more or less energy, for example, at 2 Hz vs 10 Hz but also, how energy levels vary over time. In other sciences, spectrograms are commonly used to display frequencies of sound waves produced by humans, machinery, animals, whales, jets, etc., as recorded by microphones (Pacific Northwest Seismic Network, 2022). Spectrograms help to distinguish and characterize the vibrations of signals obtained from machines in studies such as ours, and are increasingly used to examine the frequency content, mainly of continuously acquired signals. In Fig. 8 and Fig. 9, spectrogram images for each bearing condition are displayed for the unthreshold and the thresholded condition.

#### 3.2.2 Continuous Wavelet Transform (CWT)

The Continuous Wavelet Transform (CWT) is used to decompose a signal into wavelets, i.e., small oscillations that are highly localized in time. The Fourier Transform decomposes a signal into infinite-length sines and cosines, effectively losing all time-localization information; in contrast CWT's basis functions are scaled and shifted versions of the time-localized mother wavelet. The CWT is used to construct a time–frequency representation of a signal, which provides highly accurate time and frequency localization. (Büssow, 2007).

The CWT is an excellent tool for mapping the changing properties of non-stationary signals and also for determining whether a signal is stationary in a global sense. When a signal is judged non-stationary, the CWT can be used to identify stationary sections of the data stream. (Büssow, 2007).

The core computations of CWT procedures generally follow the Torrence and Compo algorithms as listed in the references section below. The nomenclature used in the equations below were also those used in the Torrence and Compo paper. (Büssow, 2007).

The mathematical computation of the Continuous Wavelet Transform is given below:

In this approach, a time series signal is built up from the members of generalized Morse wavelets. Morse wavelets are represented as $\psi_{\beta,\gamma}$, where $\beta$ denotes the Fourier-domain bandwidth of the wavelet and γ indicates the shape of the wavelet (Youngberg & Boll, 1978). Morse wavelets in time are defined in frequency domain as (1) (Lilly, 2017):

$$\psi_{\beta,\gamma}(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} \Psi_{\beta,\gamma}(\omega)e^{i\omega t}d\omega, \Psi_{\beta,\gamma}(\omega) = \alpha_{\beta,\gamma}\omega^{\beta}e^{-\omega^{\gamma}} \times \begin{cases} 1, & \omega > 0 \\ 1/2, & \omega = 0 \\ 0, & \omega < 0 \end{cases} \tag{1}$$

$\omega$ is the frequency in radians, $\alpha_{\beta,\gamma}$ is a constant value that normalizes the wavelet in frequency domain. $\alpha_{\beta,\gamma}$ is estimated with (2).

$$\alpha_{\beta,\gamma} \equiv 2\left(\frac{e\gamma}{\beta}\right)^{\beta/\gamma} \tag{2}$$

Finally, the continuous wavelet transform of a signal x(t) is computed with (3). (Lilly & Olhede, 2008)

$$CWT(t,\omega) \equiv \omega_{\beta,\gamma}(\tau,s) \equiv \int_{-\infty}^{\infty} \frac{1}{s}\psi^{*}_{\beta,\gamma}\left(\frac{t-\tau}{s}\right)x(t)dt = \frac{1}{2\pi}\int_{-\infty}^{\infty} e^{i\omega t}\Psi^{*}_{\beta,\gamma}(s\omega)X(\omega)d\omega \tag{3}$$

where $X(\omega)$ is the Fourier transform of x(t), the asterisk is conjugate operation, the scale variable s denotes the compression/stretching of the wavelet in time domain. Rescaled frequency domain wavelet $\Psi^{*}_{\beta,\gamma}(s\omega)$ has a global maximum where $\omega_s = \omega_{\beta,\gamma}/s$, which is the scale frequency.

In Fig. 10, the CWT images for each bearing condition is shown:

#### 3.2.3 Wavelet-based Synchrosqueezing Transform (WSST)

The wavelet synchrosqueezed transform is a time–frequency analysis method that can be used for analyzing multicomponent signals with oscillating modes, such as speech waveforms, machine vibrations, and physiologic signals. Many of these real-world signals with oscillating modes can be written as a sum of amplitude-modulated and frequency-modulated components. A general expression for these types of signals with summed components is (4).

$$\sum_{k=1}^{K} A_k(t)cos\left(2\pi\phi_k(t)\right) \tag{4}$$

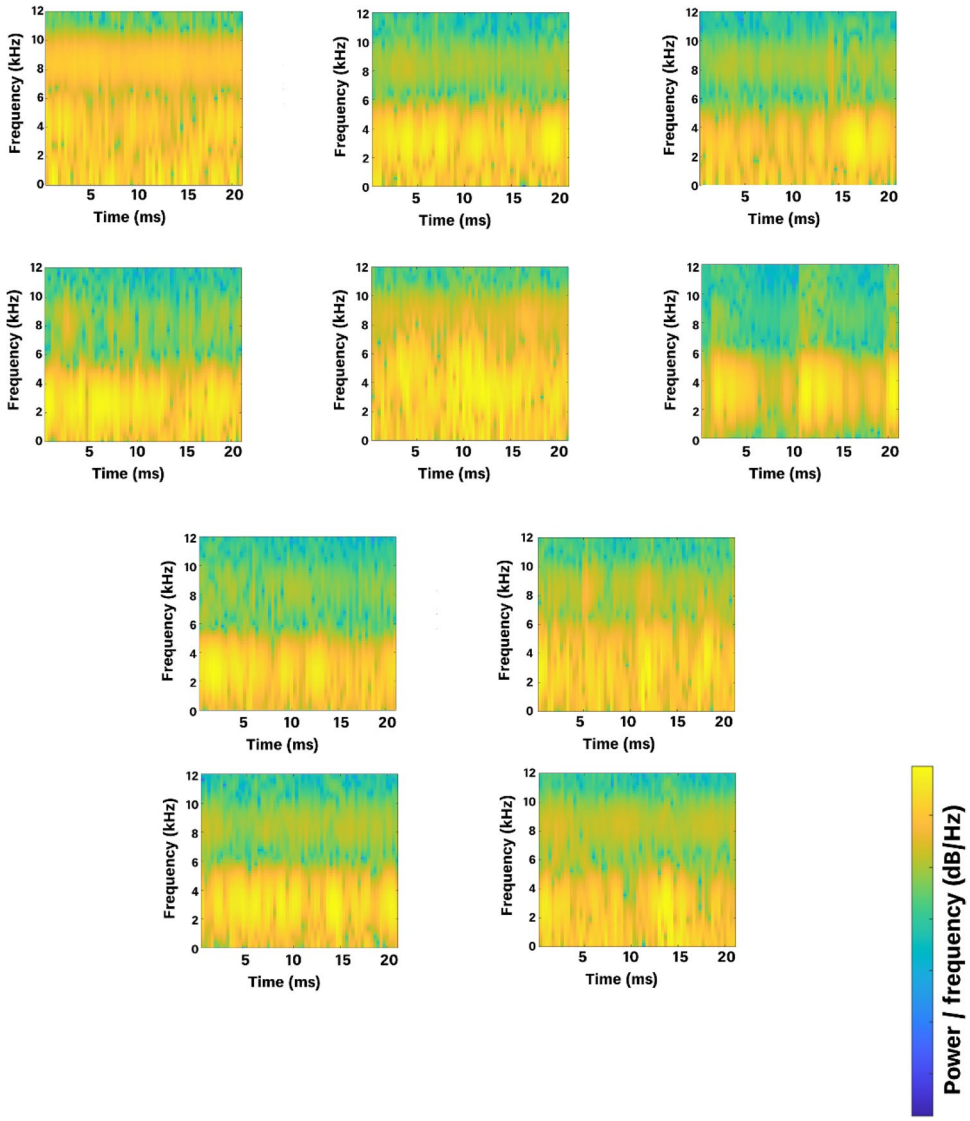



**Fig. 8** Spectrogram images for each bearing condition (unthresholded)

where $A_k(t)$ is the slowly varying amplitude and $\phi_k(t)$ is the instantaneous phase. A truncated Fourier series, in which there is no variation in where the amplitude and frequency do not vary with time, is a special case of these signals.

The wavelet transform and other linear time–frequency analysis methods decompose these signals into their components by correlating the signal with a dictionary of time–frequency atoms (Mallat, 1999). The wavelet transforms use translated and scaled versions of a mother wavelet as its time–frequency atom. These time–frequency atoms are associated with a degree of some time–frequency spreading, that is associated with all of these time–frequency atoms, which weakens and affects the sharpness of the signal analysis.

The wavelet synchrosqueezed transform is a time–frequency method that reassigns the signal energy in frequency in order to compensate for the spreading effects of the mother wavelet. Unlike other time–frequency reassignment methods, synchrosqueezing reassigns the energy only in the frequency direction, thus preserving the signal's time resolution. By preserving the time, the inverse synchrosqueezing algorithm can reconstruct an accurate representation of the original signal. To enable its use, each term in the summed components signal expression must be an intrinsic mode type (IMT) function (Daubechies et al., 2011).

In Fig. 11, the WSST images for each bearing condition are displayed.

#### 3.2.4 Fourier-based Synchrosqueezing Transform (FSST)

When the Fourier transform and the Fourier-based Synchrosqueezing Transform are considered separately, the

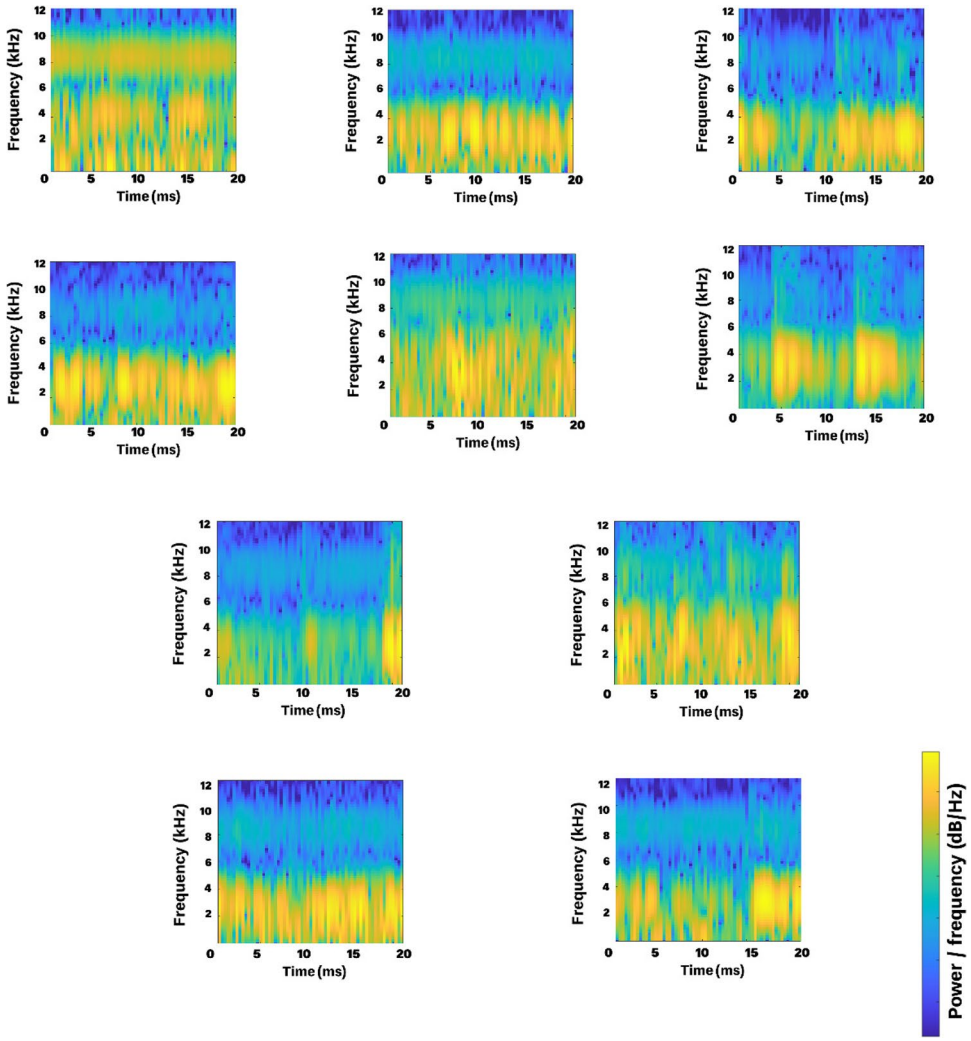



**Fig. 9** Spectrogram images for each bearing condition (thresholded)

Fourier transform (FT) is intrinsically related to linear time-invariant filters which are often used to model the response of physical systems or sensors (Oberlin et al., 2014). Synchronous-compressed transform, on the other hand, is a time–frequency analysis method which can be used for analyzing multicomponent signals with oscillating modes (Daubechies et al., 2011). The main difference is that SST is 2-way. The first method is the representation of multicomponent signals in the TF plane, and the second is a decomposition method enabling the separation and demodulation of the different modes (Oberlin et al., 2014).

Synchrosqueezing transform (SST) is a method based on a modified Short-Time Fourier Transform (STFT) (Kong & Koivunen, 2019). SST was originally based on Continuous Wavelet Transform, Fourier-based Synchrosqueezing Transform was built upon CWT based Synchrosqueezing Transform (WSST) (Rueda & Krishnan, 2019).

The mathematical computation of Fourier based synchrosqueezing is given in Eqs. 5–7.

Firstly, the modified STFT of a signal x(t) is calculated as (5) (Abdelkader et al., 2018).

$$STFT(\tau,\eta) = \int \omega(t-\tau)x(t)e^{-i\eta t}dt \tag{5}$$

where $\eta$ is the frequency index, $\omega(t)$ is the window function used for localization. The instantaneous frequency for each index η is calculated with (6) (Oberlin et al., 2014).

$$\omega_s(\tau,\eta) = -i(STFT(\tau,\eta))^{-1}\frac{\partial STFT(\tau,\eta)}{\partial\eta} \tag{6}$$

Finally, the synchrosqueezing of STFT (FSST) transform is found with (7).

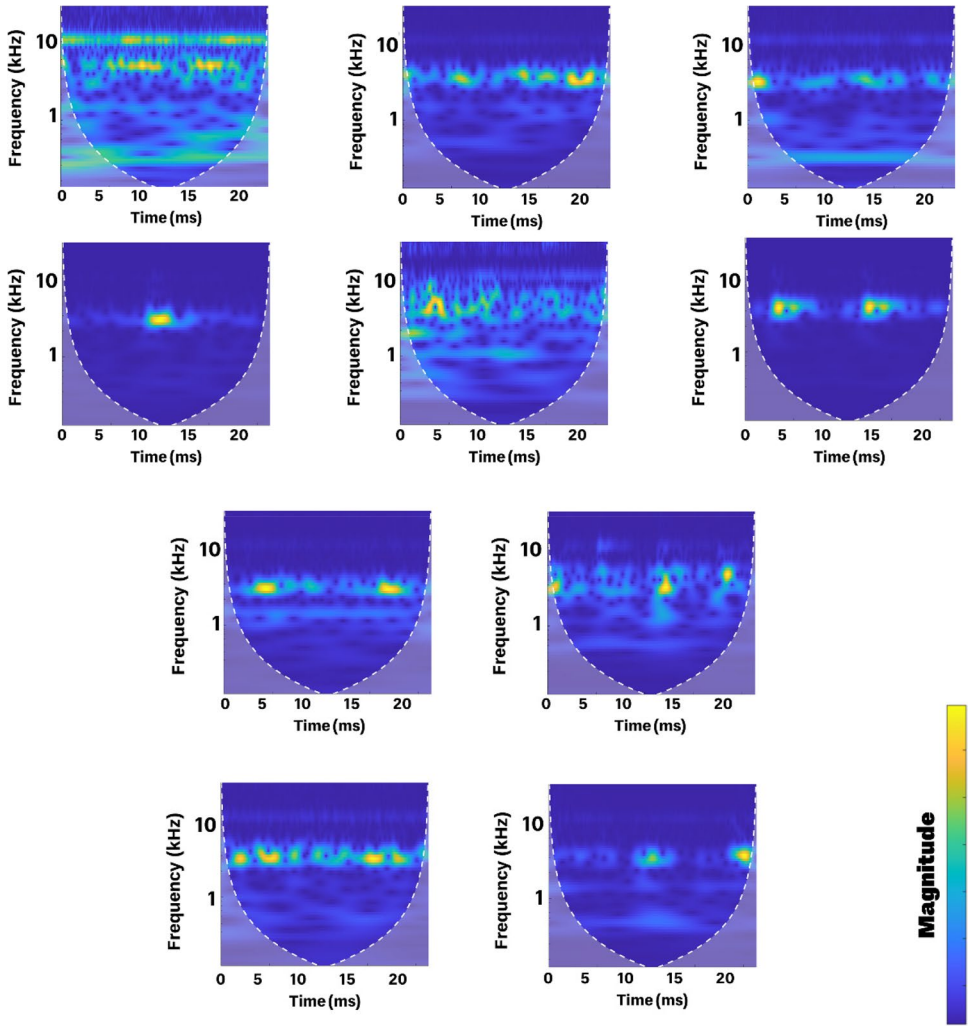



**Fig. 10** CWT images for each bearing condition

$$FSST(t,\omega) = \frac{1}{\omega(0)} \int \delta\big(\omega - \omega_s(t,\eta)\big) STFT(t,\eta) d\eta \tag{7}$$

In Fig. 12, the FSST images for each bearing condition are present.

### 3.2.5 Wigner-Ville Distribution (WVD)

The Wigner-Ville Distribution is an important and widely used time–frequency method (Kutz, 2013). In the past, it was used mainly for object detection and imaging (Barbarossa & Farina, 1992), but with the rise of deep learning, it is increasingly used for fault detection purposes (Singru et al., 2018). WVD provides great time–frequency resolution (Li & Zheng, 2008). The Wigner-Ville Distribution is defined as (8).

$$\mathcal{W}_{f,g}(t,\omega) = \int_{-\infty}^{\infty} x(t+\tau/2)\overline{g}(t-\tau/2)e^{-iw}\tau d\tau \tag{8}$$

where $\mathcal{W}_{f,g}(t,\omega)$ is the kernel that transforms the function $x(t+\tau/2)\overline{g}(t-\tau/2)$ (Kutz, 2013). $\tau$ denotes the shift of the signal.

In MATLAB, the Wigner-Ville Distribution of a signal can easily be obtained by calling "wvd (signal,sampling_rate)" function.

In Fig. 13, the Wigner-Ville Distribution of each bearing condition is shown.

### 3.2.6 Constant-Q Nonstationary Gabor Transform (CQT)

The Constant-Q Nonstationary Gabor Transform (CQT) is a time–frequency representation generally used for nonlinear and nonstationary signals. The Gabor transform is a well-known Spectrogram, also known as Short Time Fourier Transform (STFT). CQT is a specialized version of Spectrogram designed for audio processing introduced in 1978 by J.Youngberg and S. Boll (1978). The transform is determined by a Q factor, i.e., the ratio of center frequency to bandwidth of each window. Constant Q factor results in finer

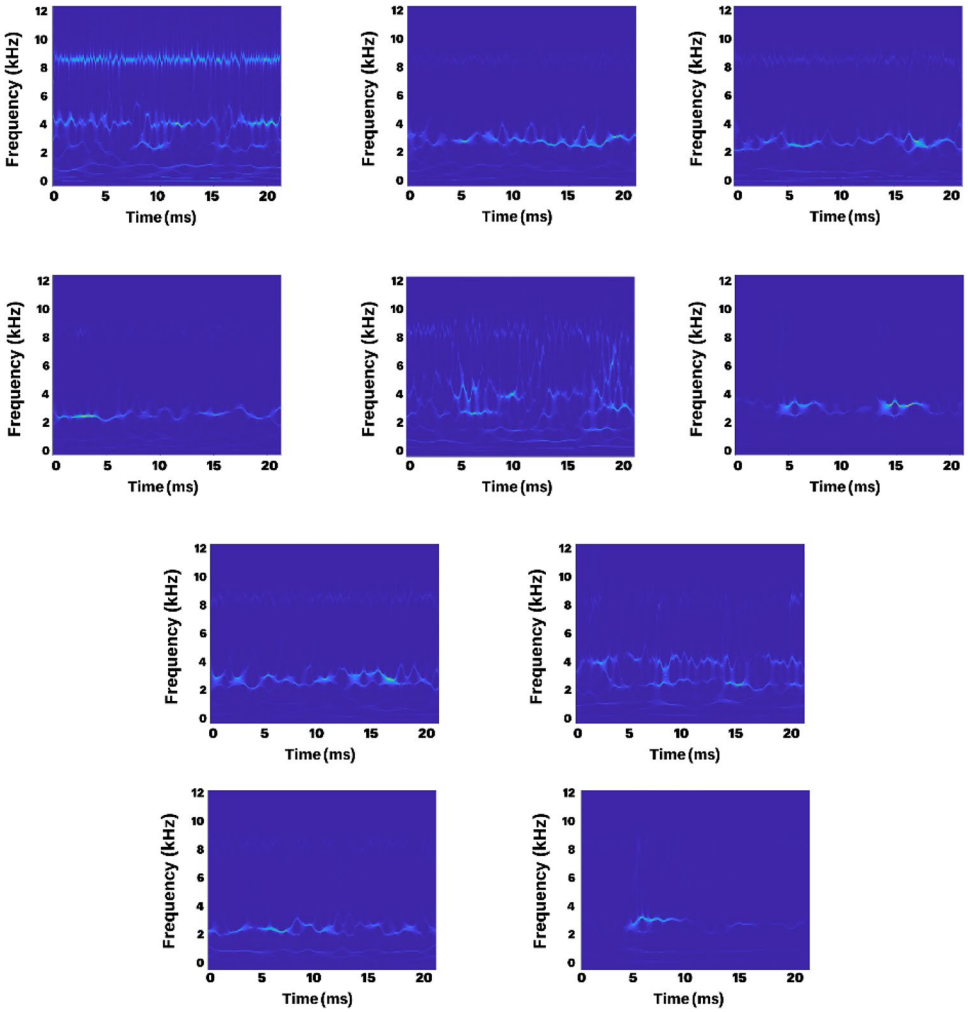



**Fig. 11** WSST images for each bearing condition

low frequency resolution, and the time resolution increases with frequency.

The mathematical background of CQT is specified in Eq. 9–11 (Schörkhuber et al., 2014)

$$a_w(k) = g_w(k)e^{i2\pi k f_w/f_s}, k \in \mathbb{Z} \tag{9}$$

$$f_w/f_s = Q\ Factor \tag{10}$$

where $g_w(k)$ is the zero centered window function (usually Gaussian), the bin center frequency is $f_w$ and the sampling rate is $f_s$. After obtaining the conjugate of $a_w(k)$, CQT is calculated with stated sum block:

$$CQT(t,w) = \sum_{k=0}^{K} x(k)a_w^*(k-t) \tag{11}$$

In MATLAB, the CQT transform is perfectly reconstructable, meaning that the signal is retractable after the transformation.

In Fig. 14, the CQT of each bearing condition is shown.

### 3.2.7 Instantaneous Frequency

Instantaneous frequency is a method mainly used in seismic, radar, communications and biomedicals (Boashash, 1992). Designed for the visualization of nonstationary signals in a 2-D representation, it is suitable for bearing fault classification. This method draws a time-varying line that represents the average of frequency components on a signal.

To draw the line, it first computes the power spectrum of the signal using pspectrum() function, that results in $P(t,f)$. Then, it estimates the time-varying line using time–frequency moment, which is the Eq. (12) (MathWorks, 2022):

$$\boldsymbol{IF(t)} = \frac{\int_0^\infty \boldsymbol{fP(t,f)df}}{\int_0^\infty \boldsymbol{P(t,f)df}} \tag{12}$$

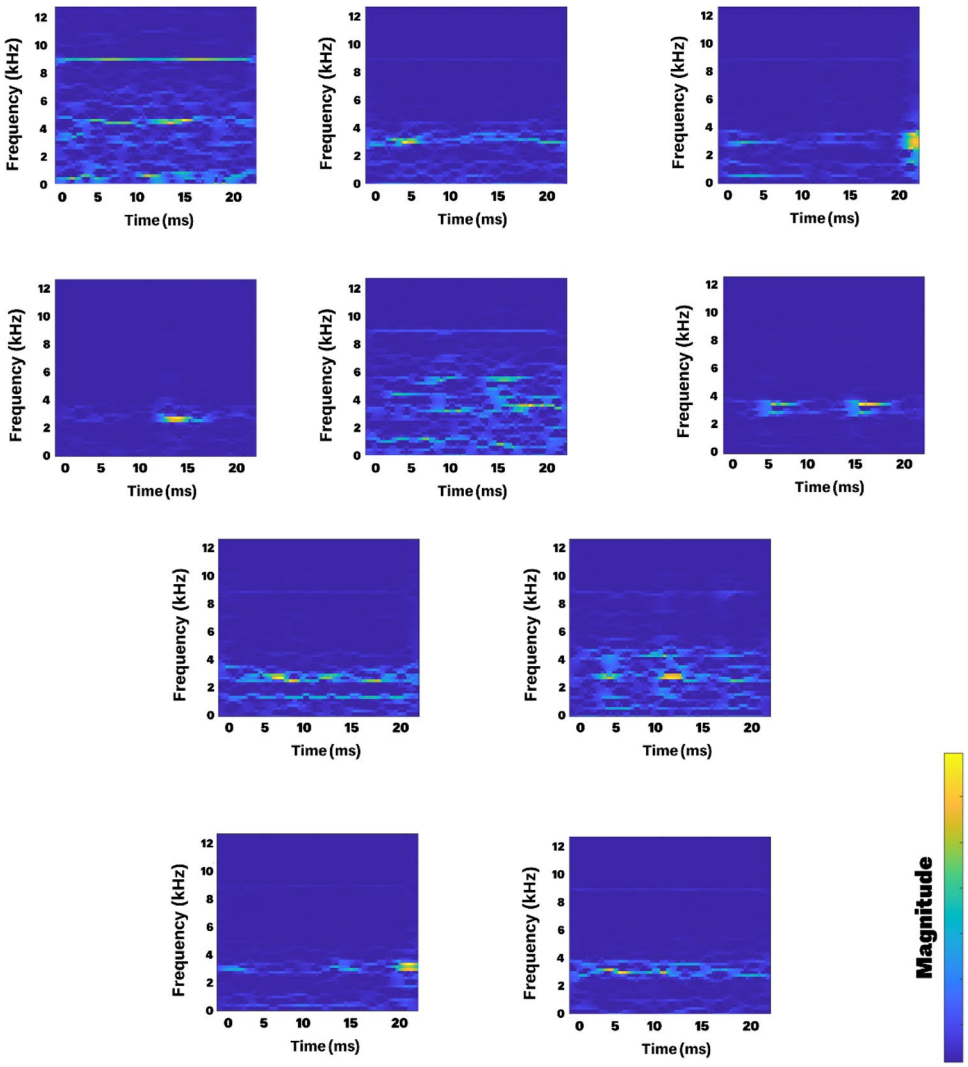



**Fig. 12** FSST images for each bearing condition

After calculating the time-varying instantaneous frequency $IF(t)$, MATLAB positions this newly created signal on a time–frequency representation, simultaneously showing the characteristics of the signal and the instantaneous frequency on a plot.

The instantaneous frequency image for each bearing condition is shown in Fig. 15 the red line is the instantaneous frequency, and the background is a time–frequency representation.

### 3.2.8 Hilbert Huang Transform (HHT)

Data gathered from accelerometers are nonlinear and non-stationary. Hilbert-Huang Transform aims to represent nonlinear and non-stationary data (Huang, 2014), and is widely used in fault classification. Compared to other methods, this method is relatively recent (1998–1999), but nevertheless, is featured in many articles focusing on non-stationary and nonlinear signals (Huang, 2014).

Hilbert-Huang Transform consists of two steps. The first step (and the key part) is Empirical Mode Decomposition (EMD). Briefly stated, EMD extracts Intrinsic Mode Functions (IMF) from the signal, using convolution. EMD satisfies two criterions (Huang, 2014):

- Number of extrema is equal to the number of zero crossings or differs from it by not more than one.
- Mean of envelopes (upper-lower) approach zero.

These criteria reduce the frequency of riding waves and smooth uneven amplitudes (smoothing outliers, normalizing).

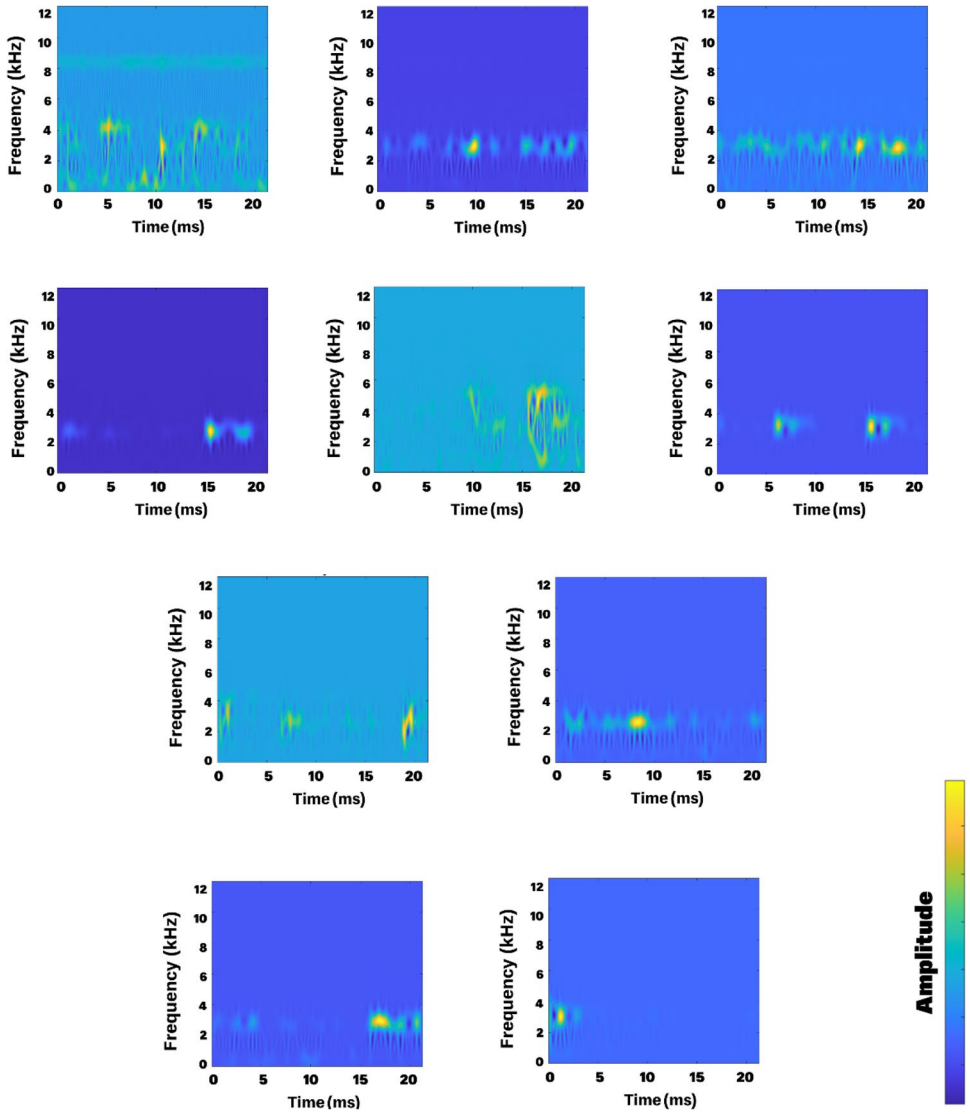



**Fig. 13** WVD images for each bearing condition

After computing the IMFs of the signal using EMD, the Hilbert transform is applied on the EMD. The mathematical computation of Hilbert-Huang Transform is presented by the Eqs. 13–16.

$$y_i(t) = \frac{1}{\pi} P \int_{-\infty}^{\infty} \frac{x_i(\tau)}{t-\tau} d\tau \tag{13}$$

$$z_i(t) = x_i(t) + \rangle y_i(t) = a_i(t) e^{\rangle \theta_i(t)} \tag{14}$$

$$a_i(t) = \left| x_i(t) + \rangle y_i(t) \right|, \theta_i(t) = \tan^{-1}\left(\frac{y_i(t)}{x_i(t)}\right) \tag{15}$$

$$\textit{The frequency component of } HHT(t,w) = w_i(t) = \frac{d\theta_i(t)}{dt} \tag{16}$$

where $P$ is the Cauchy principal value, $x_i(\tau)$ is the shifted EMD output, $y_i(t)$ is the result of Hilbert Transform. Forming a complex conjugate pair concludes the Hilbert-Huang Transform.

In Fig. 16, the Hilbert-Huang Transform of each bearing condition is visible.

The images of HHT, between 30–100 in y-axis show minimal information present. Therefore, to improve resolution, and thus, accuracy, the y-axis was limited between 0–30. The resulting HHT images are presented in Fig. 17.

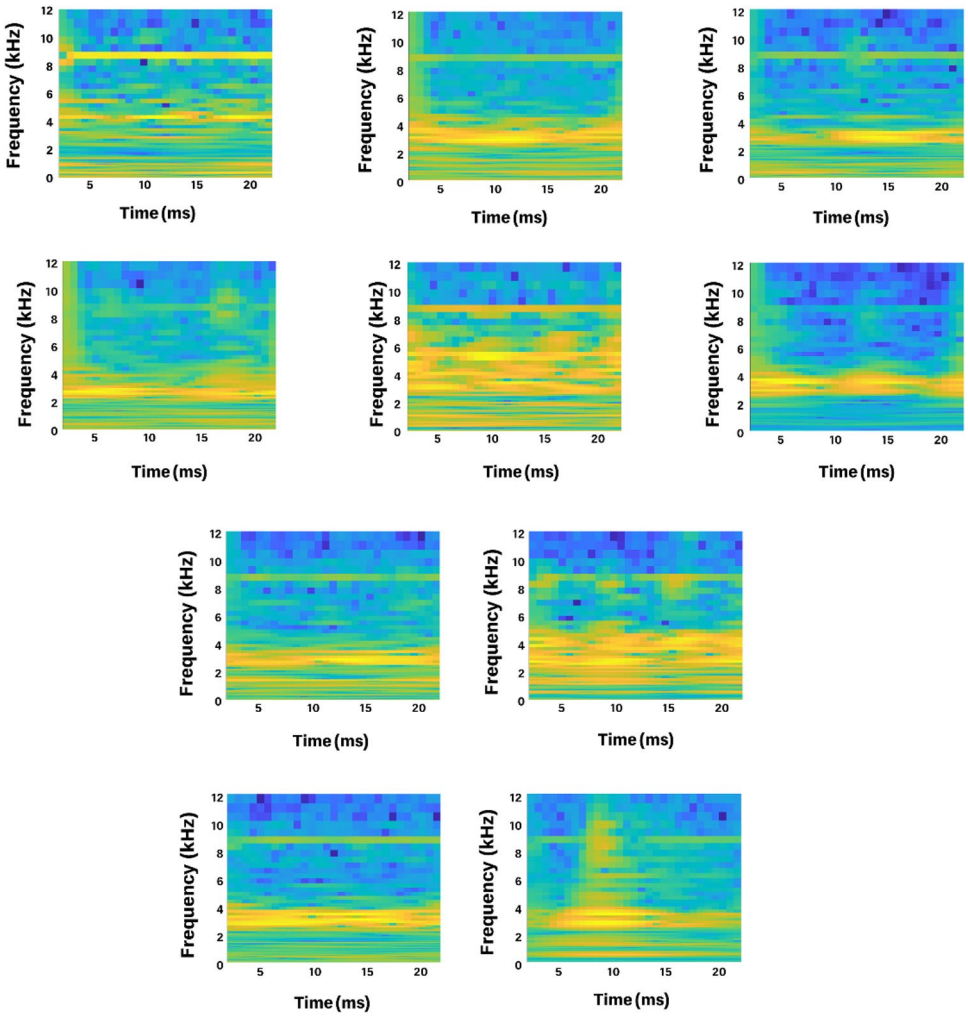



**Fig. 14** CQT images for each bearing condition

### 3.2.9 Scattergram

The scattergram is a method of visualizing (scattering) spectral coefficients. Scattering transforms are computed with different wavelets, resulting in better frequency resolution (Andén & Mallat, 2014). This is the most recent of the methods used in this study (2012), introduced in Mallat (2012). Although not originally intended to be used as transfer learning input for bearing fault classification, scattergrams have proven useful for image classification (Andén & Mallat, 2014).

Scattergram calculates multi-order spectrum coefficients by recursive iterations of convolution and modulus operations. The modulus operation, which assures nonlinearity, is put into effect after convolution with $\psi_{\lambda_1}, \psi_{\lambda_2}, \psi_{\lambda_3}, ..., \psi_{\lambda_k}$ wavelets. The output of these operations then convolves with a low pass filter $\phi(t)$, which is used for averaging. Different low-pass filters output different time–frequency images; therefore, it is necessary to use two separate filters. A visualization of scattering transform procedures is shown in Fig. 18.

When the 'FilterBank' property is assigned to 1 in MATLAB, the resulting scattergram images of each bearing condition is on Fig. 19.

To be able to change averaging by using another $\phi(t)$ and obtain different images, the 'FilterBank' property is assigned to 2 in MATLAB, the resulting scattergram images of each bearing condition is on Fig. 20.

When the lowpass filter $\phi(t)$ is changed, the averaging, thus the scattergrams are observed to differ dramatically.

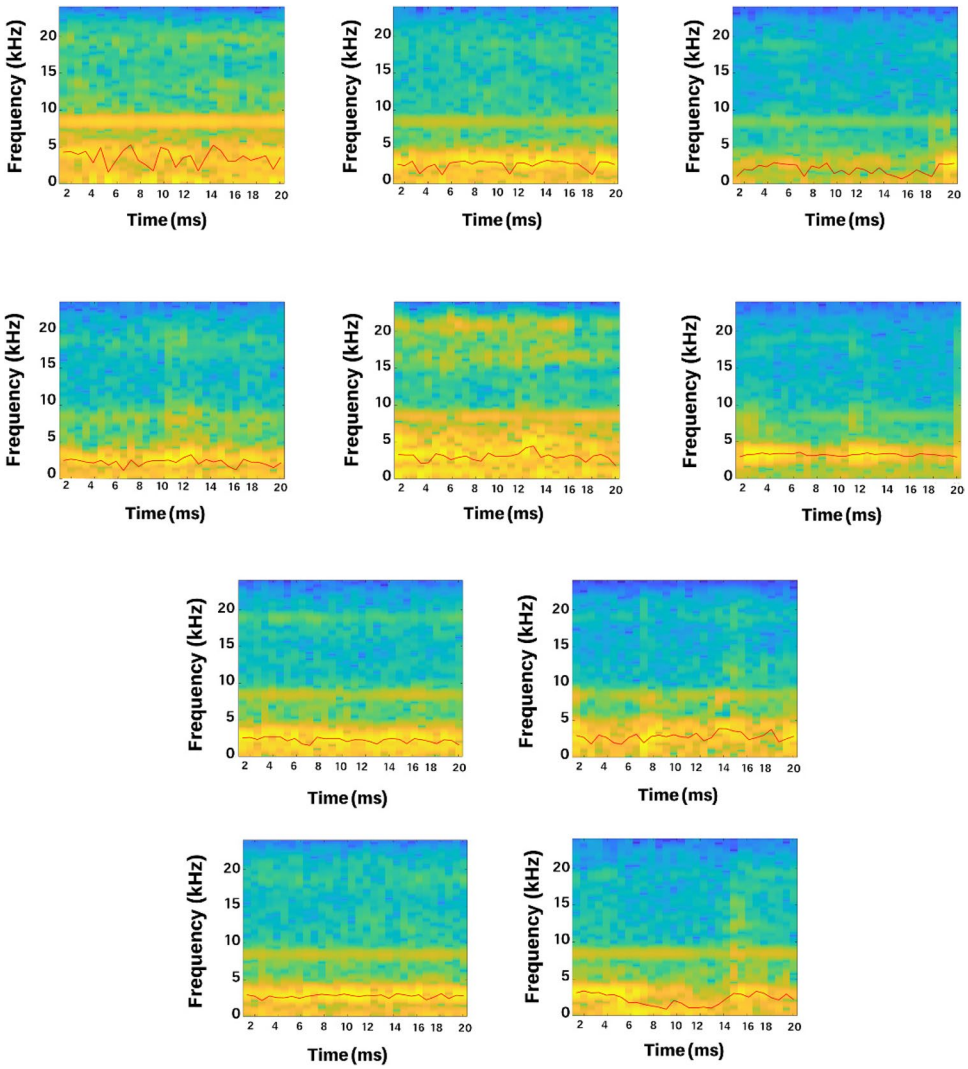



**Fig. 15** Instantaneous frequency images for each bearing condition

## 3.3 Frequency Methods

### 3.3.1 Power

MATLAB's pspectrum() function calculates the time-localized power spectrum of any time series signal (Bossau, 2020) by automatically choosing the optimum window size, window overlap and window type (Abed et al., 2012). Power spectrum is the square of FFT transformation. The y axis denotes the power in dB, and the x axis is frequency in Hertz. This is a frequency image, since there is no time element present in this method. The power spectrum is calculated as:

$$Power\ Spectrum = |FFT(x(t))|^2 \tag{17}$$

where $x(t)$ is the time series signal.

The power spectrum images for each bearing condition is present in Fig. 21.

### 3.3.2 Persistence

Persistence is very similar in nature to power spectrum, as discussed in 5.2.1. The persistence spectrum is not widely used in bearing fault classification; of the few studies in this area, the most significant is (Lee & Le, 2021). In persistence spectrum, the signal is further divided into smaller, overlapping sections. The summing of power spectrum of smaller in size signals results in the persistence spectrum. In Fig. 22, the whole process is visualized (Lee & Le, 2021).

As can be seen from Fig. 22, the original signal (of length 1024) is windowed again. The power spectrum of these

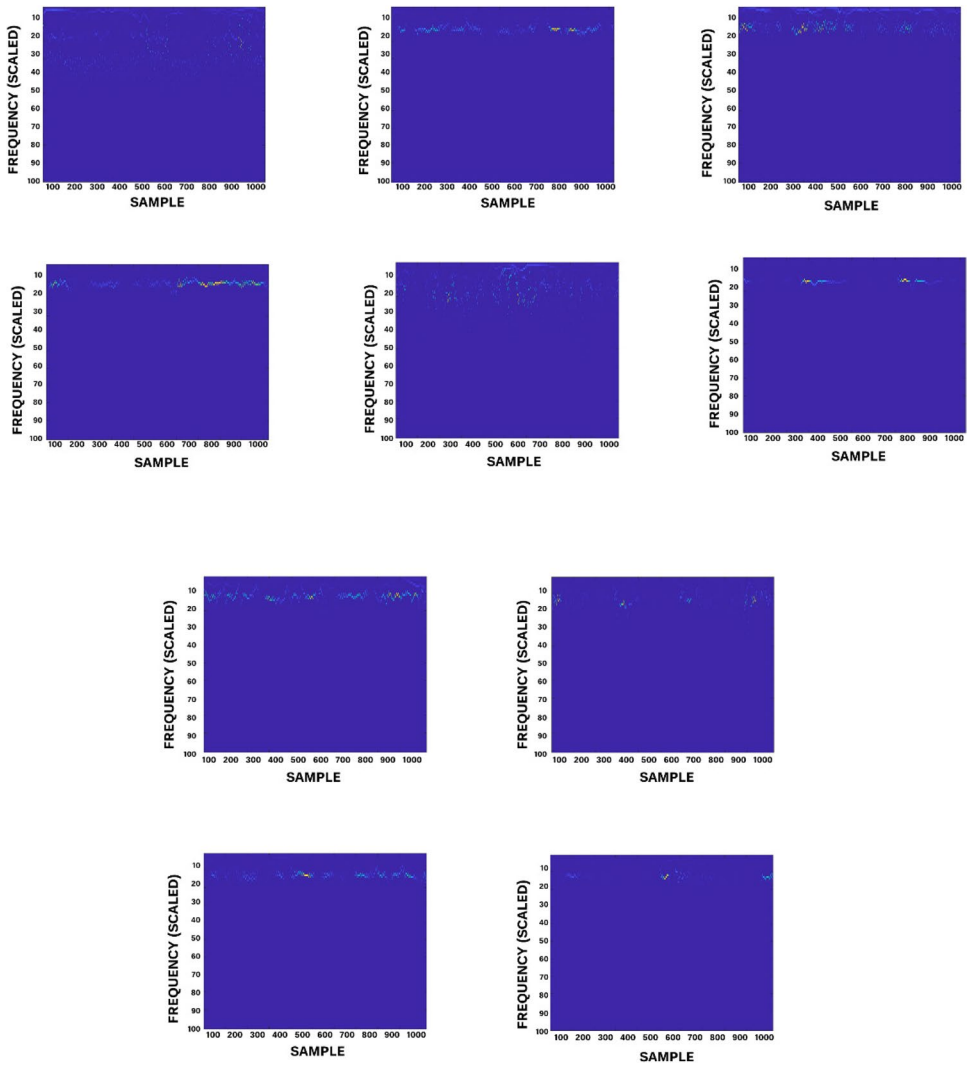



**Fig. 16** HHT images for each bearing condition (not cropped)

windowed signals is scattered into various parts, and then the sums of the power spectrum are shown on a 2-D image. In this way, the "persistence" of the signals on the frequency domain are clearly visualized.

The resulting persistence spectrums of each bearing condition is shown in Fig. 23.

### 3.3.3 Spectral Kurtosis

Spectral kurtosis (SK) is a statistical tool that can indicate the presence of a series of transients and their locations in the frequency domain. This creates a valuable supplement to the classical power spectral density, completely eradicating non-stationary information (Antoni, 2006). The resulting spectral kurtosis of each bearing condition is shown in Fig. 24.

### 3.3.4 Kurtogram

The Kurtogram allows the visualization of e Spectral Kurtosis (SK) in a frequency-frequency resolution image. Spectral Kurtosis, discussed later in this paper, is a way to distinguish transients in a signal. There are various ways to visualize kurtograms.

In the first approach, uniform filter banks are used. The signal is passed through bandpass filters in frequency, which allows certain set frequencies to be amplified. Data obtained on these frequencies are sent through low pass filters, and then downsampled. Many filters are involved, and this method is computationally expensive (Huang et al., 2018).

The second approach is to use binary tree filter banks. Using a combination of lowpass and highpass filters (bandpass filter), the signals are decomposed into frequency bands. With level k, there are $2^k$ differing frequency bands. This operation only requires two filters for each area, thus MATLAB's kurtogram()

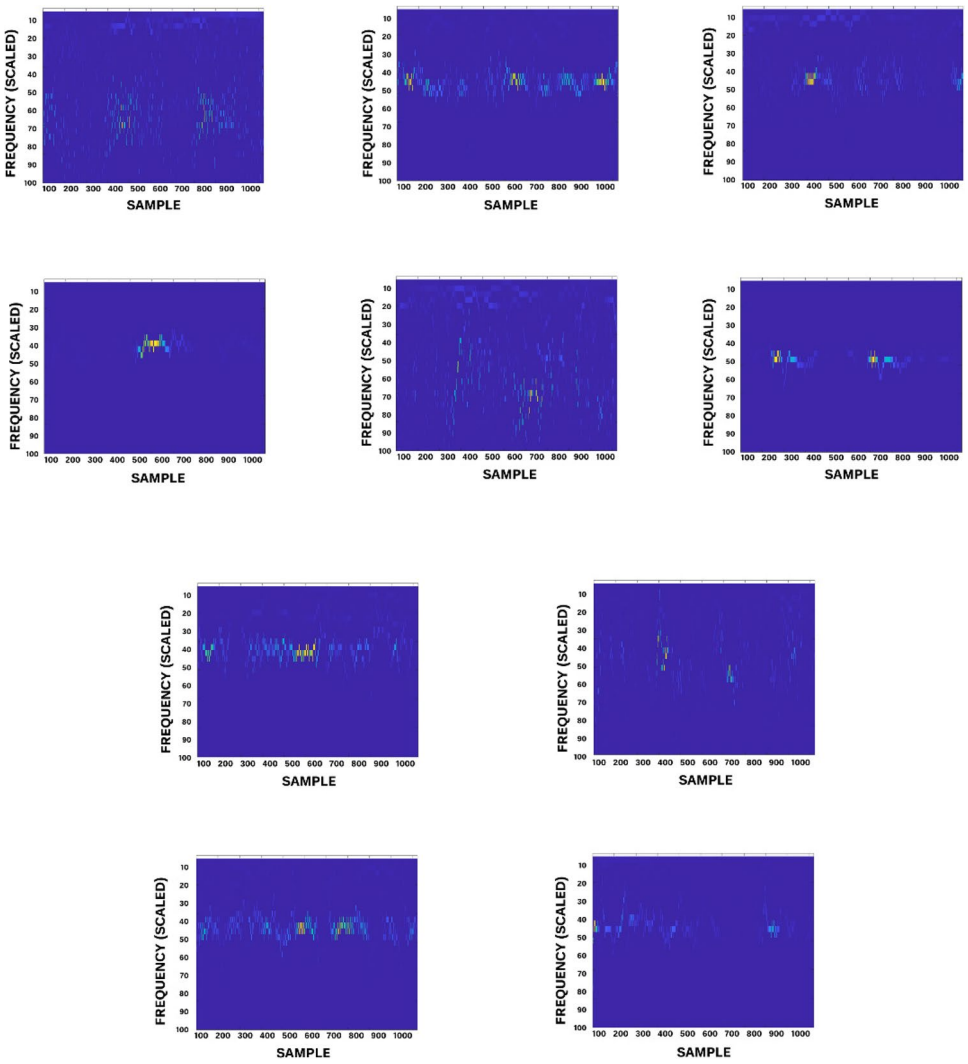



**Fig. 17** HHT images for each bearing condition (cropped)

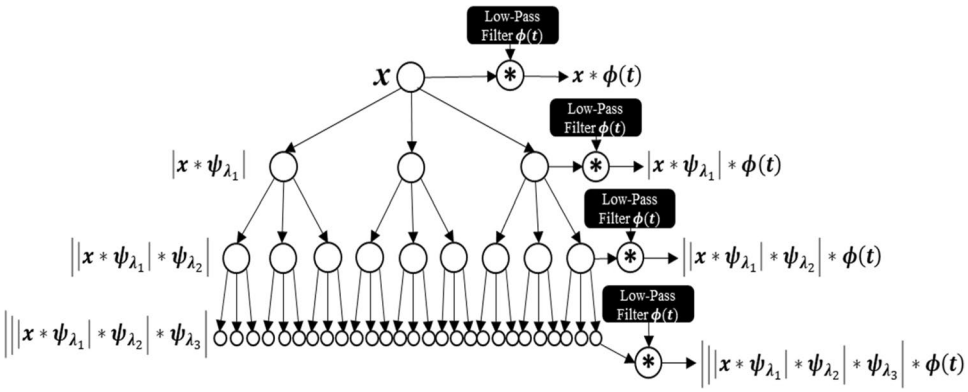



**Fig. 18** Visualization of scattering transform

function is completed quickly, with model complexity being $O(NlogN)$, similar to Fast Fourier Transform (Antoni, 2007).

In this work, two different Kurtogram types were tested; one contained 1024 datapoints with 256 datapoint window shifts, and the other, 4096 datapoints with 128 datapoint window shifts. The aim of this operation was to compare different datapoint lengths, and the effect of generalization on accuracies.

An example paving of the frequency/frequency resolution plane is shown in Fig. 25. The levels and

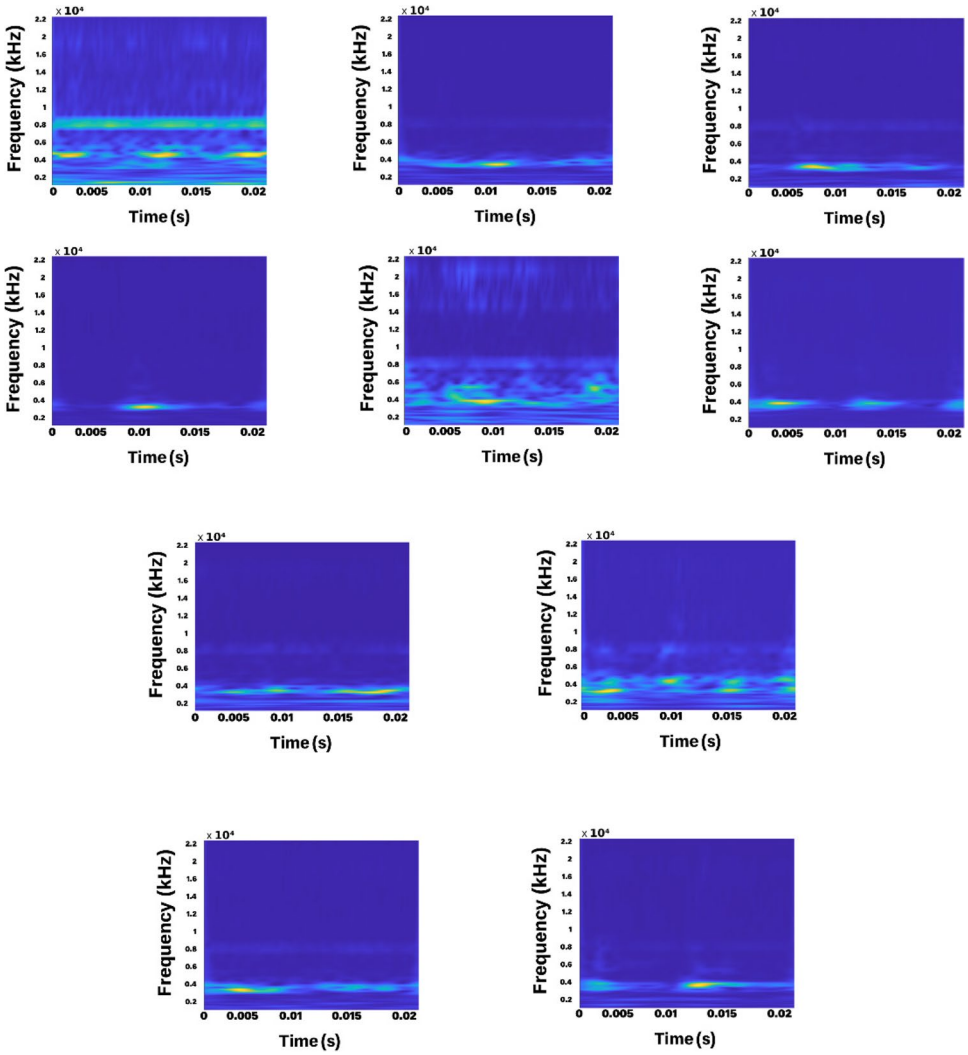



**Fig. 19** Scattergram with filter bank 1 images for each bearing condition

the frequency bands of those levels are modified for CWRU data, where sampling rate is 48000. It is possible to observe the 1/3 binary-tree structure. In this structure, there are 2^k different filters used, where k is the level. These band-pass filters divide the frequency content of the signal into 2^k levels. As can be seen, in level one there are two filters, which divide the frequency domain into two. The intensities of the spectral kurtosis transformation on said frequency bands are then colorized. Naturally, when the level k is increased, the transformation becomes less discrete.

Figure 26 shows the Kurtogram transformations for each bearing condition when the data length is set to 1024.

Figure 27 shows the Kurtogram transformations for each bearing condition when the data length is 4096. Because of the greater number of datapoints, the level k is increased from 4 to 6, resulting in improved, more continuous resolution.

### 3.3.5 Spurious Free Dynamic Range (SFDR)

Spurious Free Dynamic Range (SFDR) is the difference in dB between the power at peak frequency and the next largest frequency (spur). Essentially, it analyses the peaks in power spectrum discussed in 5.2.1, identifies the two highest peaks on the graph and calculates the difference in dB. It marks the highest as F, and the second highest as S. This study is the first one where SFDR method has been used for bearing fault classification.

The SFDR transform of each bearing condition is shown in Fig. 28. The y-axis is power in dB and the x axis is frequency in Hz.

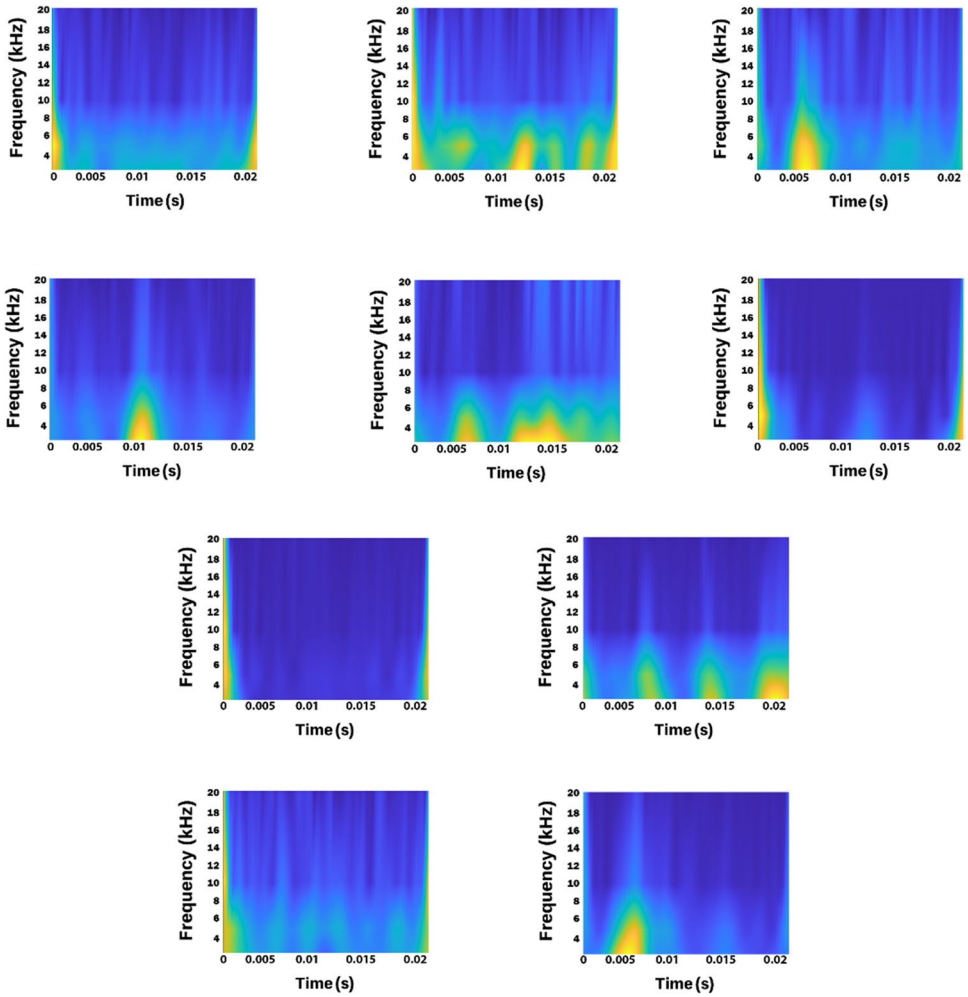



**Fig. 20** Scattergram with filter bank 2 images for each bearing condition

## 4 Datasets and Methodology Used

### 4.1 Case Western Reserve University Bearing Data

The dataset provided by Case Western Reserve University (CWRU) (2005) was used to validate the method proposed in this study. After importing the signals into MATLAB, data signals were prepared for Transfer Learning, using time–frequency conversion methods. After the transfer learning stage, there is a divergence between the methods that detect errors accurately, and those that detect them quickly. After windowing and basic signal processing, MATLAB and MATLAB's Signal Processing Toolbox was used to create hierarchical time–frequency and frequency representations from vibration signals of accelerometers positioned on the machine.

A large amount of data is needed to train systems with machine learning CNN, and thus, CWRU was selected. CWRU dataset contains accelerometer data in both 12 k and 48 k sampling frequencies. These chosen accelerometers were placed at both the drive-end and fan-end on a 2 hp reliance electric motor system. The bearings at either end were 6205-SKF. Signals from the accelerometers were then recorded by 16 channel DAT recorder and post-processed to MATLAB (.mat) format for ease of use (Case Western Reserve University, 2005).

The total number of samples was divided by 256 to approximate the maximum number of images. The data used for 2-D image creation is 1024 datapoints in length and are windowed with shifting 256 samples to the right at each step. This method provides more extensive information about the data signal. The sampling rate of the CWRU vibration signal is 48000, and in the method, only 1024 samples are used, i.e., each proposed time–frequency image lasts for 21.33 ms. Extracting many images using this data also improves the accuracy of CNN networks. CWRU test stand used for data acquisition is shown in Fig. 29. There is no standard approach for this data collection method, and data may be acquired from microphones, vibration sensors, or transducers. The key factor is data quality, which is achieved by creating a stable state free from exterior effects, such as noise.

In Fig. 29, the test stand that is used for recording bearing vibration is shown. CWRU obtained data on an electric motor, using accelerometers with high sampling frequencies

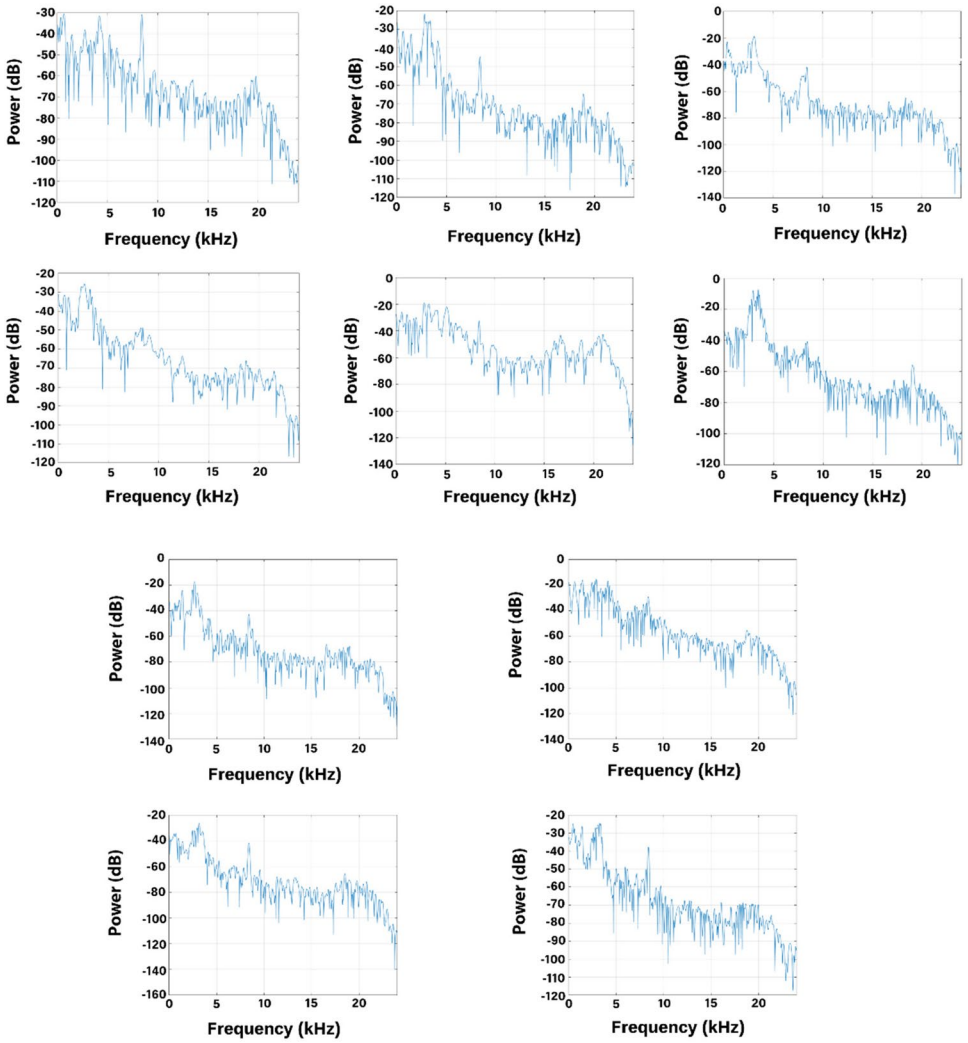



**Fig. 21** Power spectrum images for each bearing condition

on said test bench. Most previous articles that use CWRU dataset (Cao et al., 2020; Duan et al., 2021; Han et al., 2020; Li et al., 2019a; Neupane & Seok, 2020) for fault classification use a single working load on their proposed approaches, and calculate their accuracy based on this. Although convenient, using a single working load is not very practical, as the load can change simultaneously in a production environment. In this study, all possible loads (load 0–1-2–3) were used for training and validation.

## 4.2 Machine Fault Prevention Technology Bearing Dataset

To be able to validate the results obtained with CWRU bearing dataset, another similar dataset was used on the best performing time–frequency imaging method (Scattergram Filter Bank 1) on another dataset.

MFPT (Machine Fault Prevention Technology) bearing dataset (Bechhoefer, 2013) is another dataset that is widely used in the literature and has similar characteristics to CWRU bearing dataset. This dataset contains:

- 3 baseline (normal) condition signals on constant load of 270 lbs,
- 3 outer race fault condition signals on constant load of 270 lbs,
- 7 outer race fault condition signals with changing loads (25, 50, 100, 150, 200, 250 and 300 lbs),
- 7 inner race fault condition signals with changing loads (0, 50, 100, 150, 200, 250 and 300 lbs).

The sampling rate of the data varies between 97,656 samples per second, and 48,828 samples per second. Having different sampling rates would change the time–frequency images' frequency range, which would cause problems in classification since the images would have different scales and features. To fix the inconsistency in sampling rates, the frequency range of the time–frequency images are limited to 0–24 kHz (sampling

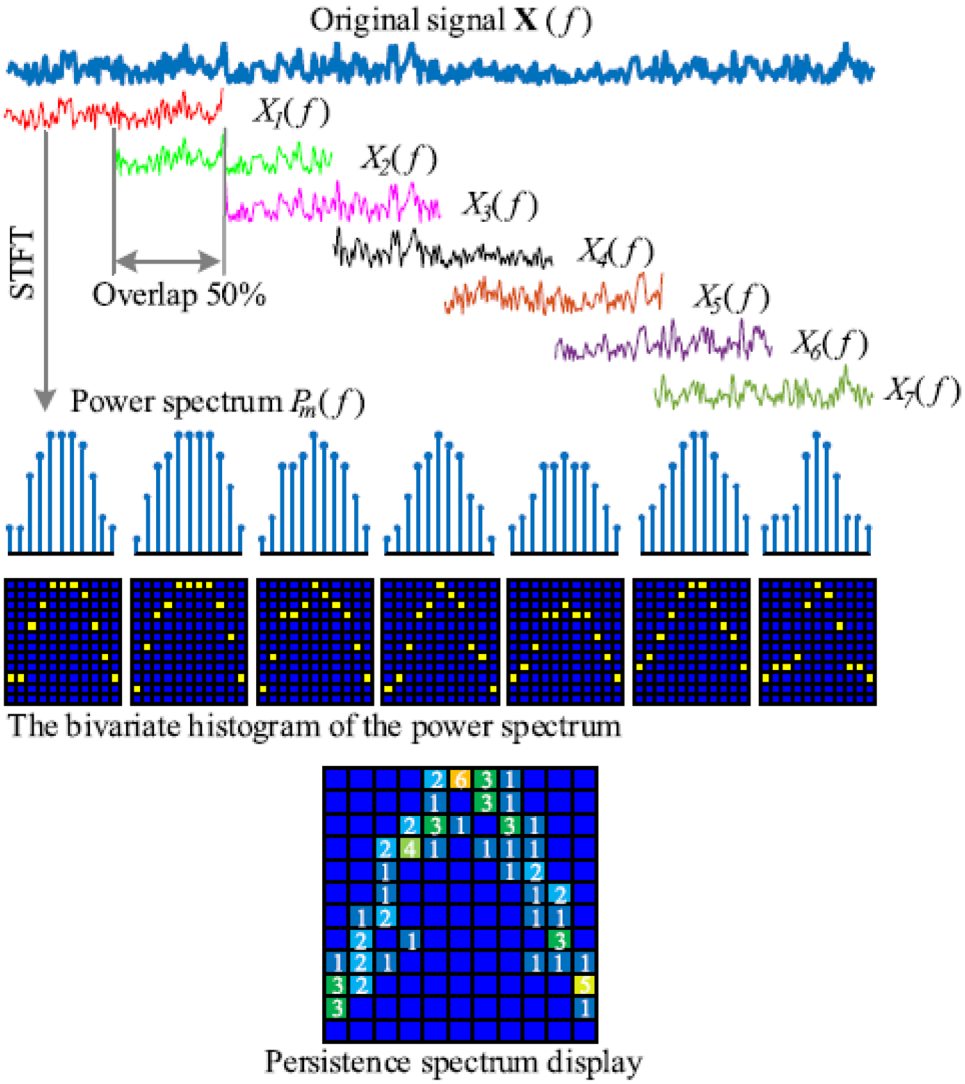



**Fig. 22** Visualization of persistence spectrum calculation

rates were standardized to 48,000 samples per second). Since fault sizes are not put into consideration in this dataset, there are only three labels to be predicted. These are normal, outer race fault, and inner race fault. Around 4800 (~1600 from each label) time–frequency images were created for the application on this dataset. These images combine differing load sizes, to achieve load-size-independent bearing fault classification.

### 4.3 Methodology Used in this Study

An examination of the basis of the deep learning structure used in bearing failures shows three different layers between image input and label output. These are convolutional layers, pooling layers and dense layers (Li et al., 2019a). Figure 30 shows how the CNN Structure processes images.

The key operation occurs in convolutional layers. In this stage, all transformation and transfer learning methods are implemented. To be more precise, vibration signals from bearings are converted into discrete frequency components using fast Fourier transform (FFT), enabling them to be used in CNN methods. The FFT and discrete Fourier transform (DFT) algorithms are used to analyze the raw vibration signal obtained from bearing in the frequency-domain. Compared to the time domain analysis, frequency-domain provides better visualization of the signal and reduction of its complexity, and can identify which frequencies produce most vibrations, and make the signal meaningful. This cannot be achieved in time domain (Neupane & Seok, 2020).

If the general structure seen in Fig. 30 is further customized, 5 steps are mentioned, as seen in Fig. 31. The input signal is raw vibration signal, therefore it is important that meaningful values are extracted by processing the data. The CWRU dataset received as input is formed by dividing the signal into 1024 segments and shifting to the right in 256 steps. Raw data as inputs are subject to many changes, as they may contain noise.

The output of time–frequency imaging box in Fig. 31, is shown in Fig. 32, where the imaging method is Fourier-based synchrosqueezing transform. The image on the left

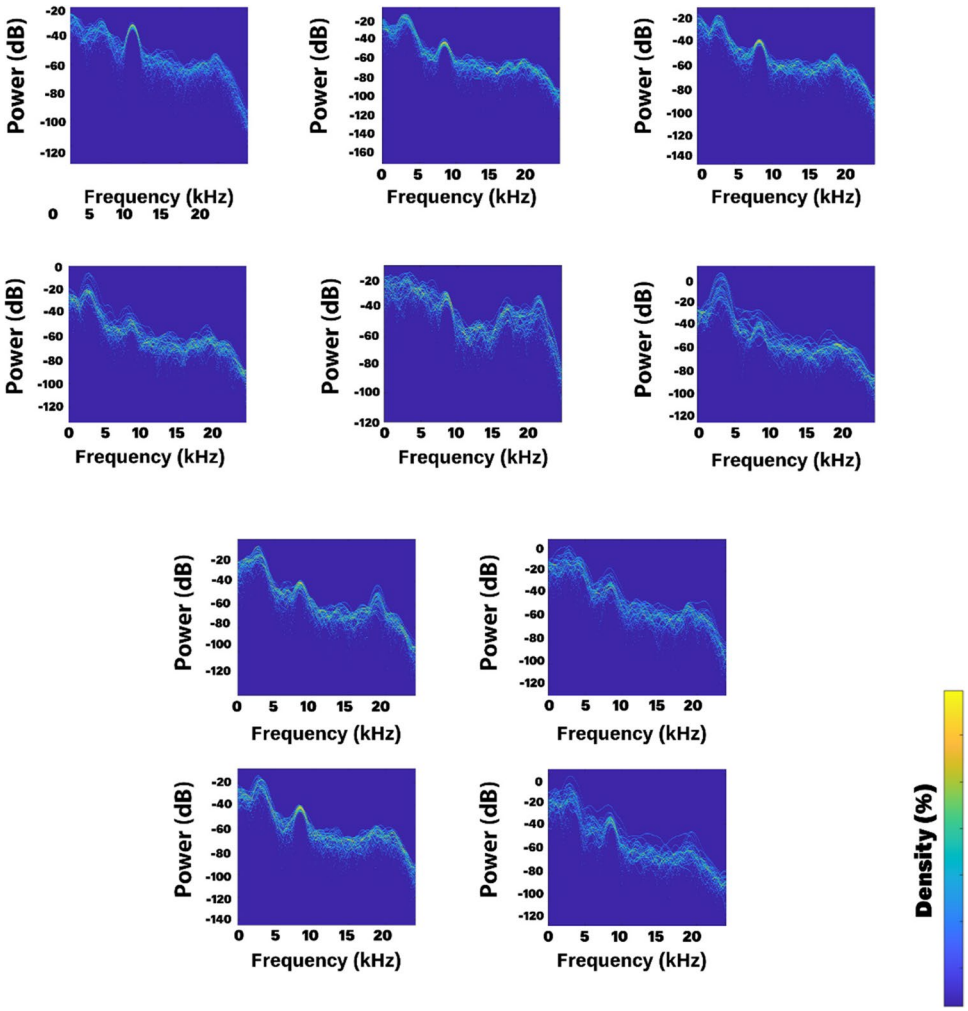



**Fig. 23** Persistence spectrum images for each bearing condition

shows an Outer Race Fault of 0.021 inches and the image on the right represents Outer Race Fault of 0.014 inches.

Figure 33 represents the results of the transfer learning CCNs in Fig. 31. The features that convolutional neural networks learned from the time–frequency image are shown. These are the features learnt by ResNet-50 CNN architecture, where the input is Scattergram Filter Bank 1 time–frequency images. Convolutional layer features in different depths are visualized. It is seen that the first convolutional layer mostly detects edges/lines and as depth is increased, more abstract features are obtained that are used for classification of different labels. The number of features firstly increase, and then decrease to the number of labels, which is expected with the use of CNN architectures.

Another aim of the study was to obtain higher accuracies by using transformation methods such as Scattergram, Spurious Free Dynamic Range (SFDR), Fourier-based Synchrosqueezing Transform (FSST), and Persistence, which are rarely or never used in the literature for bearing fault classification.

After the 2-D images passed through the transfer learning stages, different categories were defined: the normal case, and three different sizes of inner race fault, outer race fault, and ball fault, making a total of ten categories. The three types of faults were categorized by size as 7 mils, 14 mils and 21 mils). In second section of the cited study, detailed information about this dataset is given (Zhang et al., 2020c). To further validate the results obtained in the first dataset, MFPT dataset was also used in this study with the best performing imaging method obtained in the first dataset.

In the study, the vibration signals received from the relevant equipment were classified, and the status of defective and non-defective conditions are decided. Changes in vibration signals are important for fault detection. There are two approaches for fault classification. The first of these involves extracting time domain, frequency domain and phase domain features from the signal, then training on the feature space using various machine learning algorithms. This method requires knowledge of feature engineering and signal processing (An & Jiang, 2014). Bearing faults processed within the scope of the study are displayed in Fig. 34.

To validate the proposed approach, images were created with different time–frequency and frequency methods (WVD, CWT, FSST, WSST, Spectrogram, Power, Persistence, Spectral Kurtosis), from which four different pretrained networks

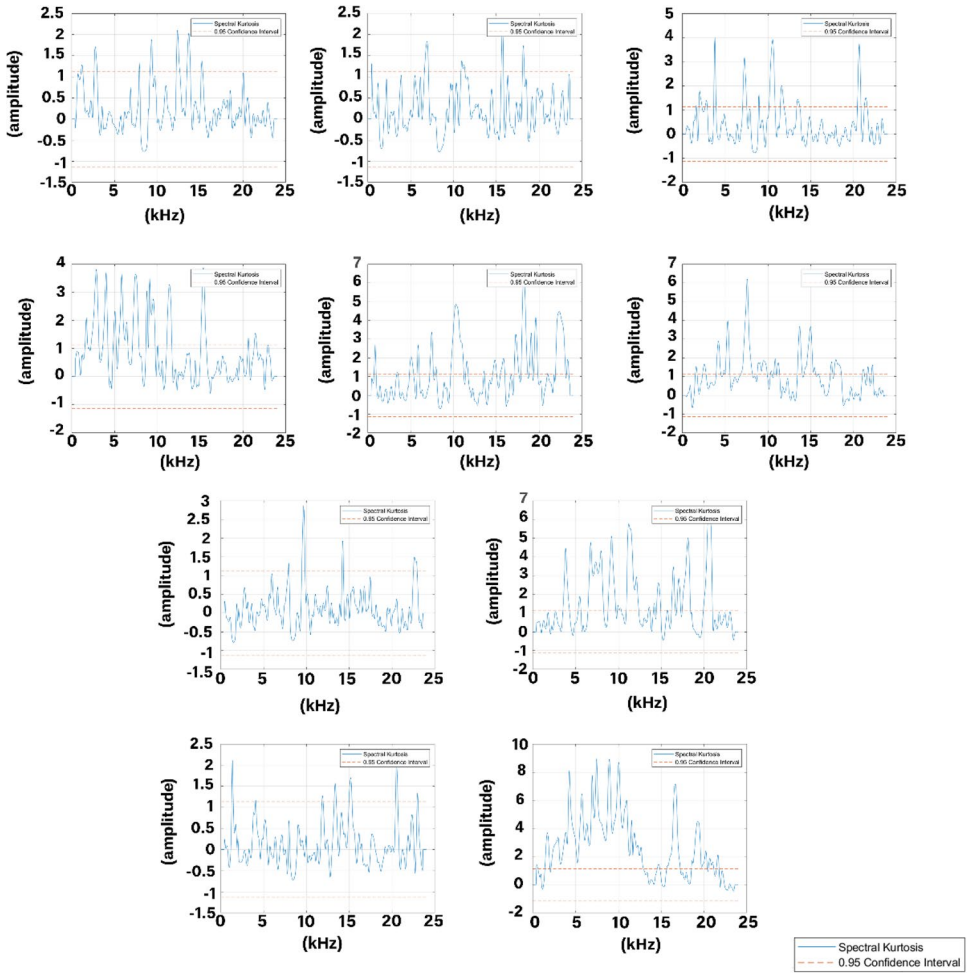



**Fig. 24** Spectral kurtosis images for each bearing condition

were retrained (SqueezeNet, GoogLeNet, Res-Net 50, Inception Resnet-v2) to predict ten bearing conditions. These networks were retrained by using transfer learning approach, which requires replacing the last fully connected layer and the output layer with appropriate ones. To train these networks, all images created were resized using MATLAB's augmentedImageDatastore() function. SqueezeNet accepts 227*227*3 images, ResNet-50 and GoogLeNet accept 224*224*3 images and Inception Resnet-v2 accepts 299*299*3 images. MATLAB's Deep Network Designer was used for downloading pretrained networks and changing the layers' properties directly, replacing layers, analysing the architecture, loading the data, and beginning the training. The process was facilitated by Deep Network Designer's user-friendly and straightforward interface, with no need to write code. The training parameters used in this case can be seen on Table 4, other parameters were left as default.

The training process involved a total of 4000 images for each case, consisting of 400 images from each health conditions (BallFault007, BallFault014, BallFault021, Normal, InnerRaceFault007, InnerRaceFault014, InnerRaceFault021, OuterRaceFault007, OuterRaceFault014, OuterRaceFault021). Fault conditions each had 4 loads, therefore every fault state consisted of 100 image samples. The training/validation split was chosen as 70/30, creating 280 training and 120 validation samples from each health condition. To further validate the results, a total of 15,800 images created for each case were classified and used to calculate the total accuracy.

On the MFPT dataset, 400 images from each health condition (Normal, Outer Race Fault, Inner Race Fault) were trained with 70/30 training/validation split. Then, the networks' performances were tested on 1600 images from each case.

Training processes took place on MATLAB r2021a.

## 5 Experimental Results

Confusion matrices of all methods used are given in Appendix A. Table 5 shows accuracy values on CWRU dataset achieved by each method, on the validation data that contains 15,800 images.

The best performing imaging method and network on CWRU dataset is Scattergram Filter Bank 1, retrained on ResNet-50

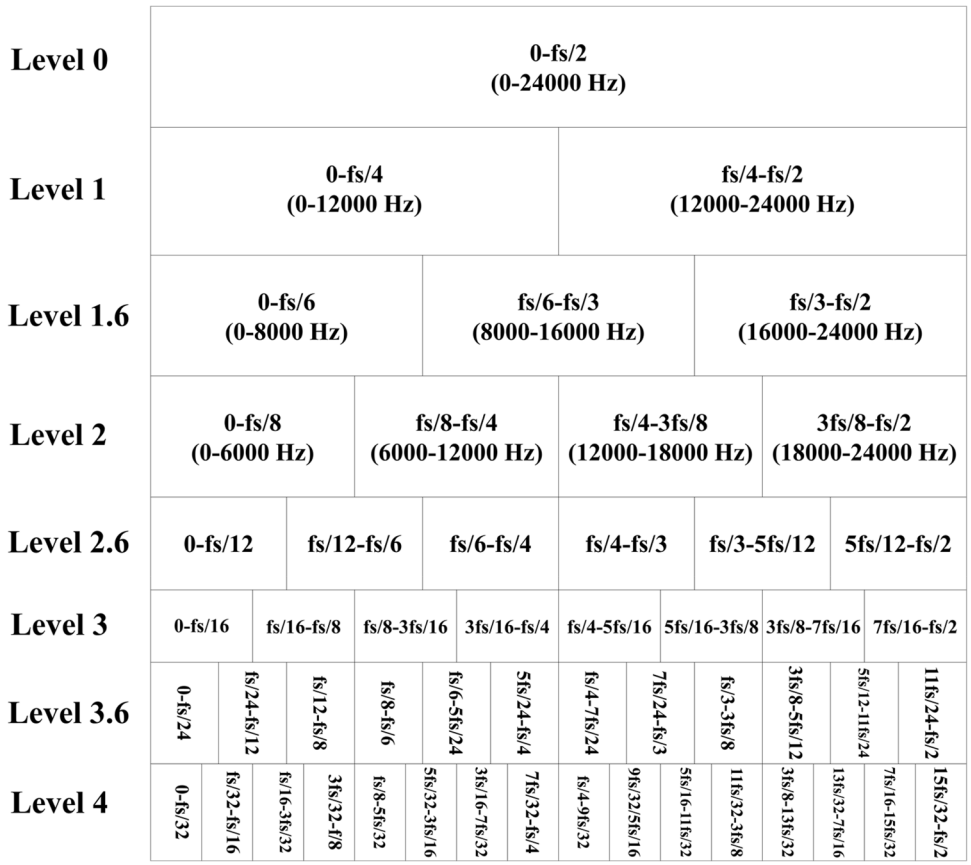



**Fig. 25** Visualization of kurtogram transformation

architecture. This imaging method was also tested on MFPT dataset. The results obtained on MFPT dataset is shown in Table 6.

## 6 Conclusion & Discussion

The study aims to predict bearing errors and minimize the material losses caused by the machine. In this study, various frequency and time–frequency images were used for training of CNN's (pre-trained neural networks) to classify bearing faults. The widely used CWRU dataset was adopted to verify the accuracy of the proposed method. To validate the results and show replicability, the MFPT dataset was also used in this study. The accelerometer signal in time was windowed, and then the 1024 time samples were used to create 18 different image types. The images were labelled and resized to appropriate measures. These images were used on available, open source pre-trained networks SqueezeNet, GoogLeNet, ResNet-50 and Inception-ResNet-v2. The proposed methods led to the following conclusions:

- In every training phase on CWRU dataset, 400 images were used for each bearing condition (100 images for each load 0–1-2–3), i.e., a total of 4000 images. After the randomized train-validation split, only 2800 images were needed in the training phase, resulting in a shorter training time.
- For the validation phase on CWRU dataset, 1600 images were used for each bearing condition (400 images for each load 0–1-2–3), summing to a total of ~ 16,000 images, with ~ 13,200 unseen images for each method. In conclusion, the train-validation split of proposed method was 17.5% for training and 82.5% for validation. The best accuracy achieved on this split was 99.89%, further proving the robustness of the method used.
- The best performing time–frequency imaging method out of the methods that were benchmarked on CWRU dataset was tested on MFPT dataset, and 100% accuracy was achieved on Inception-ResNet-v2 CNN architecture. This shows that the methodology used in this study is replicable on vibration data that are sampled with high frequencies.
- Unlike other approaches in the literature, the proposed method was trained with all the loads and on all fault sizes. The total sample size was 4–8 times greater than in most approaches. This made the proposed approach more versatile and less prone to overfitting.
- In a real-world production environment, the load on the motor can change simultaneously, and controlling the load on a set value is hard to do. The proposed method achieves great accuracy regardless of the load that is on the DC motor, making this approach unique and usable in real-world industrial applications.

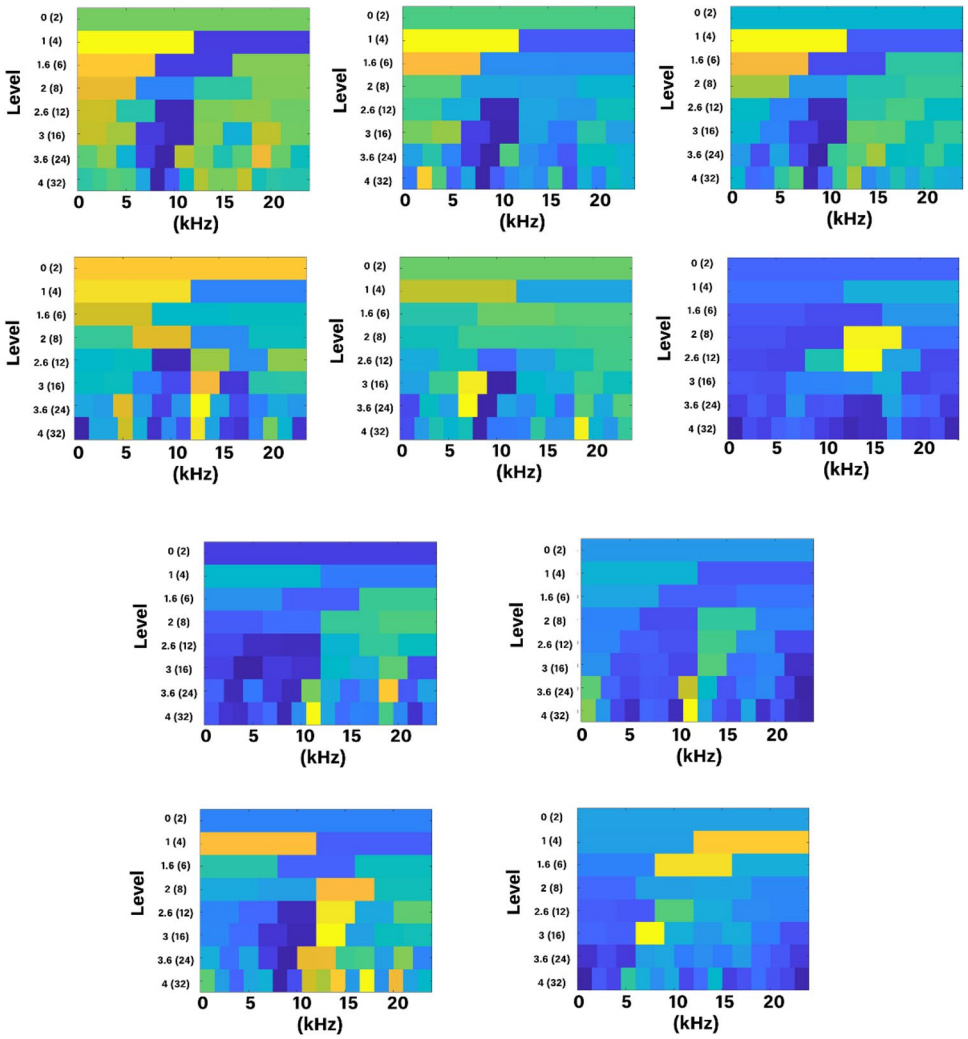



**Fig. 26** Kurtogram images for each bearing condition (non-generalized)

- MFPT dataset contained 8 different load sizes on outer race fault conditions and seven different load sizes on inner race fault conditions. Although loads differ greatly on this dataset, 100% accuracy was achieved independently from the load sizes. Load-independent bearing fault classification was successfully achieved in this study.
- Out of 72 validation accuracies on CWRU dataset, it is seen that 10 have 99.5%+accuracy, nine are between 99%-99.5% accuracy, and 25 are between 97%-99% accuracy. This statistic shows that most methods used in this study give extremely reliable results, similar to state-of-the-art studies.
- Since SqueezeNet is a sequential and simple convolutional neural network with only 18 layers of depth, its performance is lower than that of other methods. The highest accuracy achieved with SqueezeNet on CWRU dataset is 99.04% when the input is Fourier-based Synchrosqueezing Transform images.
- GoogLeNet, with more depth (22 layers) and inception modules, generally performs better than SqueezeNet, but worse than ResNet-50 and Inception-ResNet-v2. The reason for this might be the information loss between its layers. The best accuracy achieved with this CNN on CWRU dataset is 99.25% with Instantaneous Frequency image input.
- Although not the deepest network in this study (50 layers), ResNet-50 obtains the best accuracy. Of 18 methods used on CWRU dataset, ResNet-50 gave the best performing result in 11. The best accuracy achieved on CWRU dataset in this study (99.89%, Scattergram FB2) used the ResNet-50 architecture.
- Inception-ResNet-v2 is the deepest/most resource consuming network in this study, but not the best-performing network on CWRU dataset. It achieves the best result in six of the methods. The best result achieved with this CNN is 99.79%, with Fourier-based Synchrosqueezing Transform. On MFPT dataset, Inception-ResNet-v2 proved to be the most successful CNN architecture with 100% accuracy.
- The average accuracies for each CNN show that Fourier-based Synchrosqueezing Transform gives the best result overall, with an average of 99.45% accuracy. This shows that FSST holds the best/clearest information of the 18 methods.

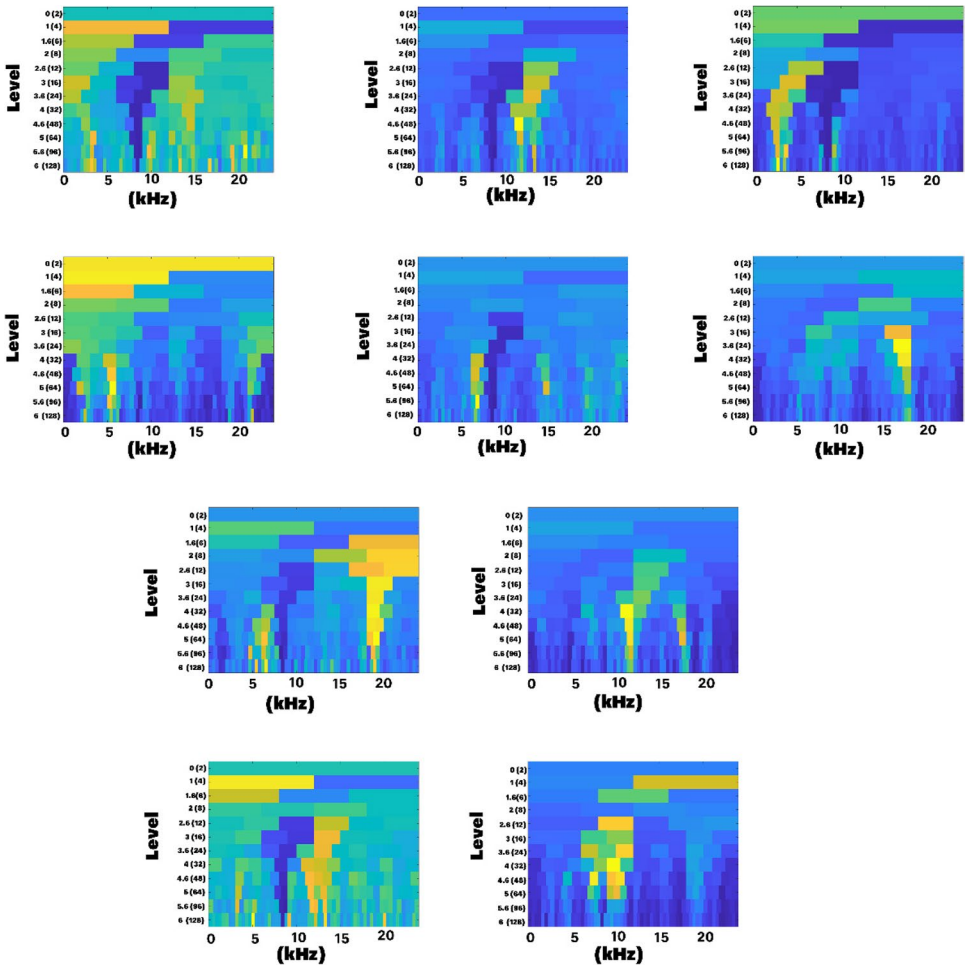



**Fig. 27** Kurtogram images for each bearing condition (generalized)

- Thresholding the spectrograms on a certain dB value decreased the total accuracy caused by the information loss.
- Cropping the HHT images to only include meaningful parts of the image increased the accuracy by ~ 1–3%, further showing the importance of cropping redundant areas on CNN training.
- Because of its tree structure, the Kurtogram is more discrete than other methods in this study, thus, the generalization process was of utmost importance. Generalization of Kurtogram resulted in a dramatic improvement of 7–11%, showing the benefit of increasing the number of datapoints on discrete imaging methods.
- Changing the filter bank on the scattergram changes the accuracy by as much as 9–12%, showing the significance of appropriate filter usage for the application.
- There were no previous studies that made use of Scattergram with Filter Bank 1 on CWRU dataset. This method gave the best results out of 18 methods on CWRU dataset and it also achieved great results on MFPT dataset, strongly suggesting that scattergrams should be used more often in the future.
- With 72 diverse training and validation phases, this study is one of the most comprehensive and detailed studies employing CWRU Bearing Data Center dataset.

## 7 Recommendations & Future Works

The results could also be evaluated on CNNs such as ResNet-18 and NASNet-Large. A MATLAB toolbox can be developed that creates these time–frequency method images from vibration signals automatically, lowering the time needed for signal processing. A smart/wireless vibration sensor can be produced that would automate the process of bearing fault classification. The benchmark done in this study can be validated with real-world applications.

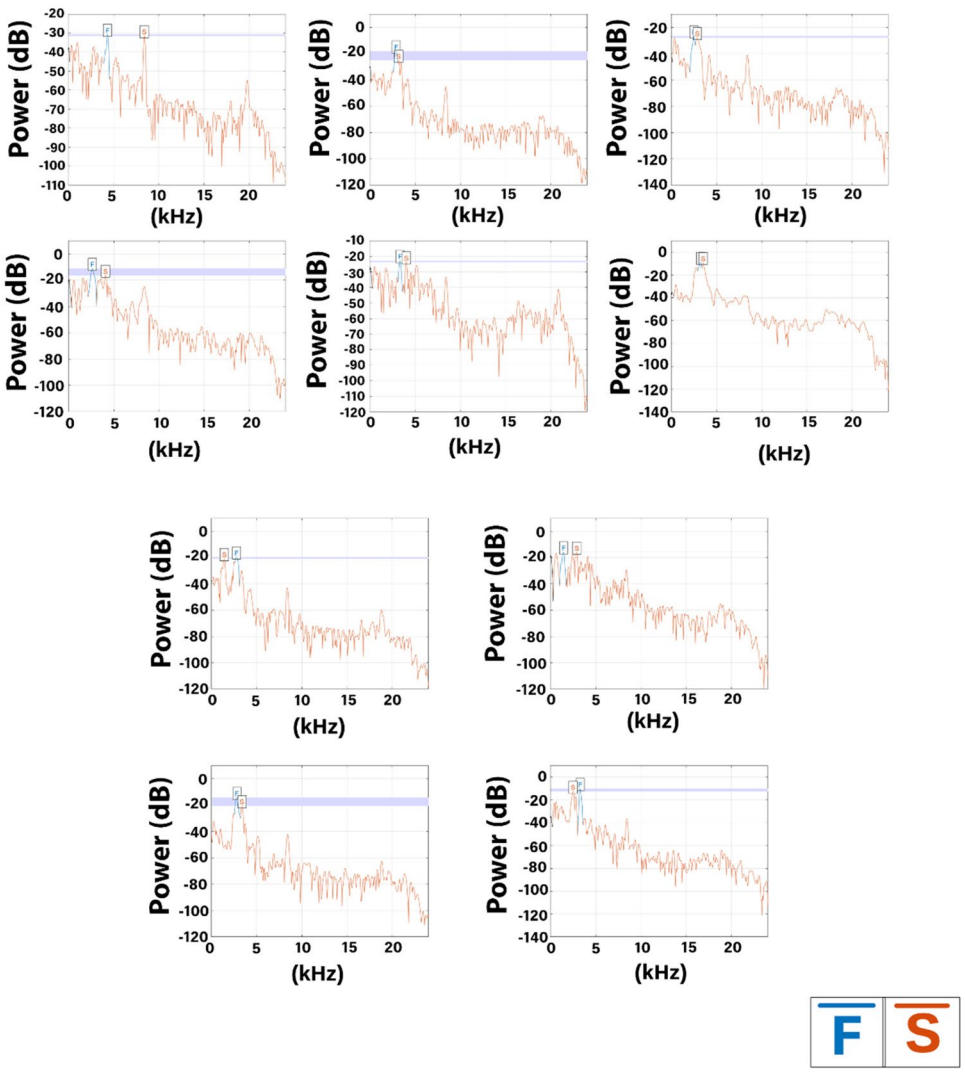



**Fig. 28** SFDR images for each bearing condition

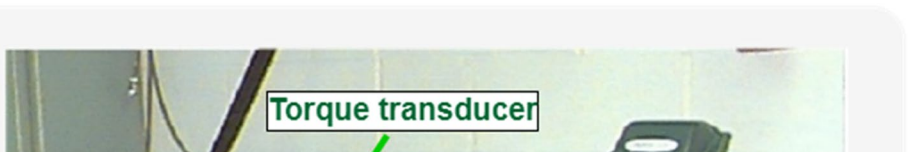





**Fig. 29** Test stand for accelerometer data acquisition (Case Western Reserve University, 2005)

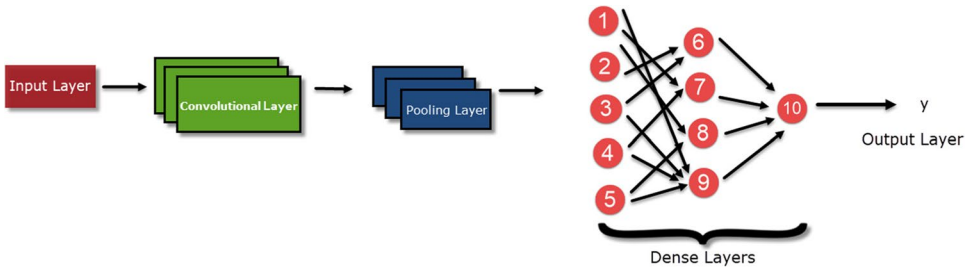



**Fig. 30** A basic CNN structure for image classification (Li et al., 2019a)

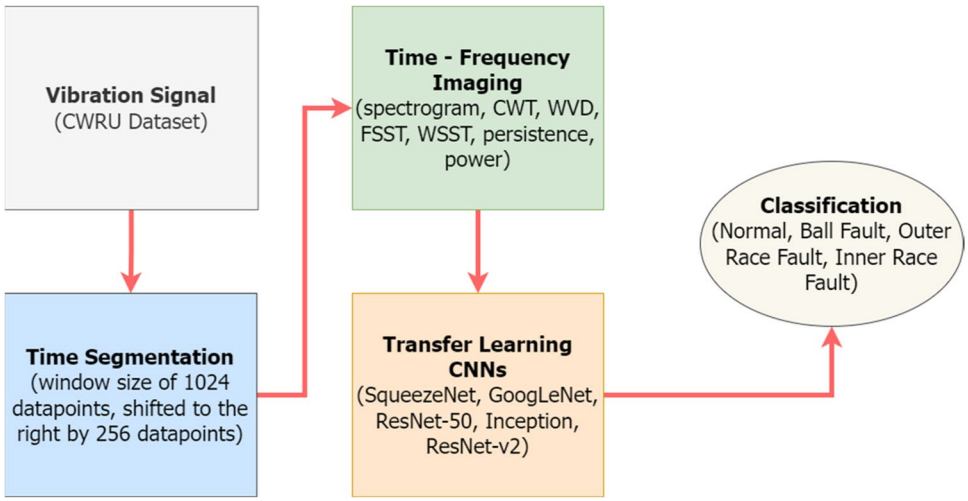



**Fig. 31** General flow of processes between input–output

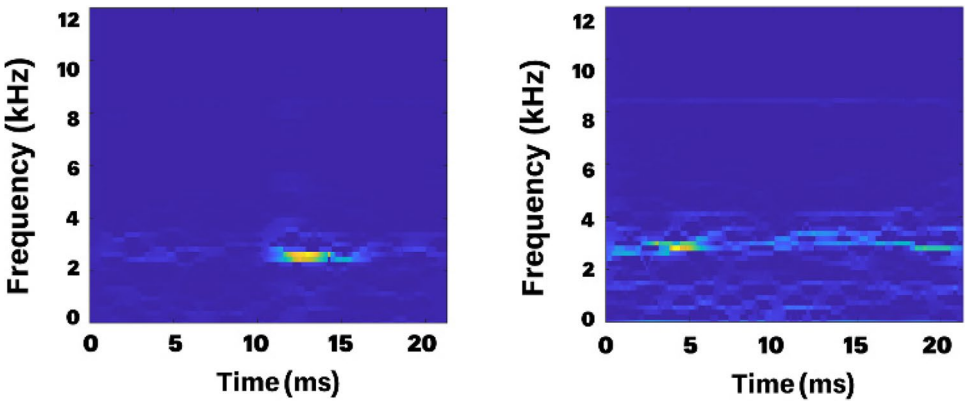



**Fig. 32** Two example bearing vibration images from this study

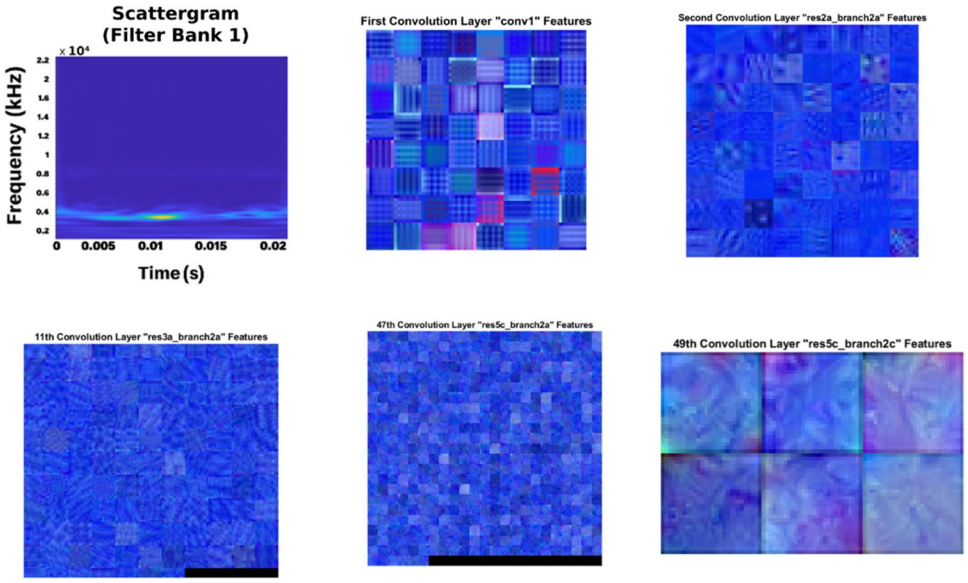



**Fig. 33** Convolutional layer features of ResNet-50

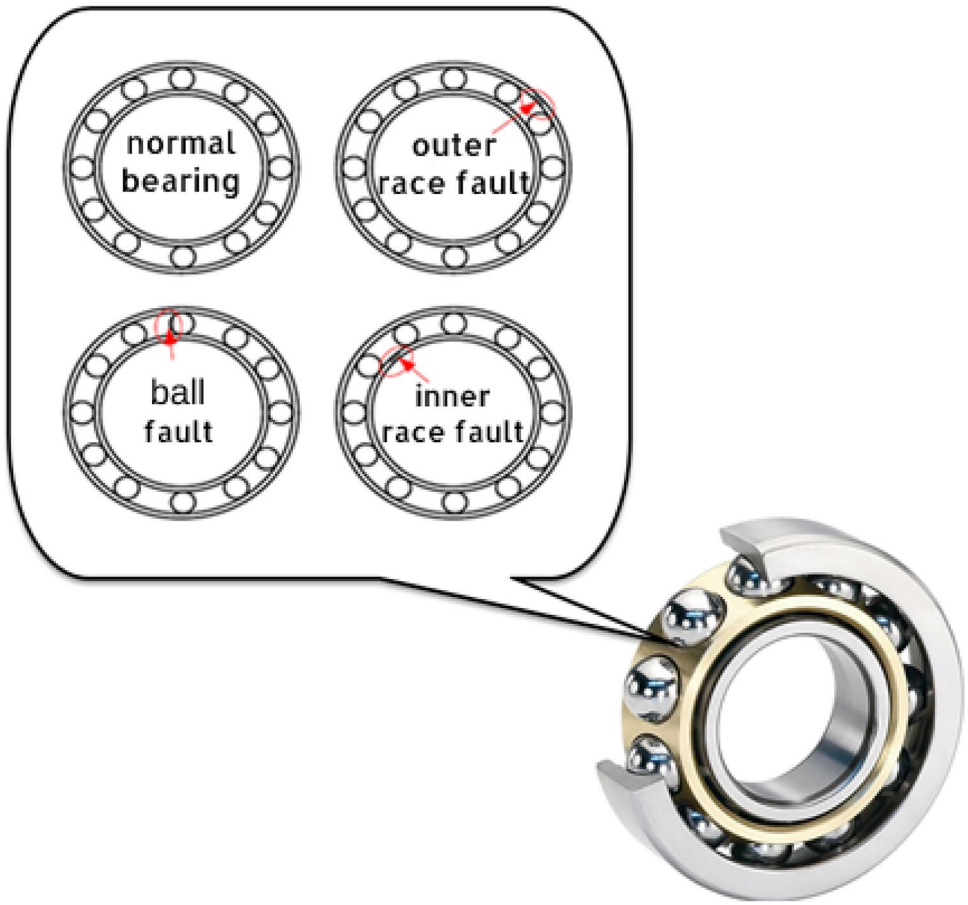



**Fig. 34** Examples of bearing faults processed in the study

**Table 4** Parameters used in training

| Networks | SqueezeNet, GoogLeNet, ResNet-50 | Inception ResNet-v2 |
|---|---|---|
| Learning Rate | 1e-4 | 1e-4 |
| Mini Batch Size | 64 | 32 |
| Optimizer | Adam | Adam |
| Validation Frequency | 50 | 50 |
| Number of Epochs | 30 | 15 |
| Number of Iterations | 1290 | 1305 |

**Table 5** Total accuracies reached (CWRU)

| Method | Networks / Total Accuracy Reached (CWRU) | | | |
|---|---|---|---|---|
| | SqueezeNet | GoogLeNet | ResNet-50 | Inception Resnet-v2 |
| Spectrogram (Unthresholded) | 97.67% | 98.27% | 98.92% | 98.59% |
| Spectrogram (Thresholded) | 96.00% | 97.25% | 98.00% | 96.13% |
| Wavelet-based Synchrosqueezing Transform | 97.04% | 97.63% | 98.07% | 98.68% |
| Fourier-based Synchrosqueezing Transform | 99.04% | 99.14% | 99.84% | 99.79% |
| Wigner Ville Distribution | 96.21% | 96.46% | 98.22% | 98.74% |
| Continuous Wavelet Transform | 98.93% | 99.16% | 99.66% | 99.34% |
| Constant-Q Nonstationary Gabor Transform | 98.44% | 98.50% | 99.55% | 99.08% |
| Hilbert Huang Transform | 88.91% | 90.91% | 93.41% | 92.16% |
| Hilbert Huang Transform (Cropped) | 91.38% | 91.55% | 93.73% | 93.47% |
| Scattergram Filter Bank 1 | 98.23% | 98.50% | 99.89% | 99.68% |
| Scattergram Filter Bank 2 | 86.08% | 86.84% | 89.04% | 90.37% |
| Instantaneous Frequency | 98.82% | 99.25% | 99.83% | 99.57% |
| Kurtogram | 85.19% | 87.85% | 90.44% | 90.44% |
| Kurtogram (Generalized) | 96.32% | 96.68% | 97.44% | 97.75% |
| Power | 97% | 97.15% | 99.25% | 99.34% |
| Persistence | 98.62% | 99.18% | 99.71% | 99.70% |
| Spectral Kurtosis | 76.40% | 76.66% | 84.39% | 83.5% |
| Spurious Free Dynamic Range | 95.53% | 95.52% | 97.15% | 98.37% |

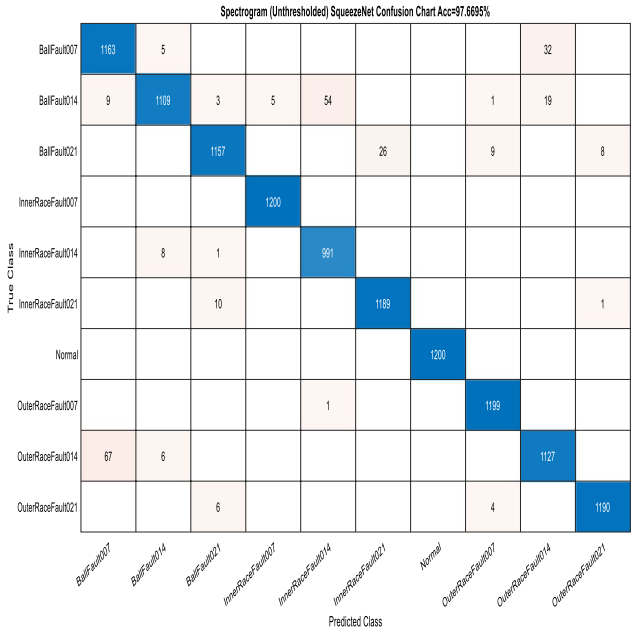
**Table 6** Total accuracies reached (MFPT)

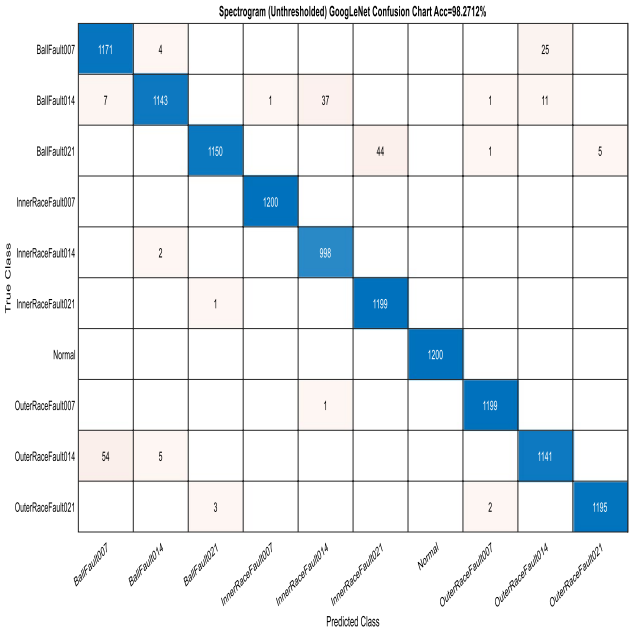
| Method | Networks / Total Accuracy Reached (MFPT) | | | |
|---|---|---|---|---|
| | SqueezeNet | GoogLeNet | ResNet-50 | Inception Resnet-v2 |
| Scattergram Filter Bank 1 | 99.44% | 99.13% | 99.92% | 100.00% |

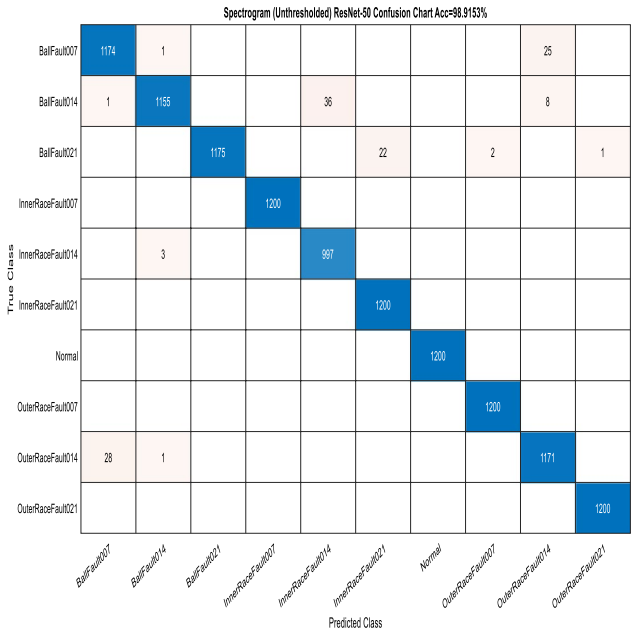
# Appendix A

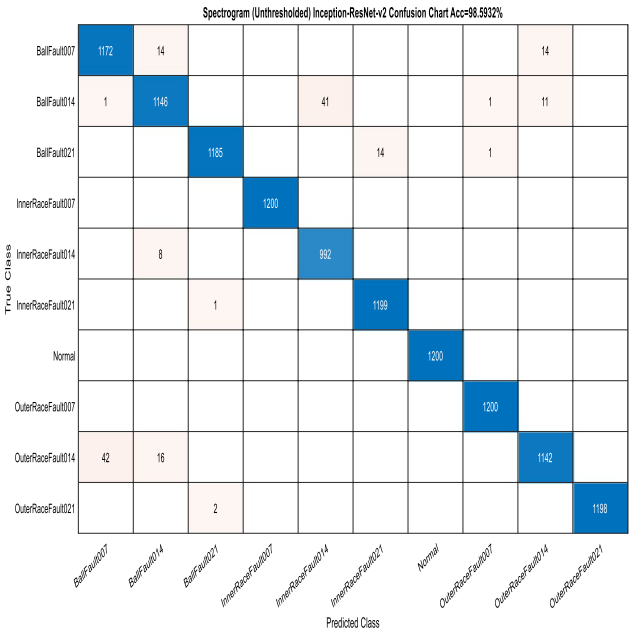
## Confusion Matrices of Each Method

1. Spectrogram (Unthresholded)

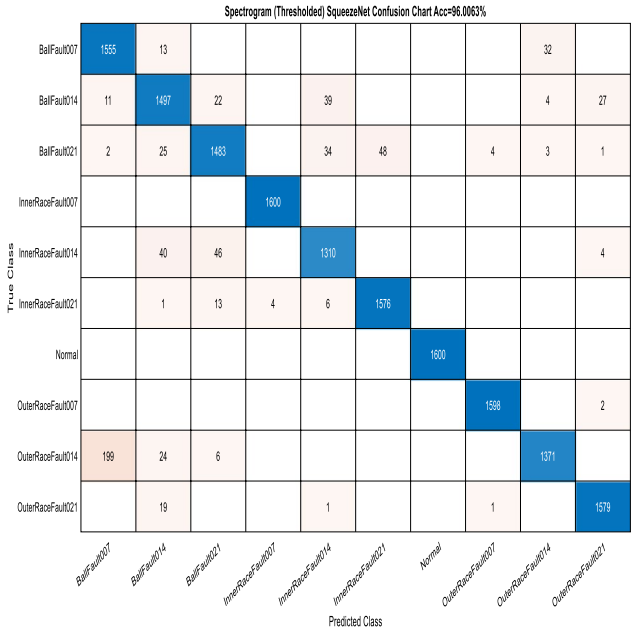


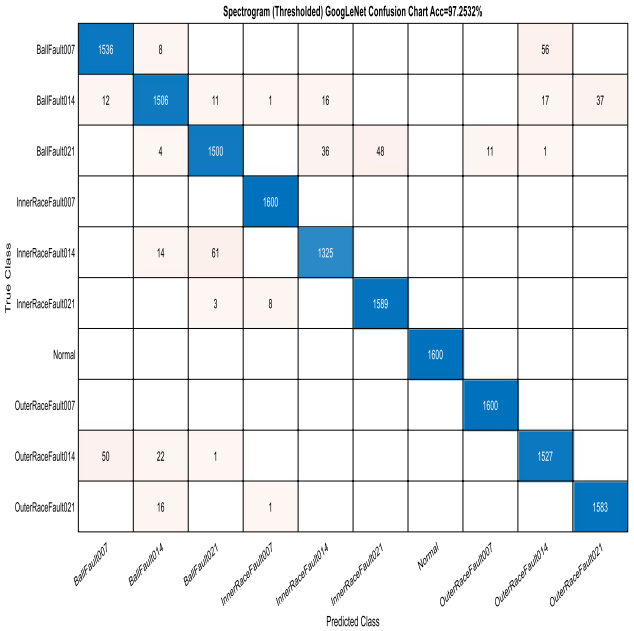



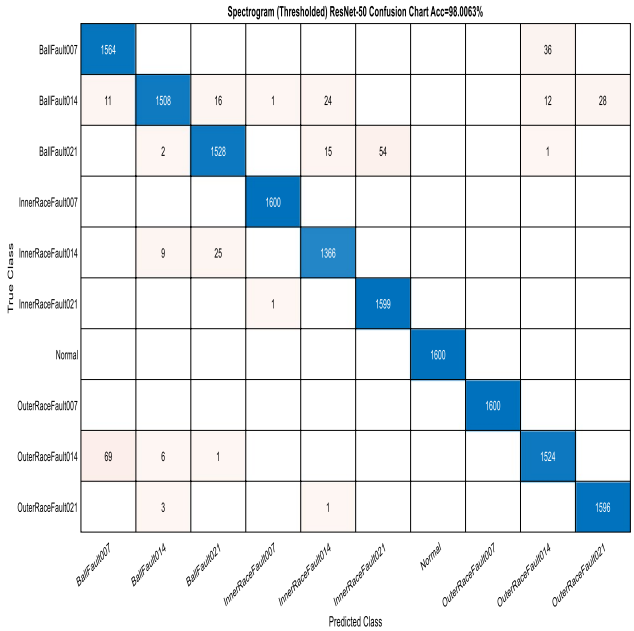



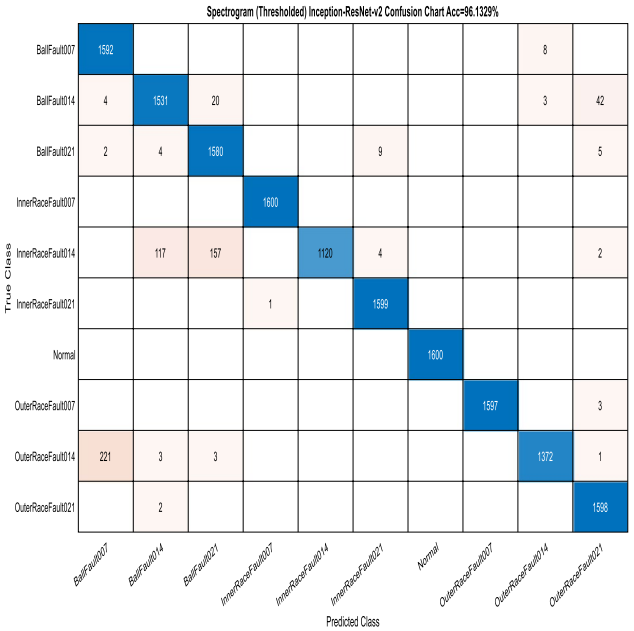



2. Spectrogram (Thresholded)

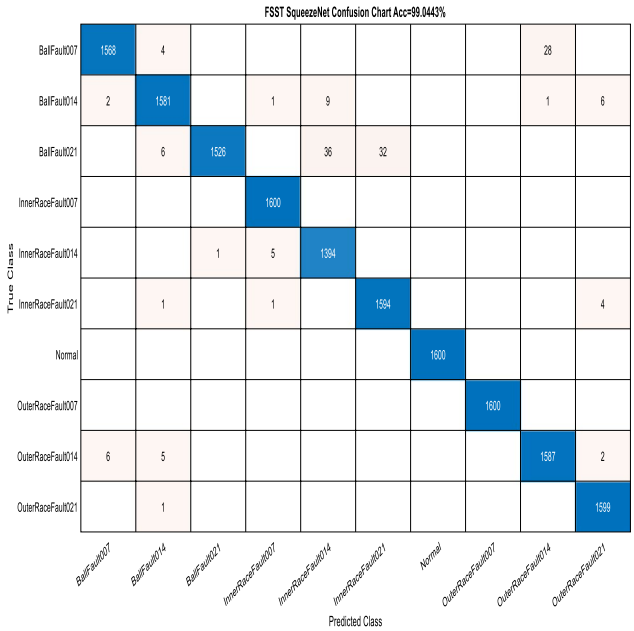


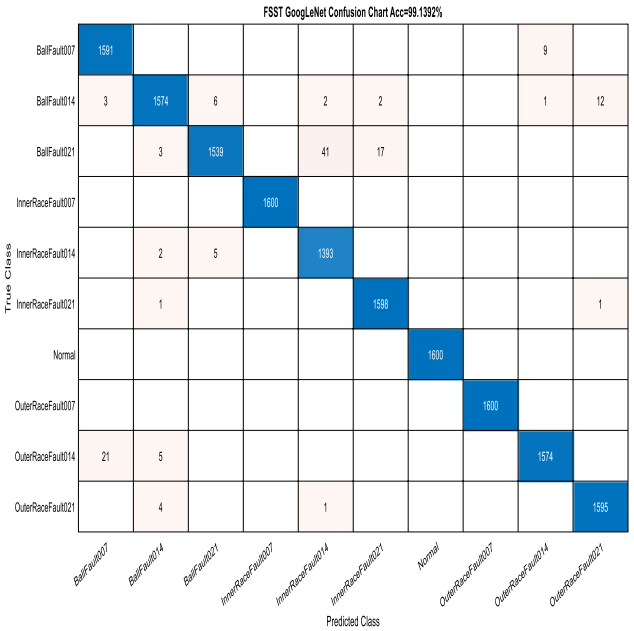



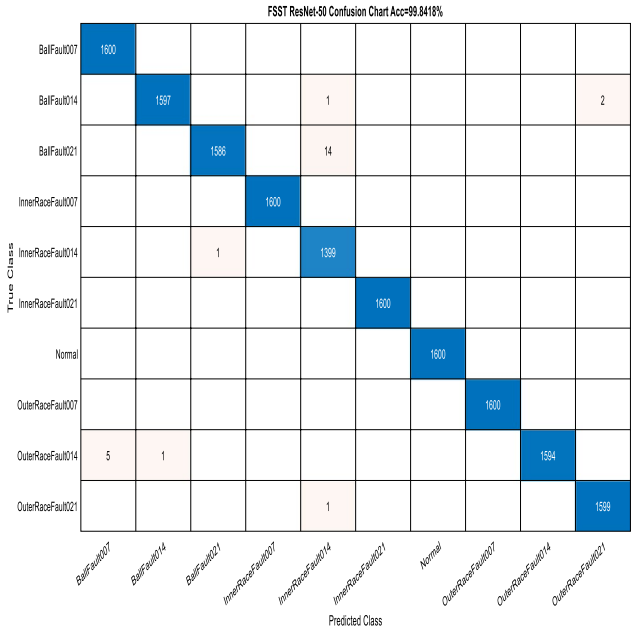



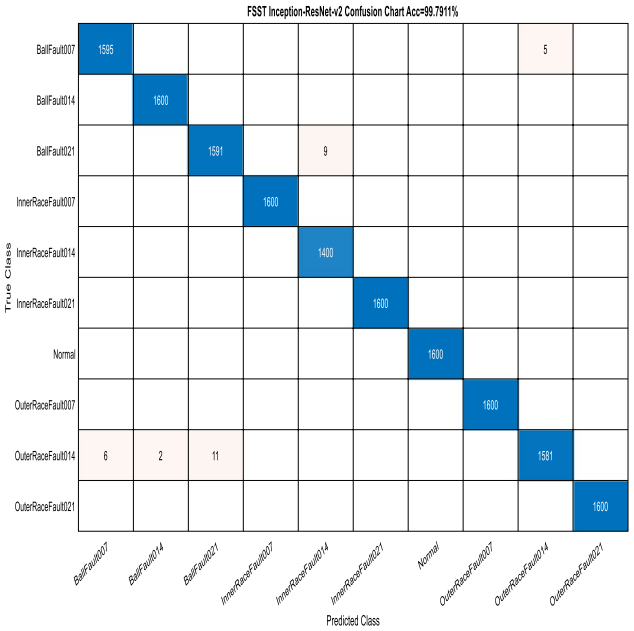



3. Fourier Synchrosqueezing Transform

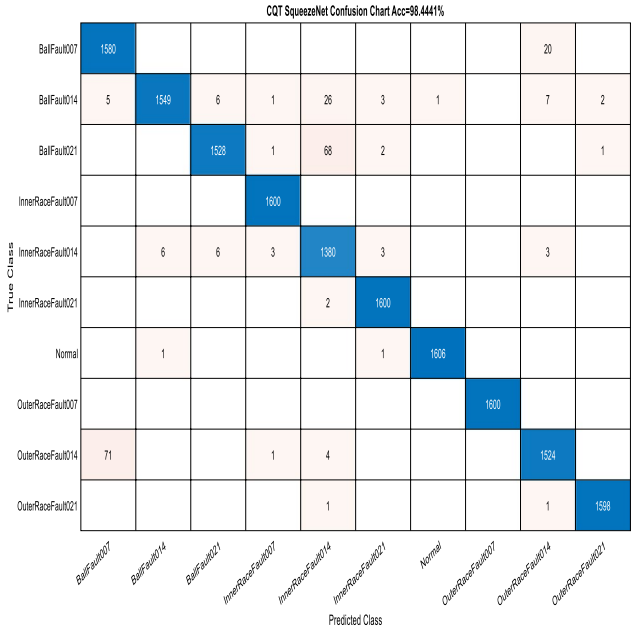


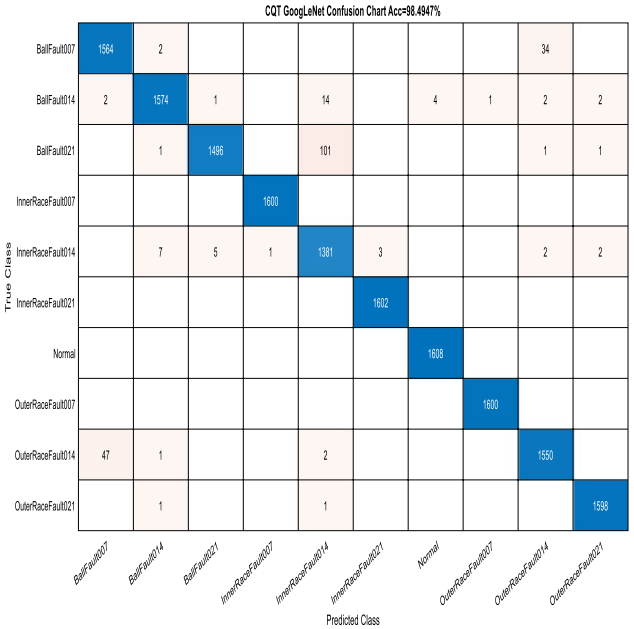



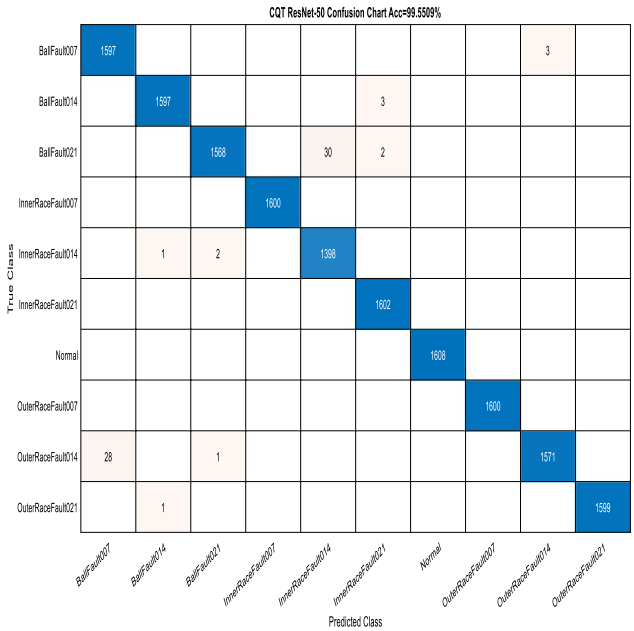



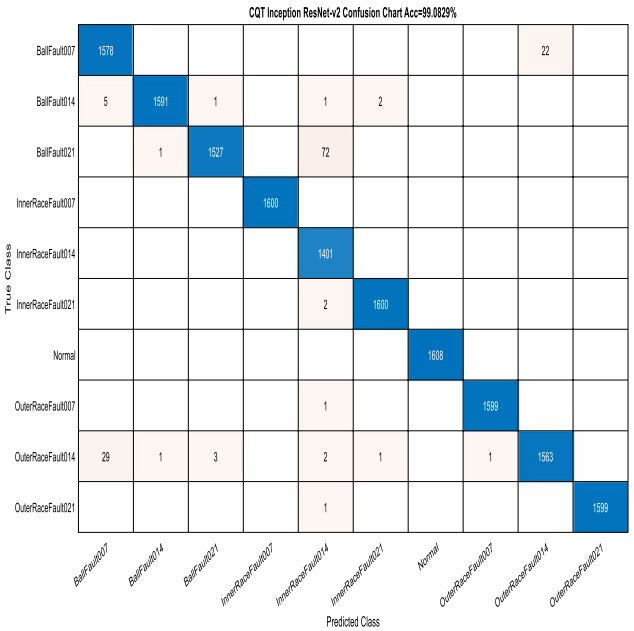



## 4. Continuous Q Transform

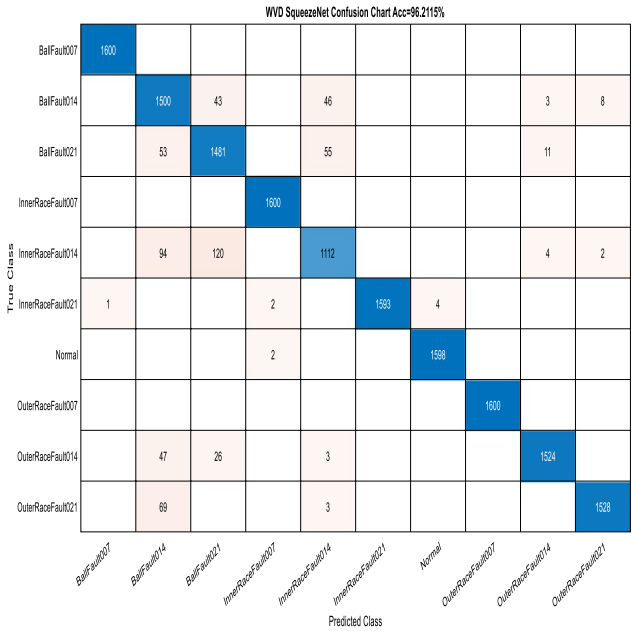


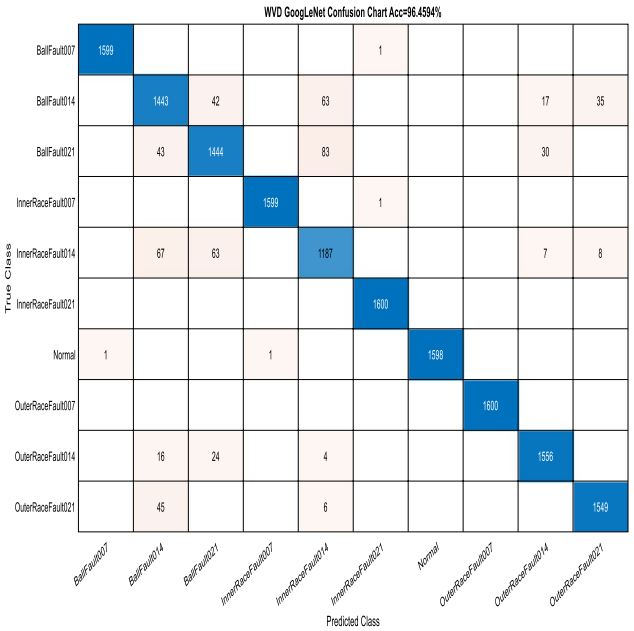



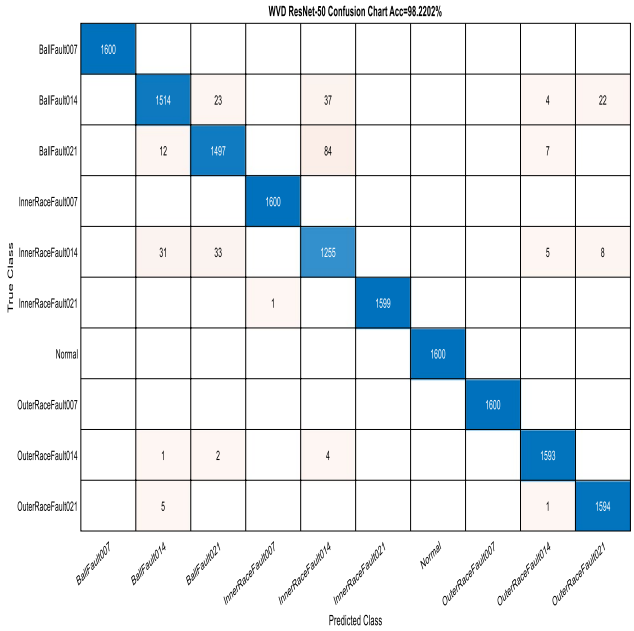



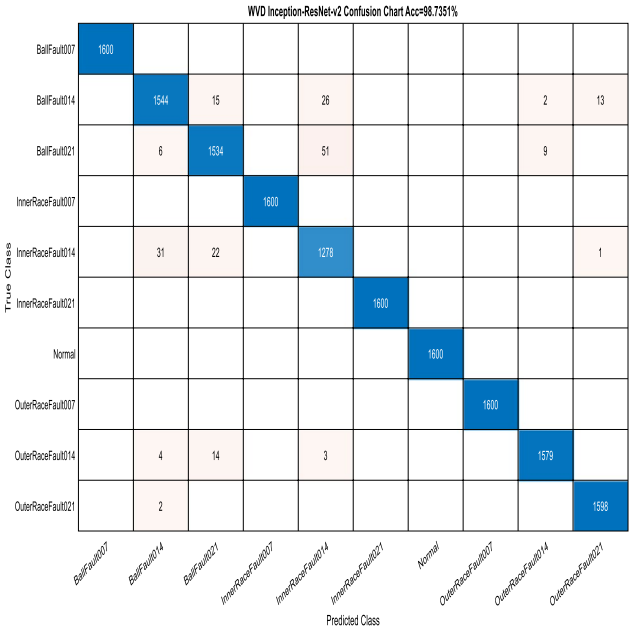



## 5. Wigner Ville Distribution

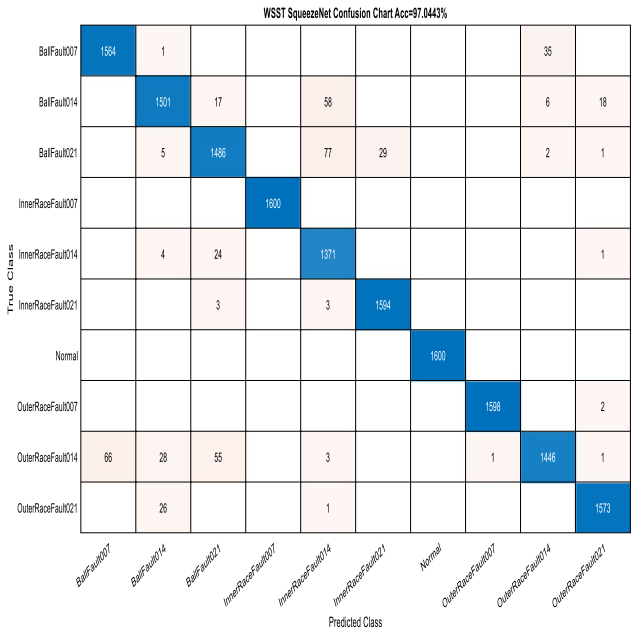


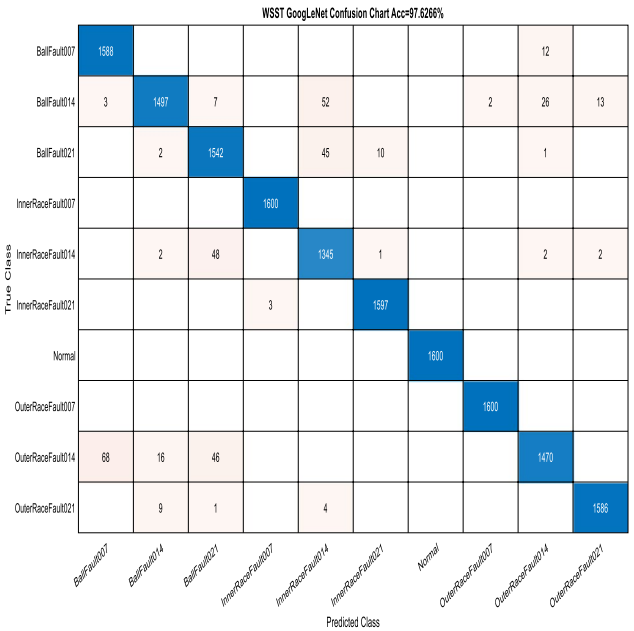



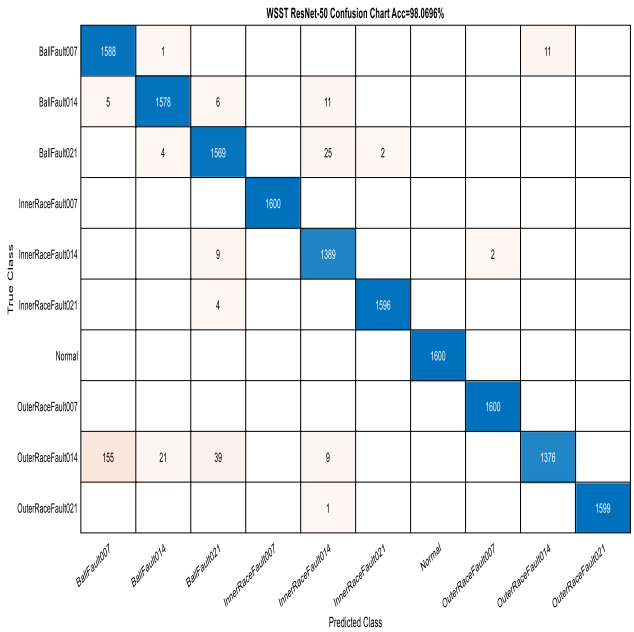



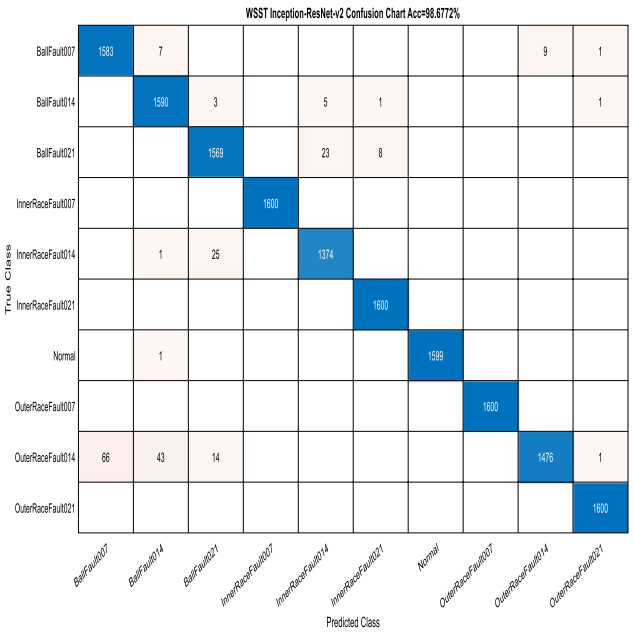



## 6. Wavelet Synchrosqueezing Transform

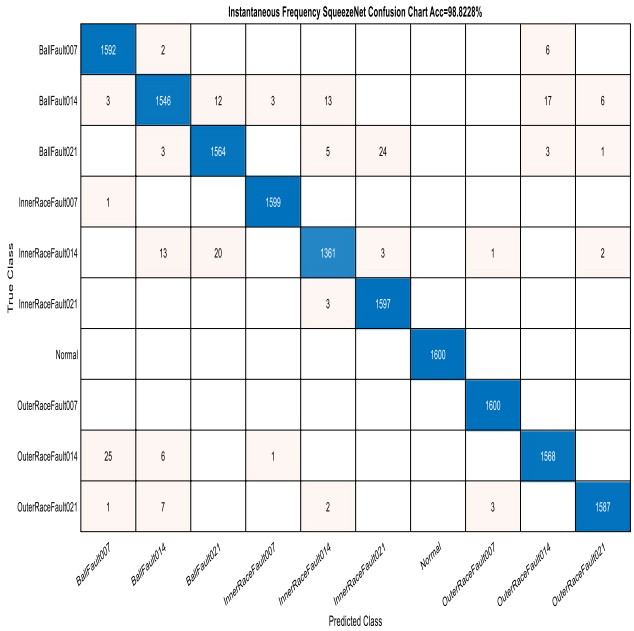


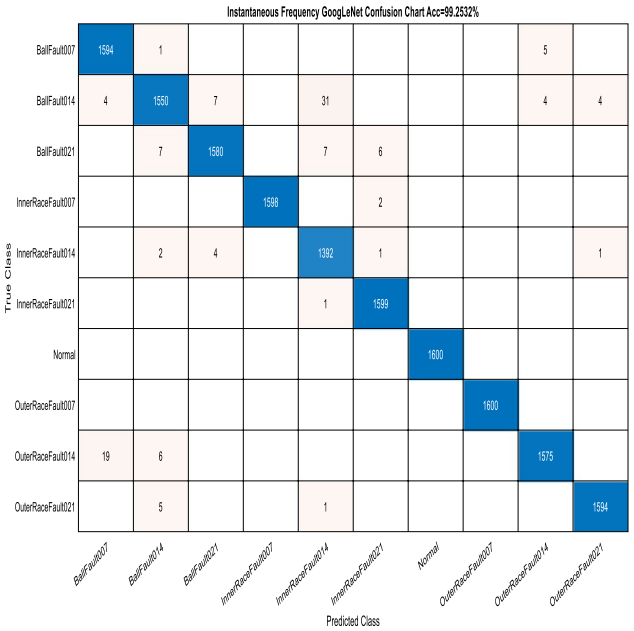



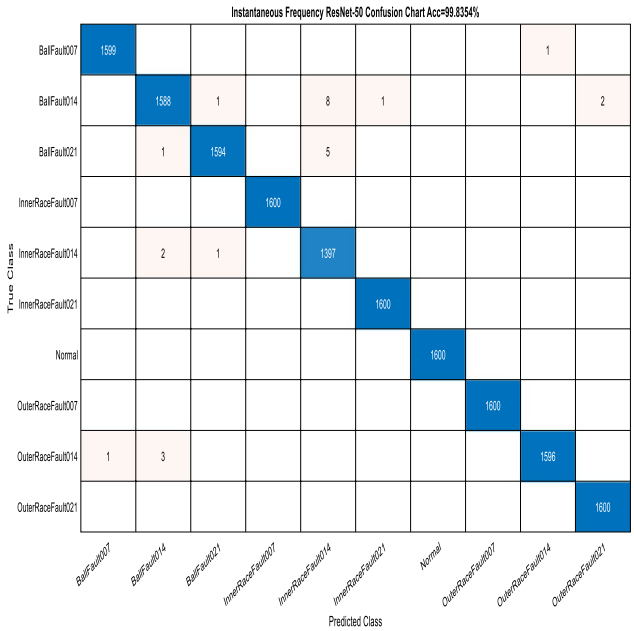



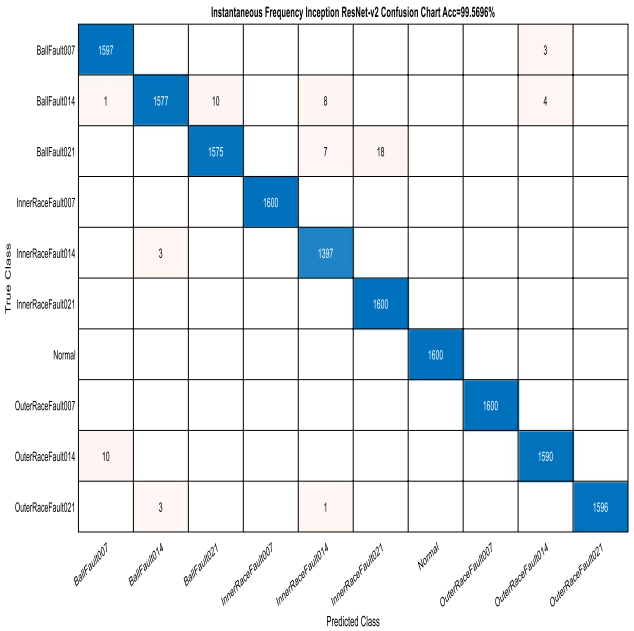



## 7. Instantaneous Frequency

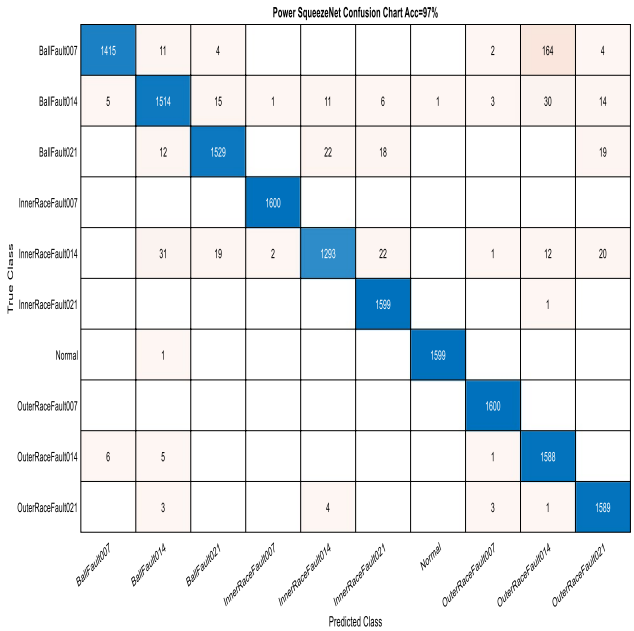


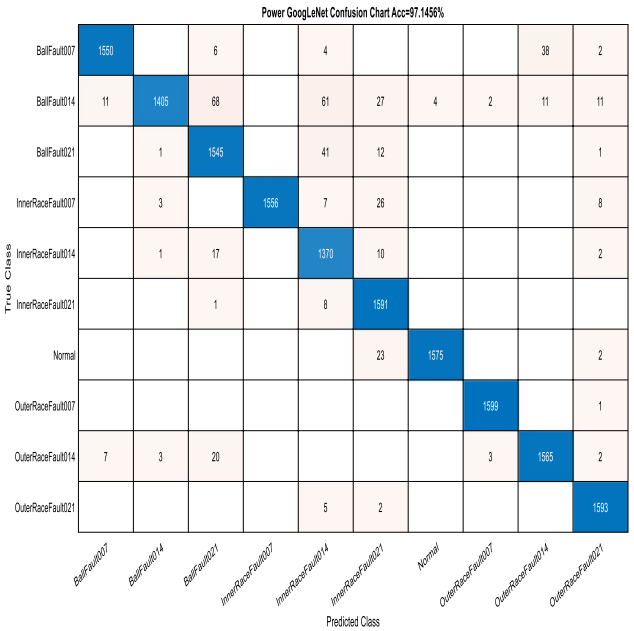



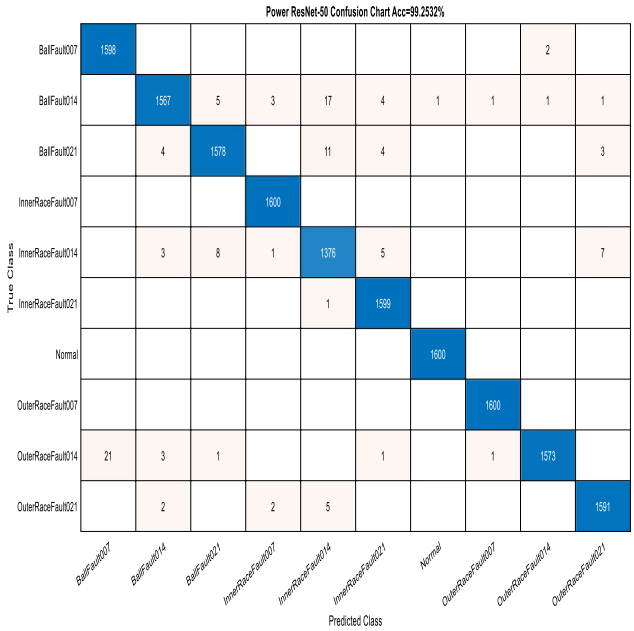



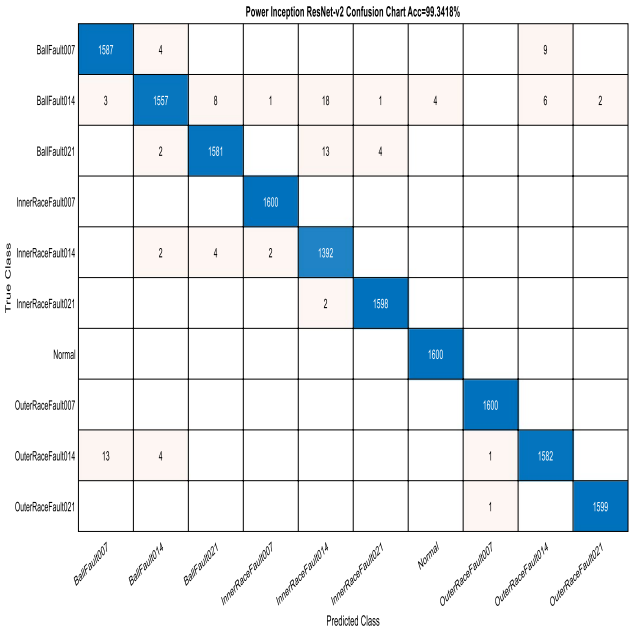



## 8. Power Transform

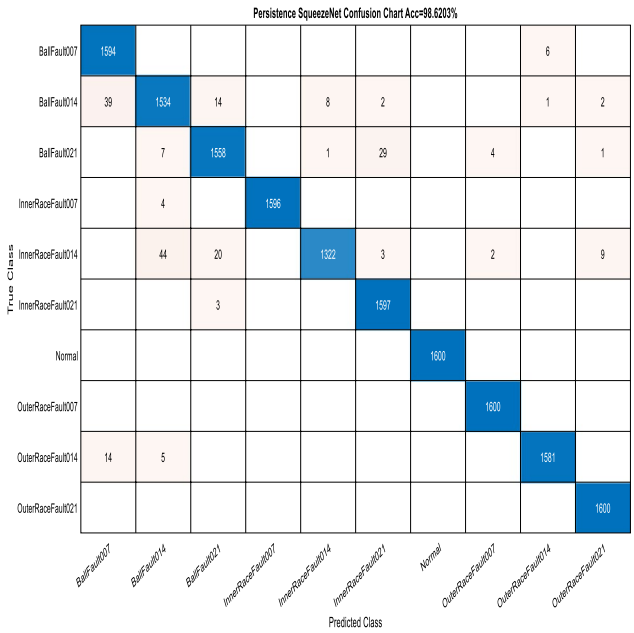


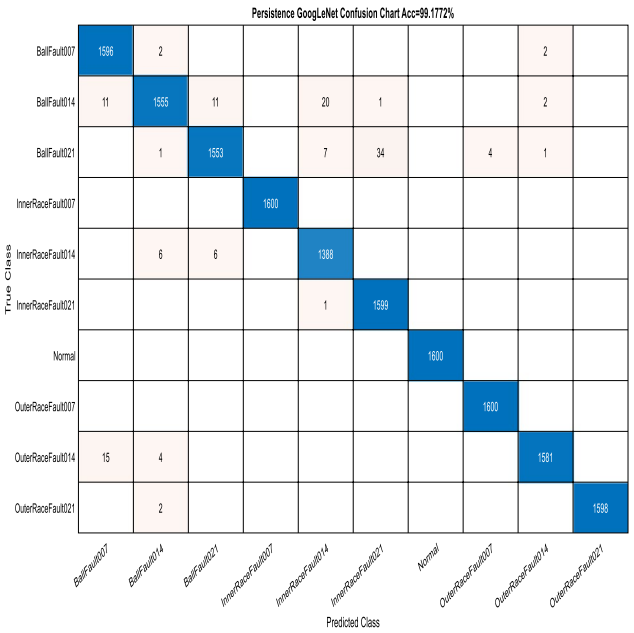



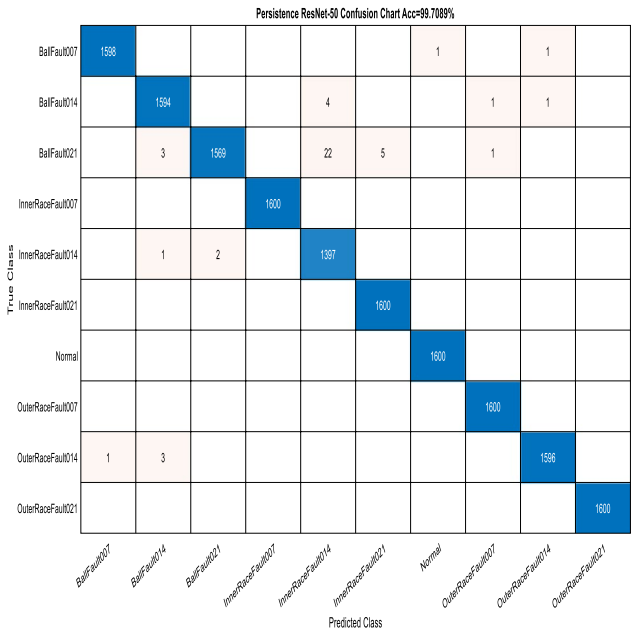



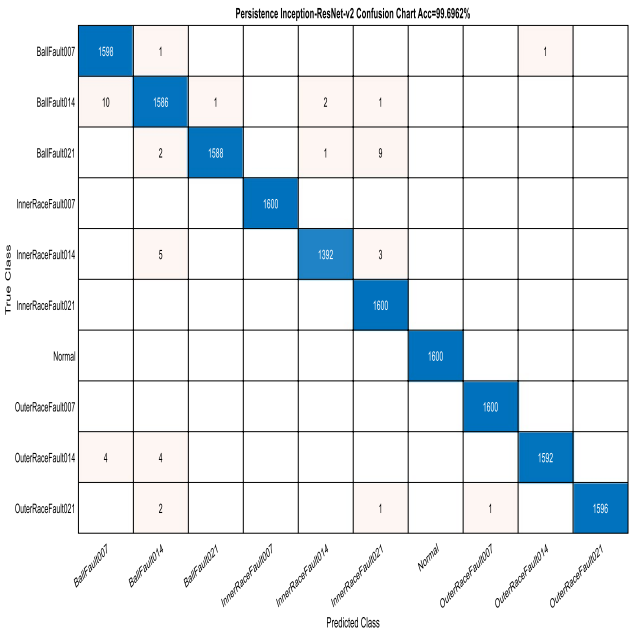



## 9. Persistence

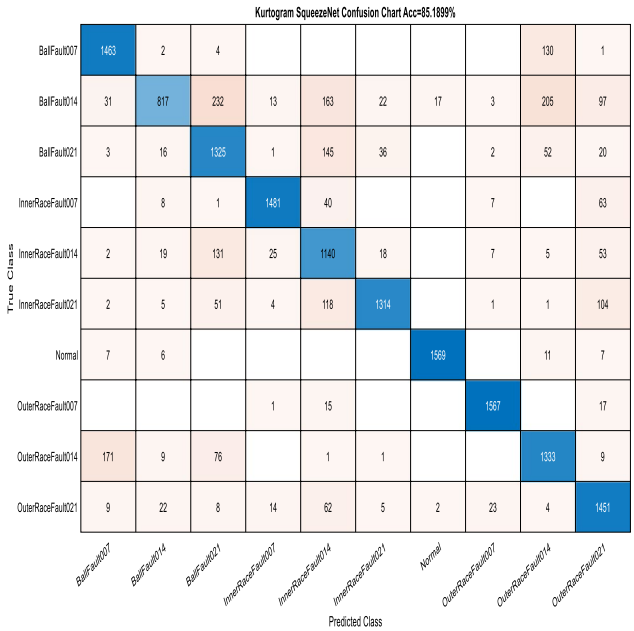


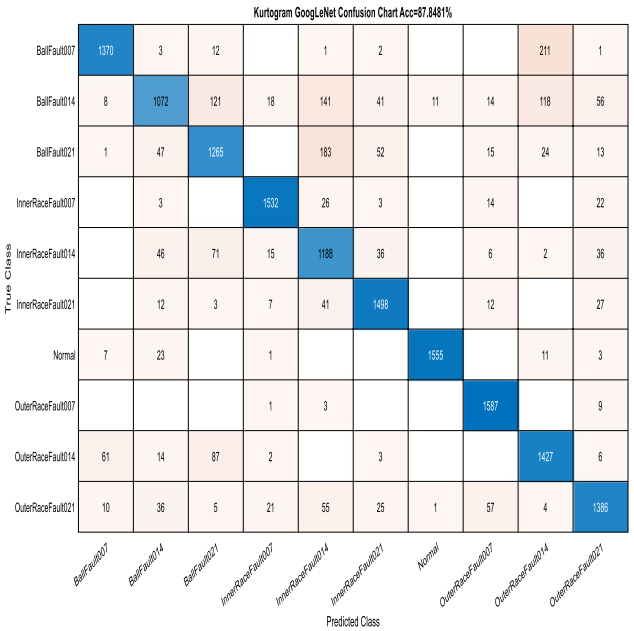



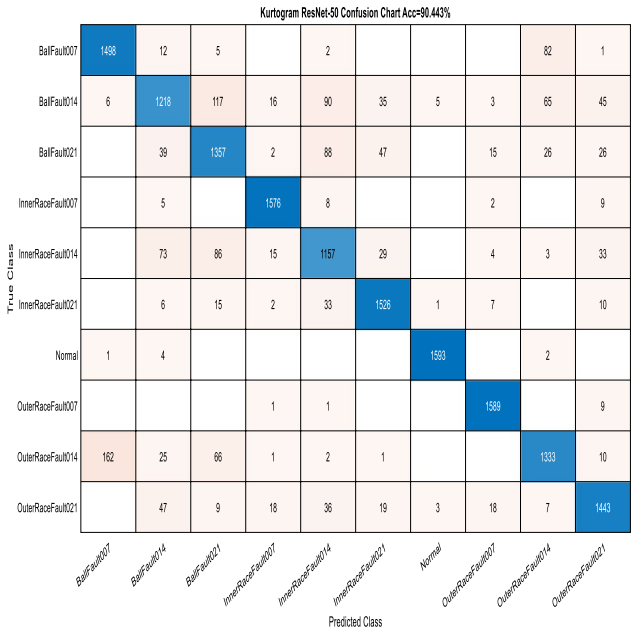



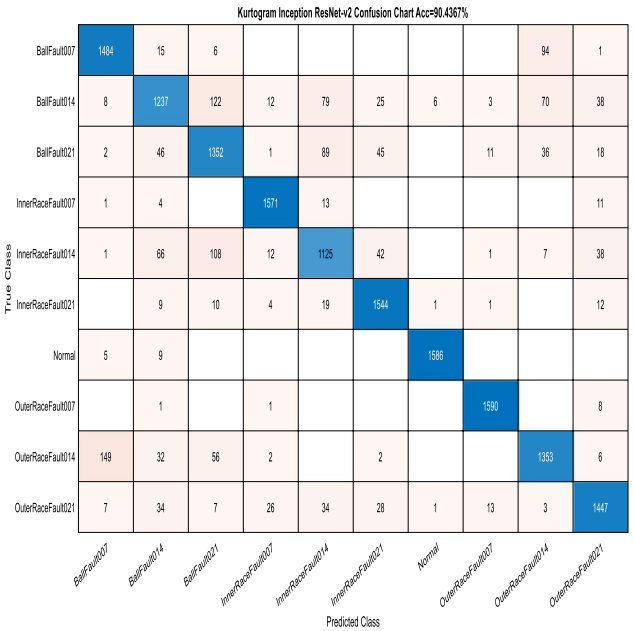



10. Kurtogram

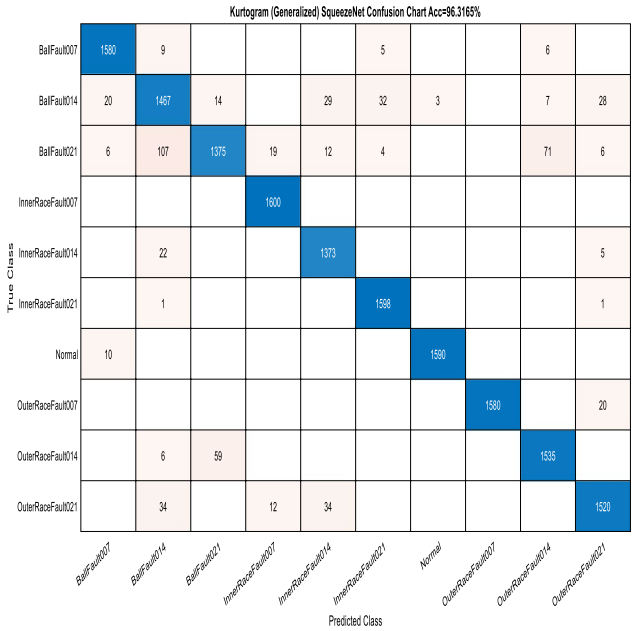


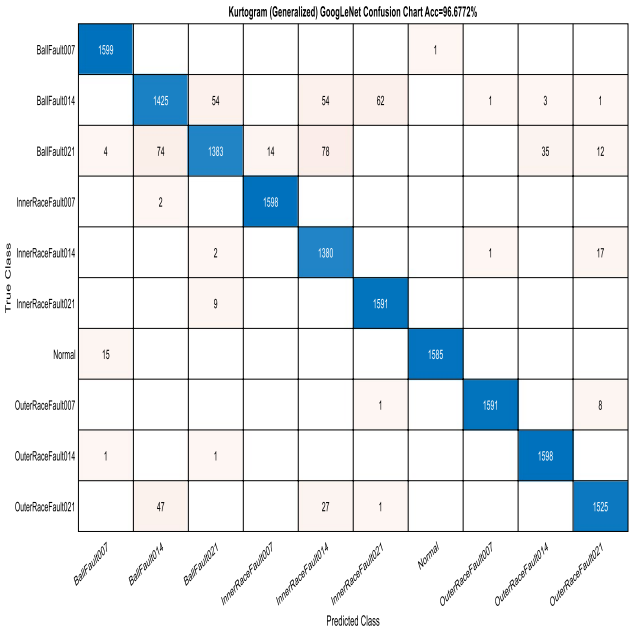



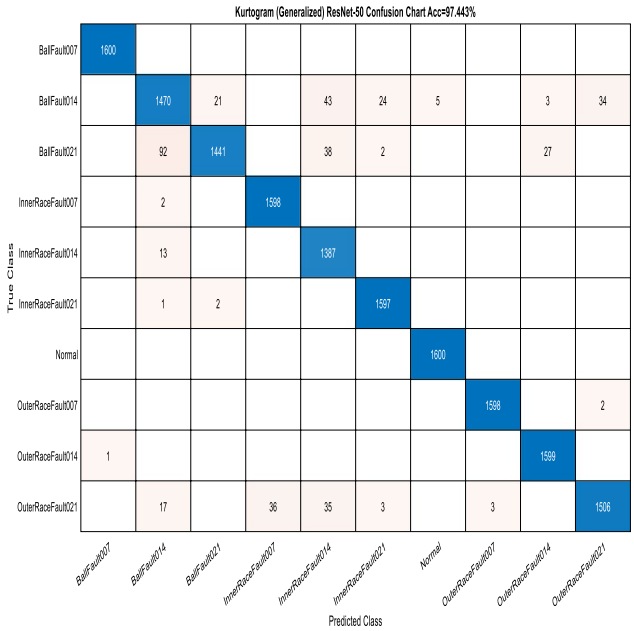



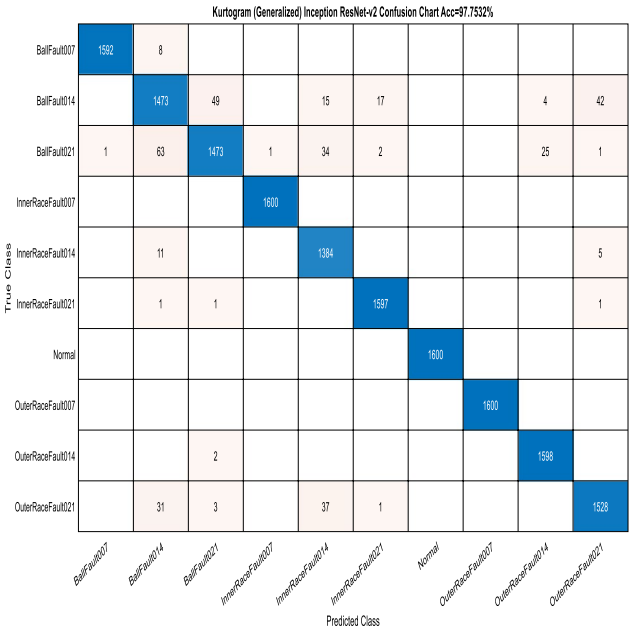



11. Kurtogram (Generalized)

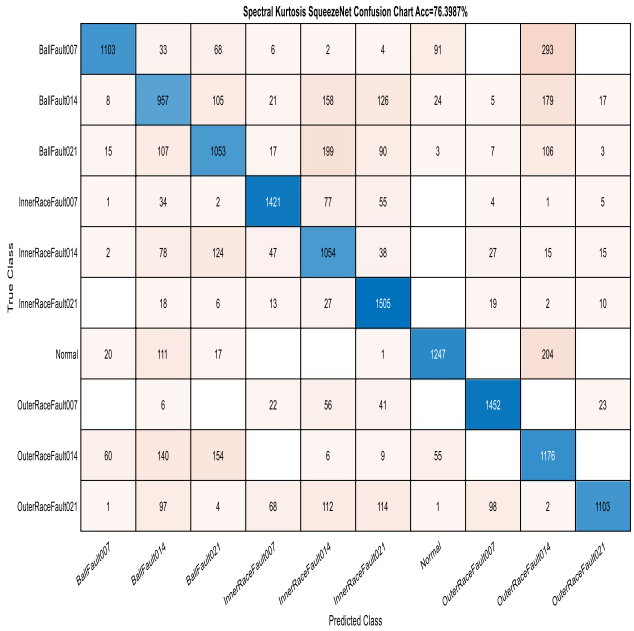


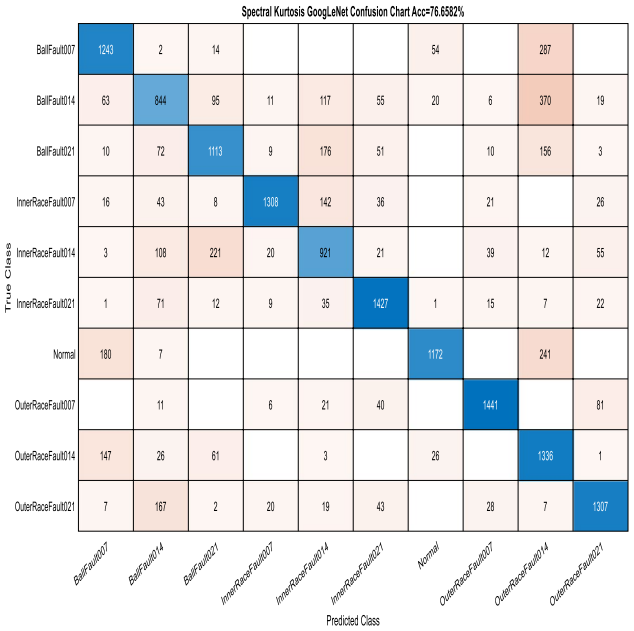



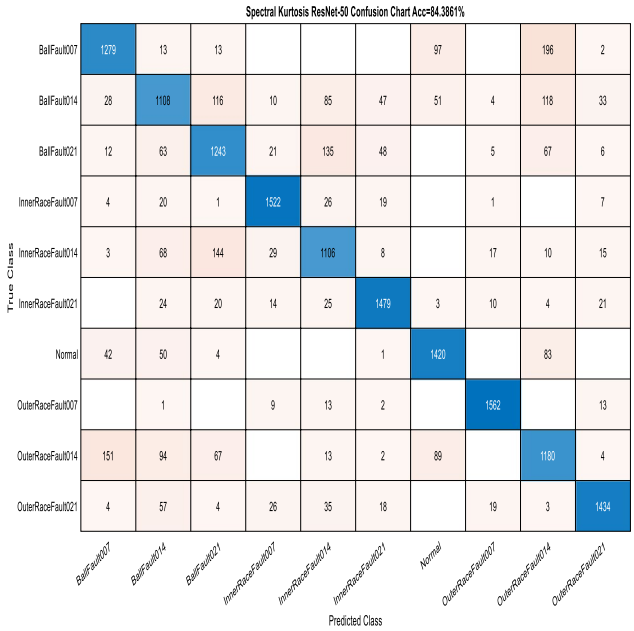



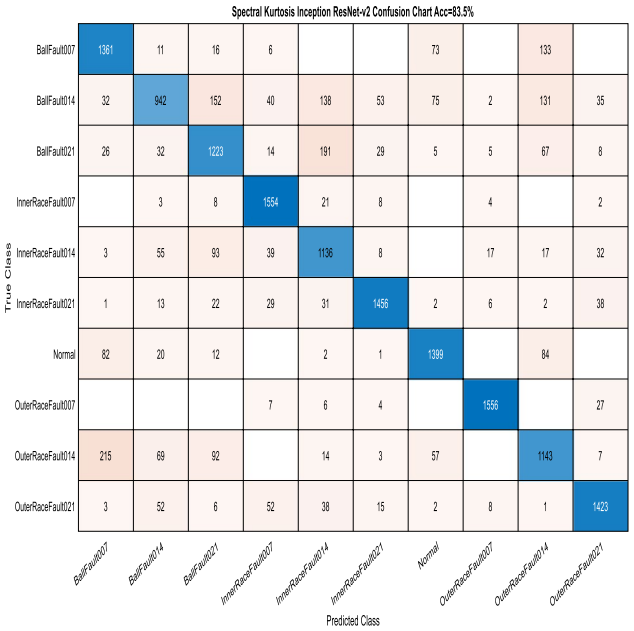



## 12. Spectral Kurtosis

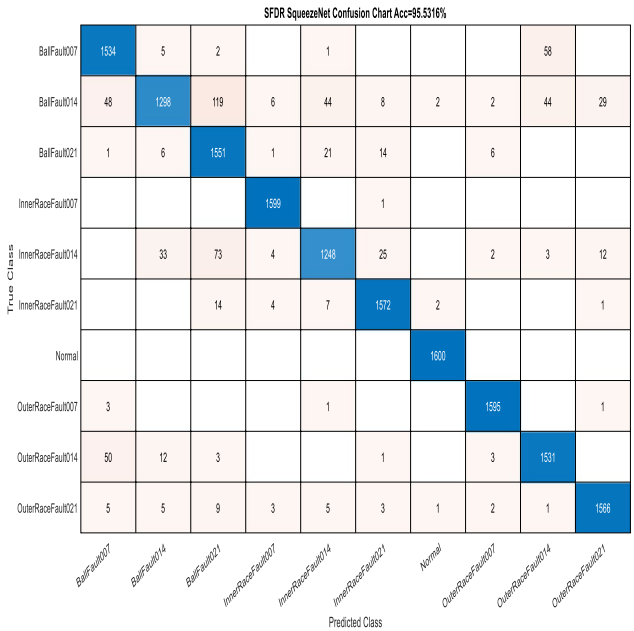


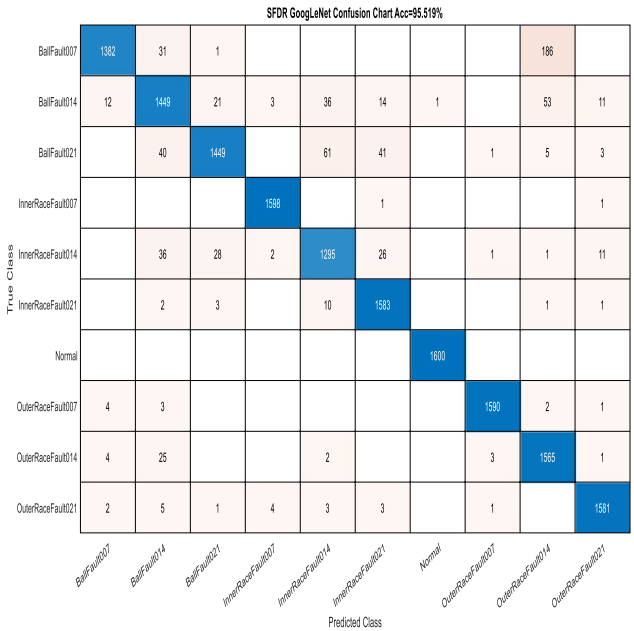



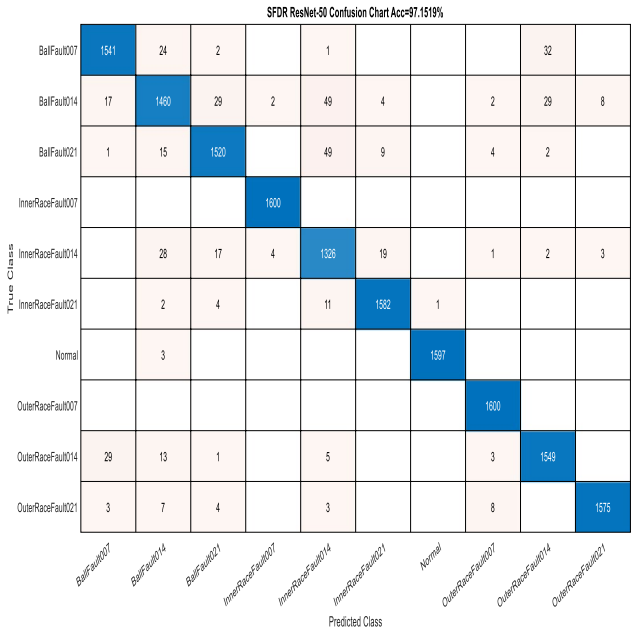



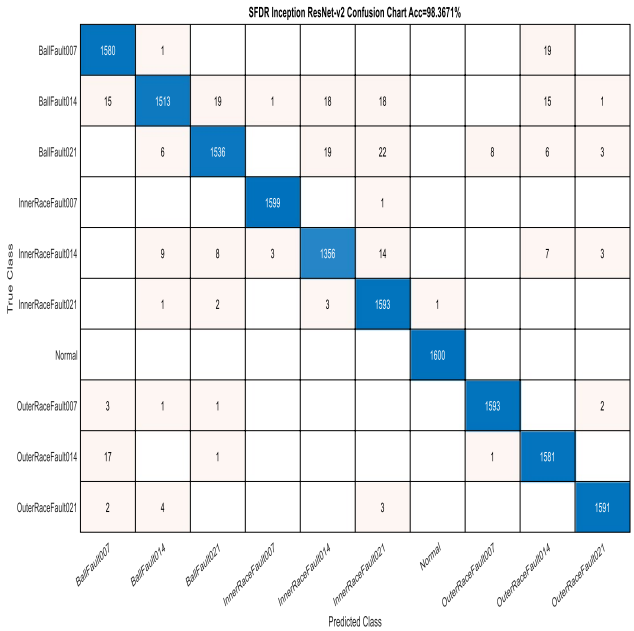



13. SFDR

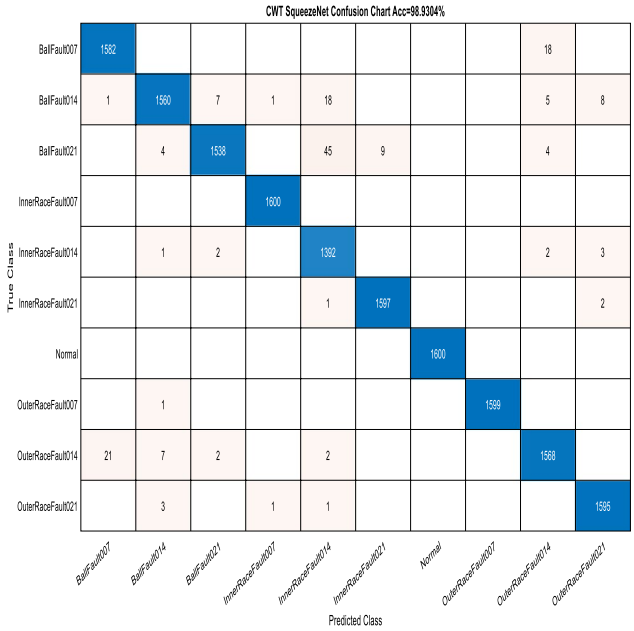

SFDR SqueezeNet Confusion Chart Acc=95.5316%
True Class
Predicted Class
BallFault007
BallFault014
BallFault021
InnerRaceFault007
InnerRaceFault014
InnerRaceFault021
Normal
OuterRaceFault007
OuterRaceFault014
OuterRaceFault021

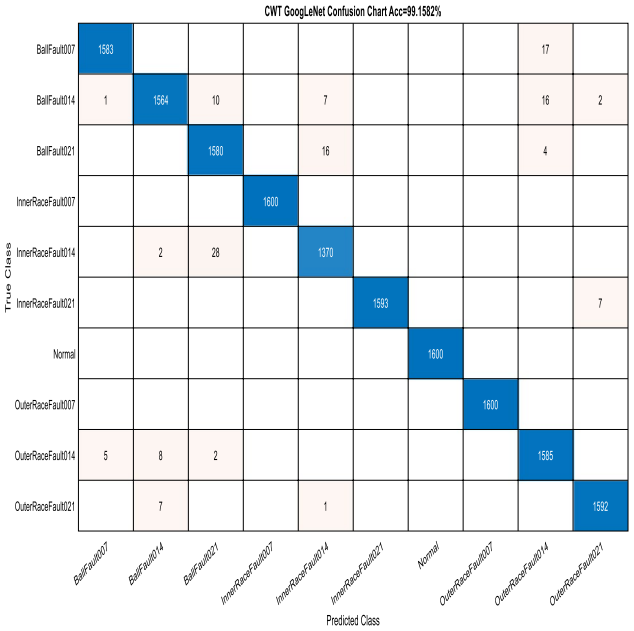

SFDR GoogLeNet Confusion Chart Acc=95.519%
True Class
Predicted Class
BallFault007
BallFault014
BallFault021
InnerRaceFault007
InnerRaceFault014
InnerRaceFault021
Normal
OuterRaceFault007
OuterRaceFault014
OuterRaceFault021

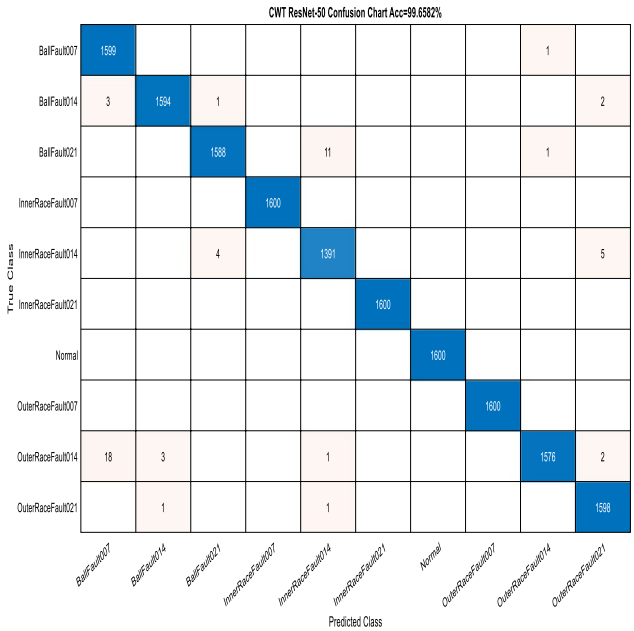

SFDR ResNet-50 Confusion Chart Acc=97.1519%
True Class
Predicted Class
BallFault007
BallFault014
BallFault021
InnerRaceFault007
InnerRaceFault014
InnerRaceFault021
Normal
OuterRaceFault007
OuterRaceFault014
OuterRaceFault021

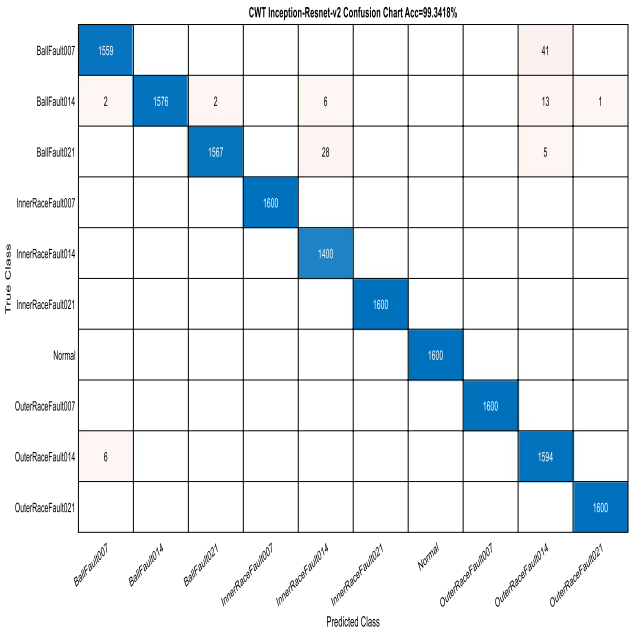

SFDR Inception ResNet-v2 Confusion Chart Acc=98.3671%
True Class
Predicted Class
BallFault007
BallFault014
BallFault021
InnerRaceFault007
InnerRaceFault014
InnerRaceFault021
Normal
OuterRaceFault007
OuterRaceFault014
OuterRaceFault021

14. Continuous Wavelet Transform

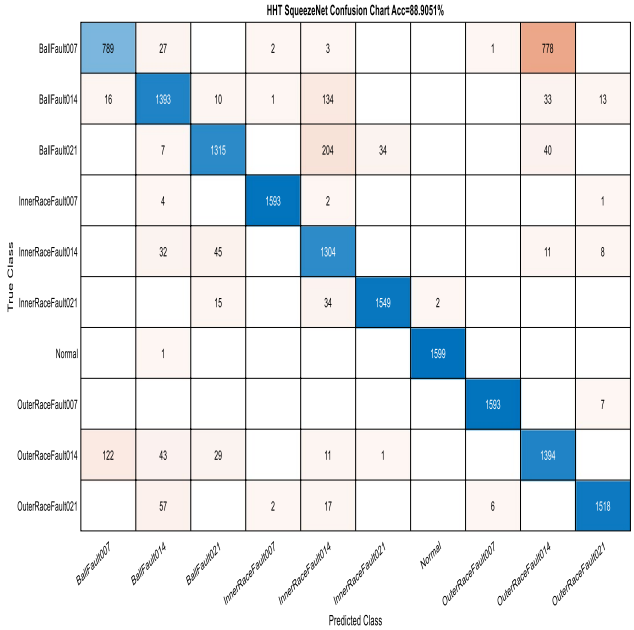


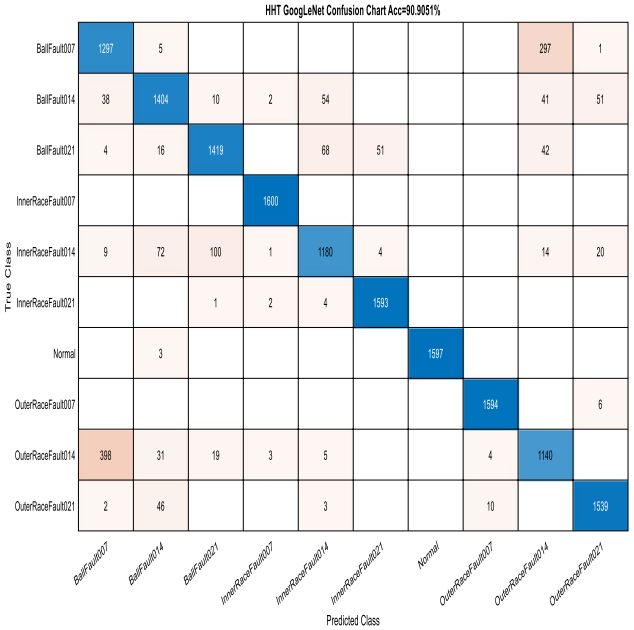



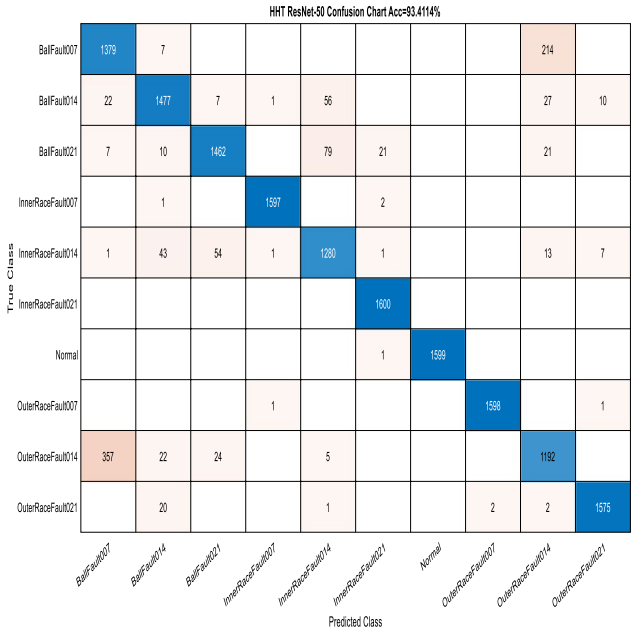



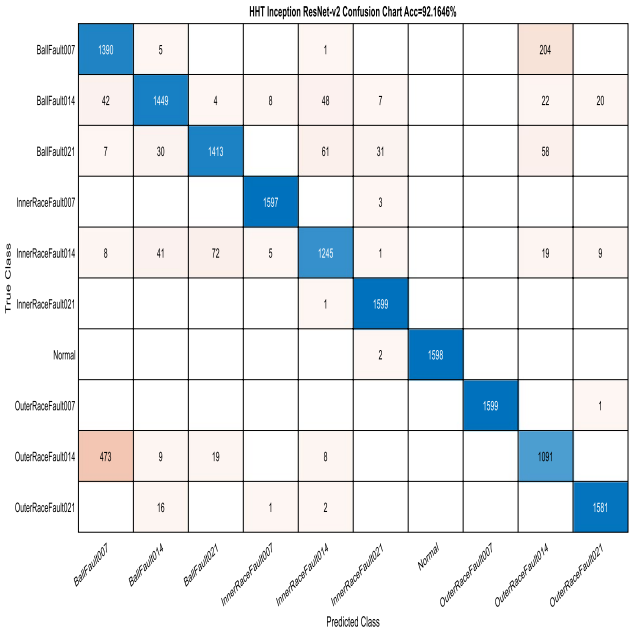



15. Hilbert-Huang Transform

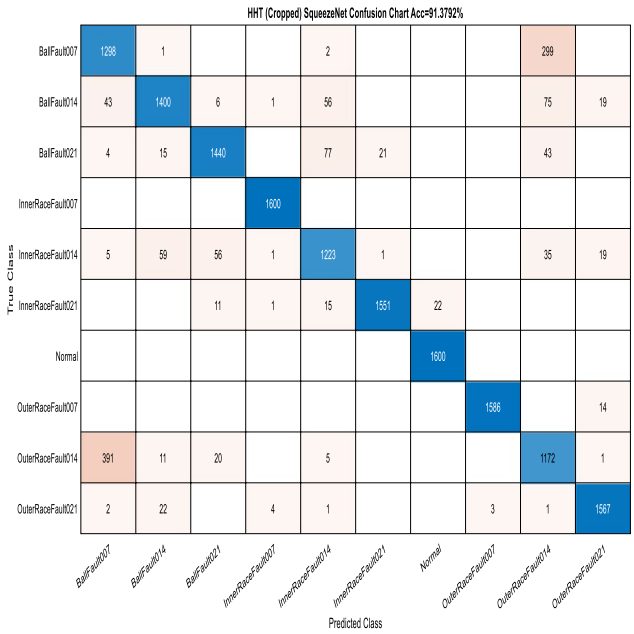


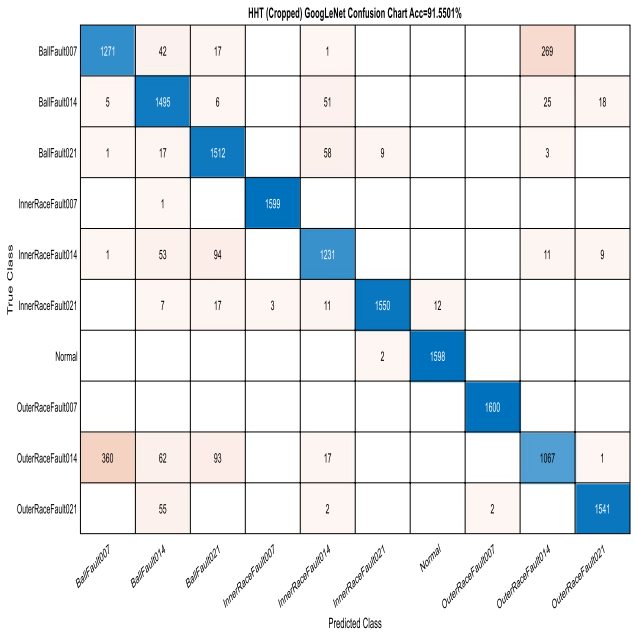



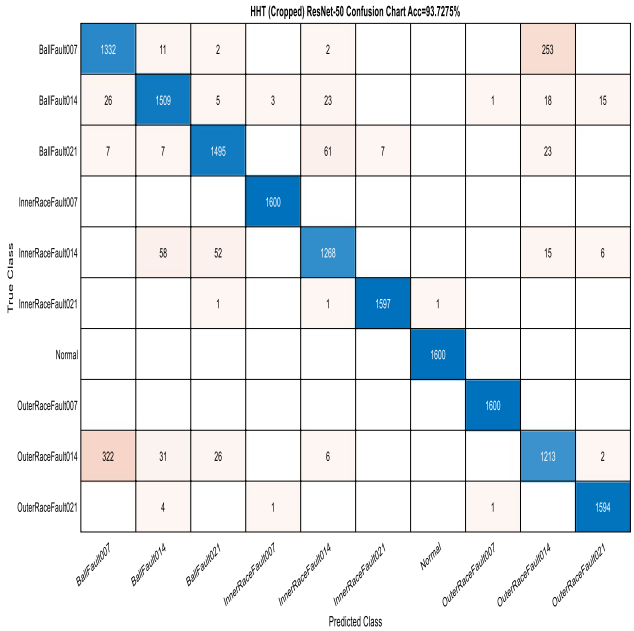



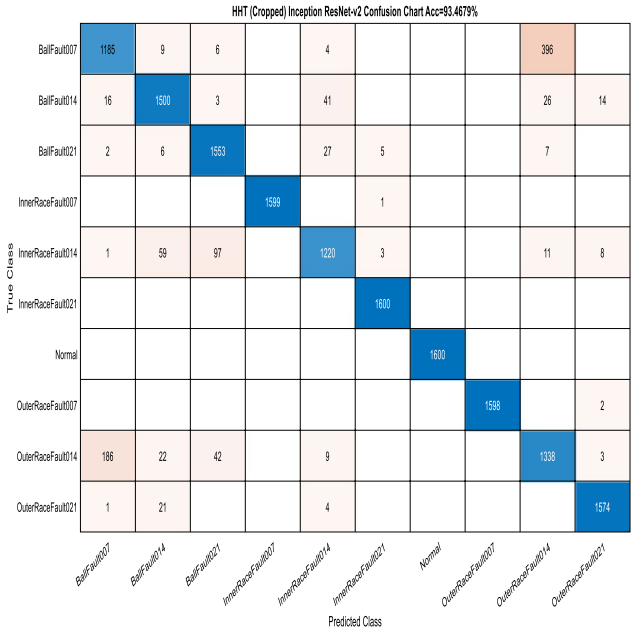



16. Hilbert-Huang Transform (Cropped)

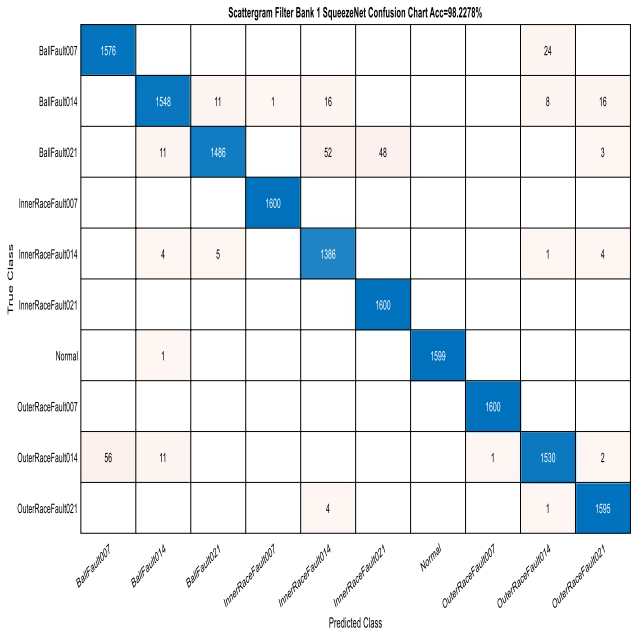


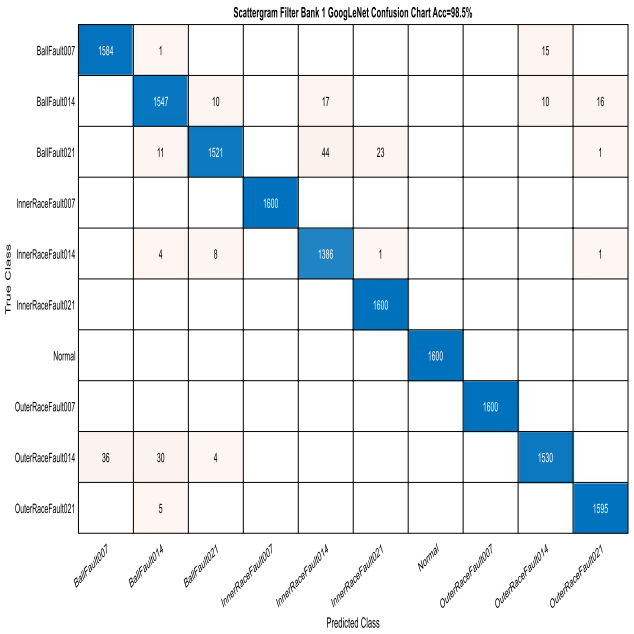



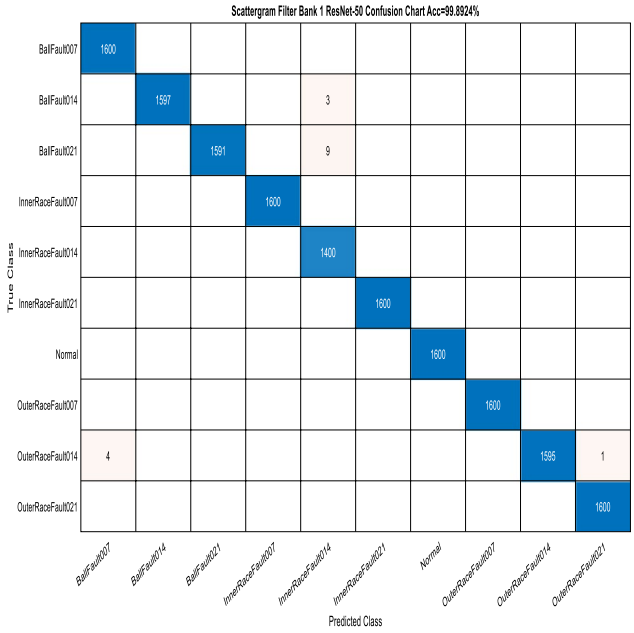



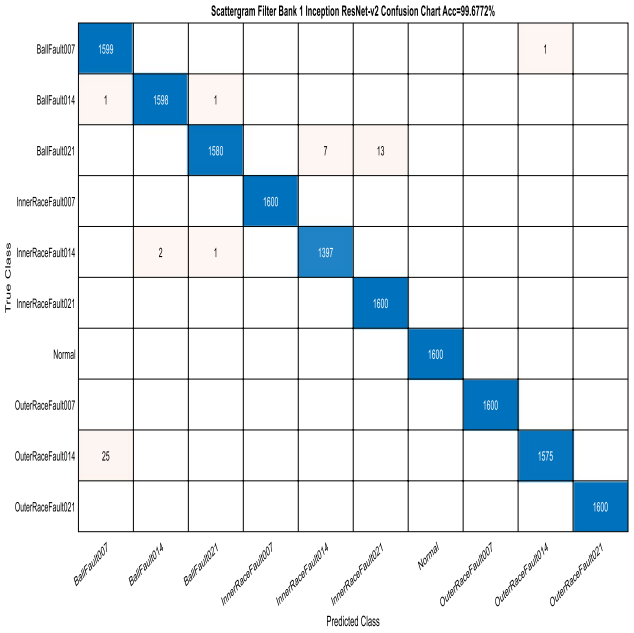



17. Scattergram Filter Bank 1

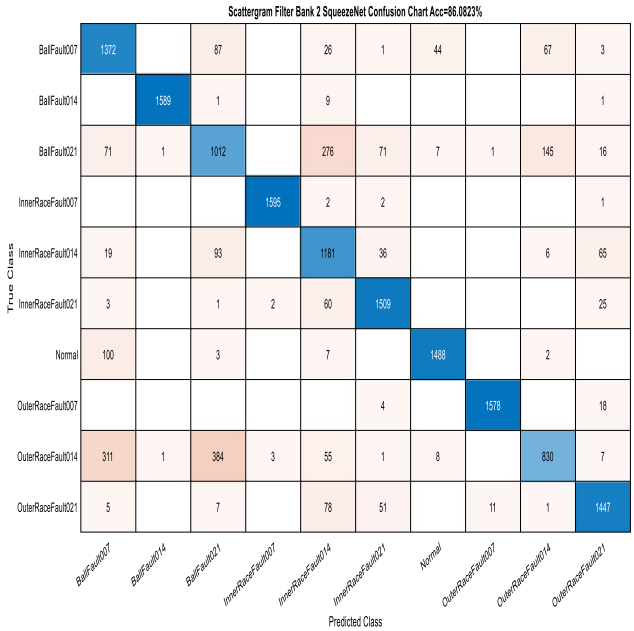


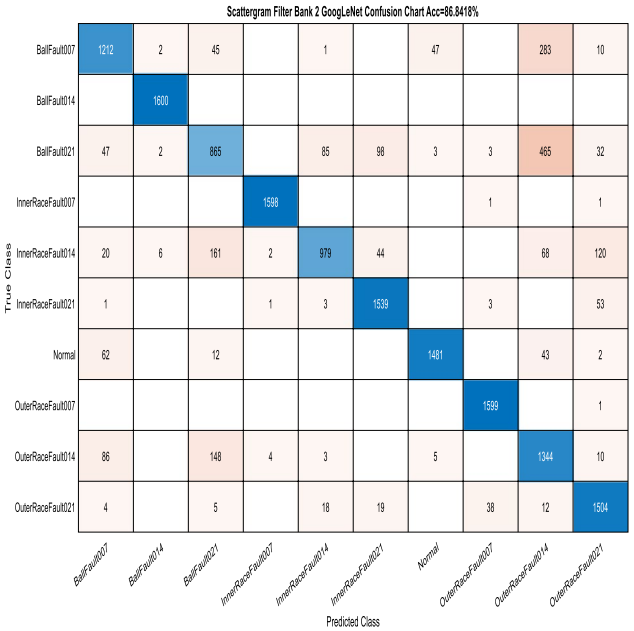



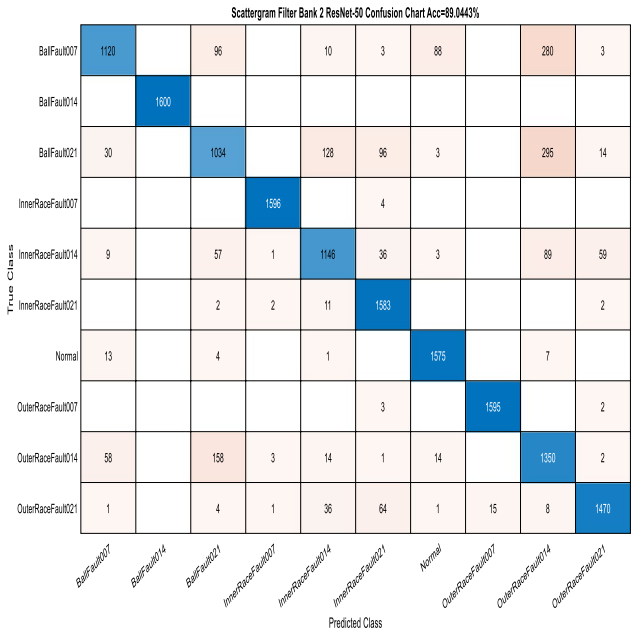



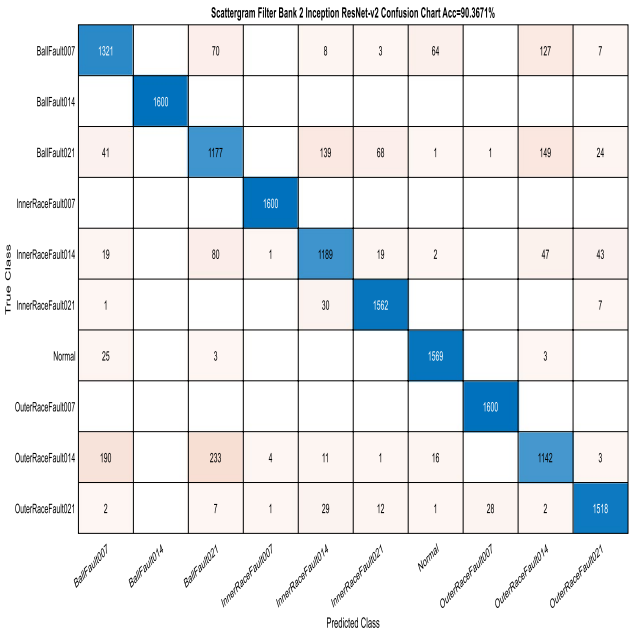



18. Scattergram Filter Bank 2

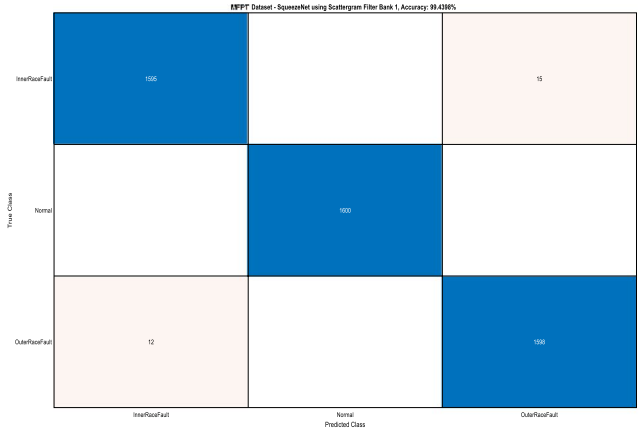


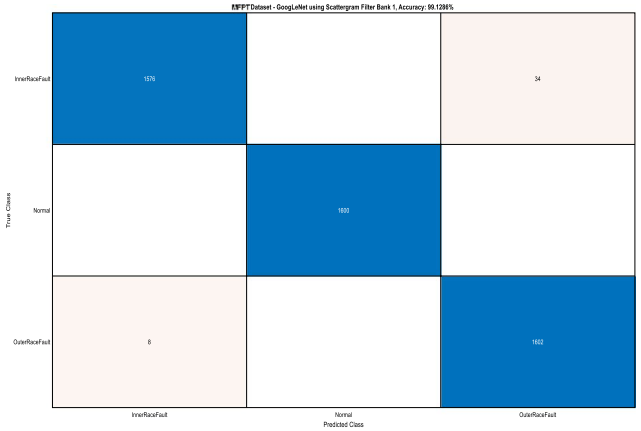



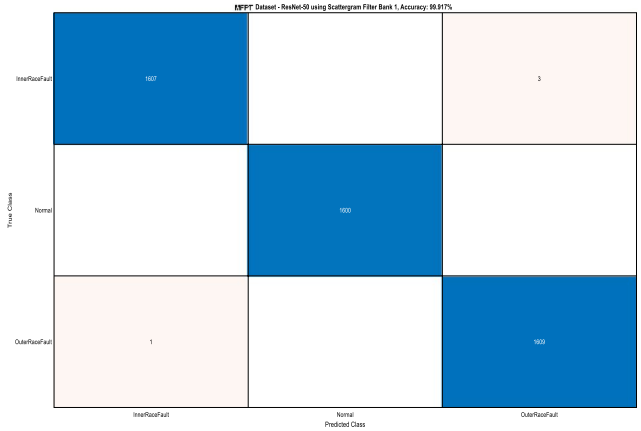



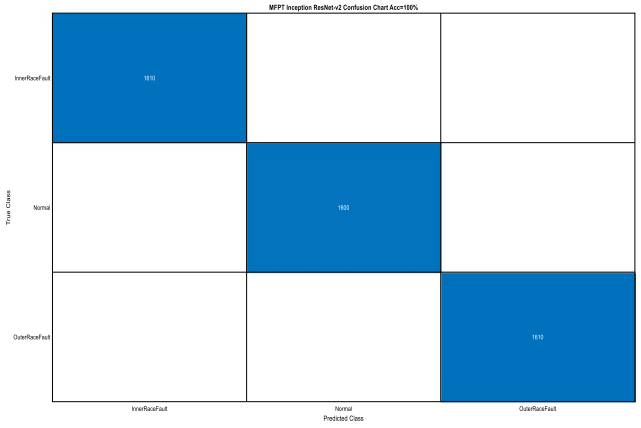



19. MFPT Dataset – Scattergram Filter Bank 1



## Declarations

**Conflict of Interests** All authors declare that they have no conflicts of interest.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

## References

Abdelkader, R., Kaddour, A., & Derouiche, Z. (2018). Enhancement of rolling bearing fault diagnosis based on improvement of empirical mode decomposition denoising method. *The International Journal of Advanced Manufacturing Technology, 97*(5), 3099–3117.

Abed, S. T., Dallalbashi, Z. E., & Taha, F. A. (2012). Studying the effect of window type on power spectrum based on MatLab. *Tikrit Journal of Engineering Sciences, 19*(2), 63–70.

An, X., & Jiang, D. (2014). Bearing fault diagnosis of wind turbine based on intrinsic time-scale decomposition frequency spectrum. *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability, 228*(6), 558–566.

Andén, J., & Mallat, S. (2014). Deep scattering spectrum. *IEEE Transactions on Signal Processing, 62*(16), 4114–4128.

Antoni, J. (2006). The spectral kurtosis: A useful tool for characterising non-stationary signals. *Mechanical Systems and Signal Processing, 20*(2), 282–307.

Antoni, J. (2007). Fast computation of the kurtogram for the detection of transient faults. *Mechanical Systems and Signal Processing, 21*(1), 108–124.

Baldassarre, F., Morín, D. G., & Rodés-Guirao, L. (2017). Deep koalarization: Image colorization using cnns and inception-resnet-v2. *arXiv preprint arXiv:1712.03400*.

Barbarossa, S., & Farina, A. (1992). Detection and imaging of moving objects with synthetic aperture radar. Part 2: Joint time-frequency analysis by Wigner-Ville distribution. In *IEE Proceedings F (Radar and Signal Processing)* (Vol. 139, No. 1, pp. 89–97). IET Digital Library.

Bechhoefer, E. (2013). Condition based maintenance fault database for testing diagnostics and prognostic algorithms. MFPT Data.

Boashash, B. (1992). Estimating and interpreting the instantaneous frequency of a signal. I. Fundamentals. *Proceedings of the IEEE, 80*(4), 520–538.

Bossau, J. C. (2020). *The detection and quantification of wave slamming from full-scale measurements on a polar vessel* (Doctoral dissertation, Stellenbosch: Stellenbosch University).

Büssow, R. (2007). An algorithm for the continuous Morlet wavelet transform. *Mechanical Systems and Signal Processing, 21*(8), 2970–2979.

Cao, G., Zhang, K., Zhou, K., Pan, H., Xu, Y., & Liu, J. (2020). A feature transferring fault diagnosis based on WPDR, FSWT and GoogLeNet. In *2020 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)* (pp. 1–6). IEEE.

Case Western Reserve University. (2005). Bearing Data Center. Available on: https://engineering.case.edu/bearingdatacenter. Accessed 6 Dec 2022. Archived.

Case Western Reserve University. (2005). Bearing Data Center: Download Data Files. Available on: https://engineering.case.edu/bearingdatacenter. Accessed 6 Dec. 2022. Archived.

Cen, J., Yang, Z., Liu, X., Xiong, J., & Chen, H. (2022). A review of data-driven machinery fault diagnosis using machine learning algorithms. *Journal of Vibration Engineering & Technologies*, 1–27.

Cerrada, M., Sánchez, R. V., Li, C., Pacheco, F., Cabrera, D., de Oliveira, J. V., & Vásquez, R. E. (2018). A review on data-driven fault severity assessment in rolling bearings. *Mechanical Systems and Signal Processing, 99*, 169–196.

Daubechies, I., Lu, J., & Wu, H. T. (2011). Synchrosqueezed wavelet transforms: An empirical mode decomposition-like tool. *Applied and Computational Harmonic Analysis, 30*(2), 243–261.

Deveci, B. U., Celtikoglu, M., Alp, T., Albayrak, Ö., Ünal, P., & Kirci, P. (2021). A comparison of deep transfer learning methods on bearing fault detection. In *FiCloud* (pp. 285–292).

Dhar, P., Dutta, S., & Mukherjee, V. (2021). Cross-wavelet assisted convolution neural network (AlexNet) approach for phonocardiogram signals classification. *Biomedical Signal Processing and Control, 63*, 102142.

Duan, J., Shi, T., Zhou, H., Xuan, J., & Wang, S. (2021). A novel ResNet-based model structure and its applications in machine health monitoring. *Journal of Vibration and Control, 27*(9–10), 1036–1050.

Gaikwad, A. S., & El-Sharkawy, M. (2018). Pruning convolution neural network (squeezenet) using taylor expansion-based criterion. In *2018 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)* (pp. 1–5). IEEE.

Grover, C., & Turk, N. (2022). A novel fault diagnostic system for rolling element bearings using deep transfer learning on bispectrum contour maps. *Engineering Science and Technology, an International Journal, 31*, 101049.

Hakim, M., Omran, A. A. B., Ahmed, A. N., Al-Waily, M., & Abdellatif, A. (2022). A systematic review of rolling bearing fault diagnoses based on deep learning and transfer learning: Taxonomy, overview, application, open challenges, weaknesses and recommendations. *Ain Shams Engineering Journal*, 101945.

Han, T., Tian, Z., Yin, Z., & Tan, A. C. (2020). Bearing fault identification based on convolutional neural network by different input modes. *Journal of the Brazilian Society of Mechanical Sciences and Engineering, 42*(9), 1–10.

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).

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).

Hendriks, J. (2021). *Vibrations-Based Machine Fault Diagnosis and Prognosis Using Convolutional Neural Networks* (Doctoral dissertation, Université d'Ottawa/University of Ottawa).

Huang, N. E. (2014). Introduction to the Hilbert–Huang transform and its related mathematical problems. In *Hilbert–Huang transform and its applications* (pp. 1–26).

Huang, H., Baddour, N., & Liang, M. (2018). Short-time kurtogram for bearing fault feature extraction under time-varying speed conditions. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference* (Vol. 51852, p. V008T10A035). American Society of Mechanical Engineers.

Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning* (pp. 448–456). PMLR.

Jiao, J., Zhao, M., Lin, J., & Liang, K. (2020). A comprehensive review on convolutional neural network in machine fault diagnosis. *Neurocomputing, 417*, 36–63.

Khan, A., Sohail, A., Zahoora, U., & Qureshi, A. S. (2020). A survey of the recent architectures of deep convolutional neural networks. *Artificial Intelligence Review, 53*(8), 5455–5516.

Khankari, J., Yu, Y., Ouyang, J., Hussein, R., Do, H. M., Heit, J. J., & Zaharchuk, G. (2022). Automated detection of arterial landmarks and vascular occlusions in patients with acute stroke receiving digital subtraction angiography using deep learning. *Journal of NeuroInterventional Surgery*.

Kong, G., & Koivunen, V. (2019). Radar waveform recognition using Fourier-based synchrosqueezing transform and CNN. In *2019 IEEE 8th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)* (pp. 664–668). IEEE.

Kumar, A., Gandhi, C. P., Zhou, Y., Kumar, R., & Xiang, J. (2020). Improved deep convolution neural network (CNN) for the identification of defects in the centrifugal pump using acoustic images. *Applied Acoustics, 167*, 107399.

Kutz, J. N. (2013). *Data-driven modeling & scientific computation: Methods for complex systems & big data*. Oxford University Press.

Lee, C. Y., & Le, T. A. (2021). Identifying faults of rolling element based on persistence spectrum and convolutional neural network with ResNet structure. *IEEE Access, 9*, 78241–78252.

Li, Y., & Zheng, X. (2008). Spectral decomposition using Wigner-Ville distribution with applications to carbonate reservoir characterization. *The Leading Edge, 27*(8), 1050–1057.

Li, G., Deng, C., Wu, J., Xu, X., Shao, X., & Wang, Y. (2019a). Sensor data-driven bearing fault diagnosis based on deep convolutional neural networks and S-transform. *Sensors, 19*(12), 2750.

Li, M., Wei, Q., Wang, H., & Zhang, X. (2019b). Research on fault diagnosis of time-domain vibration signal based on convolutional neural networks. *Systems Science & Control Engineering, 7*(3), 73–81.

Liang, J. (2020). Image classification based on RESNET. In *Journal of Physics: Conference Series* (Vol. 1634, No. 1, p. 012110). IOP Publishing.

Lilly, J. M. (2017). Element analysis: A wavelet-based method for analysing time-localized events in noisy time series. *Proceedings of the Royal Society a: Mathematical, Physical and Engineering Sciences, 473*(2200), 20160776.

Lilly, J. M., & Olhede, S. C. (2008). Higher-order properties of analytic wavelets. *IEEE Transactions on Signal Processing, 57*(1), 146–160.

Liu, G., Shen, W., Gao, L., & Kusiak, A. (2022). Knowledge transfer in fault diagnosis of rotary machines. *IET Collaborative Intelligent Manufacturing, 4*(1), 17–34.

Liu, C., He, D., Chen, Y., & Luo, Z. (2021). Rolling bearing fault diagnosis of train running gear based on optimized deep residual network. In *2021 5th International Conference on Automation, Control and Robots (ICACR)* (pp. 168–172). IEEE.

Mallat, S. (1999). *A wavelet tour of signal processing*. Elsevier.

Mallat, S. (2012). Group invariant scattering. *Communications on Pure and Applied Mathematics, 65*(10), 1331–1398.

Mathworks Inc. Pretrained Deep Neural Networks- MATLAB and Simulink. www.mathworks.com/help/deeplearning/ug/pretrained-convolutional-neural-networks.html. Accessed 6 Dec. 2022. Archived.

MathWorks. Estimate Instantaneous Frequency – MATLAB Instfreq. Estimate Instantaneous Frequency–MATLAB Instfreq, www.mathworks.com, Accessed 6 Dec. 2022. https://www.mathworks.com/help/signal/ref/instfreq.html. Archived

Minervini, M., Hausman, S., & Frosini, L. (2021). Transfer learning technique for automatic bearing fault diagnosis in induction motors. In *2021 IEEE 13th International Symposium on Diagnostics for Electrical Machines, Power Electronics and Drives (SDEMPED)* (Vol. 1, pp. 186–192). IEEE.

Naranjo-Torres, J., Mora, M., Hernández-García, R., Barrientos, R. J., Fredes, C., & Valenzuela, A. (2020). A review of convolutional neural network applied to fruit image processing. *Applied Sciences, 10*(10), 3443.

Neupane, D., & Seok, J. (2020). Bearing fault detection and diagnosis using case western reserve university dataset with deep learning approaches: A review. *IEEE Access, 8*, 93155–93178.

Niyongabo, J., Zhang, Y., & Ndikumagenge, J. (2022). Bearing fault detection and diagnosis based on densely connected convolutional networks. *Acta Mechanica et Automatica*, 16(2).

Oberlin, T., Meignen, S., & Perrier, V. (2014). The Fourier-based synchrosqueezing transform. In *2014 IEEE international conference on acoustics, speech and signal processing (ICASSP)* (pp. 315–319). IEEE.

Pan, S. J., & Yang, Q. (2009). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering, 22*(10), 1345–1359.

Peng, Z., Chu, F., & He, Y. (2002). Vibration signal analysis and feature extraction based on reassigned wavelet scalogram. *Journal of Sound and Vibration, 253*(5), 1087–1100.

Rueda, A., & Krishnan, S. (2019). Augmenting dysphonia voice using Fourier-based synchrosqueezing transform for a CNN classifier. In *ICASSP 2019–2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 6415–6419). IEEE.

Schörkhuber, C., Klapuri, A., Holighaus, N., & Dörfler, M. (2014). A matlab toolbox for efficient perfect reconstruction time-frequency transforms with log-frequency resolution. In *Audio Engineering Society Conference: 53rd International Conference: Semantic Audio*. Audio Engineering Society.

Shao, S., McAleer, S., Yan, R., & Baldi, P. (2018). Highly accurate machine fault diagnosis using deep transfer learning. *IEEE Transactions on Industrial Informatics, 15*(4), 2446–2455.

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Singru, P., Krishnakumar, V., Natarajan, D., & Raizada, A. (2018). Bearing failure prediction using Wigner-Ville distribution, modified Poincare mapping and fast Fourier transform. *Journal of Vibroengineering, 20*(1), 127–137.

Sit, M., Demiray, B. Z., Xiang, Z., Ewing, G. J., Sermet, Y., & Demir, I. (2020). A comprehensive review of deep learning applications in hydrology and water resources. *Water Science and Technology, 82*(12), 2635–2670.

Sun, G., Gao, Y., Xu, Y., & Feng, W. (2020). Data-driven fault diagnosis method based on second-order time-reassigned multisynchrosqueezing transform and evenly mini-batch training. *IEEE Access, 8*, 120859–120869.

Sun, G., Hu, Y., Wu, B., & Zhou, H. (2021). Rolling bearing fault diagnosis method based on multisynchrosqueezing S transform and faster dictionary learning. *Shock and Vibration*, 2021.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., … & Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1–9).

Szegedy, C., Ioffe, S., Vanhoucke, V., & Alemi, A. A. (2017). Inception-v4, inception-resnet and the impact of residual connections on learning. In *Thirty-first AAAI conference on artificial intelligence*.

Tama, B. A., Vania, M., Lee, S., & Lim, S. (2022). Recent advances in the application of deep learning for fault diagnosis of rotating machinery using vibration signals. *Artificial Intelligence Review*, 1–43.

Voulodimos, A., Doulamis, N., Doulamis, A., & Protopapadakis, E. (2018). Deep learning for computer vision: A brief review. *Computational Intelligence and Neuroscience*, 2018.

Wang, D., Guo, Q., Song, Y., Gao, S., & Li, Y. (2019). Application of multiscale learning neural network based on CNN in bearing fault diagnosis. *Journal of Signal Processing Systems, 91*(10), 1205–1217.

Wang, Z., Shangguan, W., Peng, C., & Cai, B. (2022). A fault diagnosis method based on data feature reconstruction and deep transfer learning. In *2022 IEEE Asia-Pacific Conference on Image Processing, Electronics and Computers (IPEC)* (pp. 1–5). IEEE.

Wei, H., Zhang, Q., Shang, M., & Gu, Y. (2021). Extreme learning Machine-based classifier for fault diagnosis of rotating Machinery using a residual network and continuous wavelet transform. *Measurement, 183*, 109864.

Wen, L., Li, X., Li, X., & Gao, L. (2019). A new transfer learning based on VGG-19 network for fault diagnosis. In *2019 IEEE 23rd international conference on computer supported cooperative work in design (CSCWD)* (pp. 205–209). IEEE.

What Is a Spectrogram? | Pacific Northwest Seismic Network. Pacific Northwest Seismic Network. Pnsn.org. Accessed Dec 6, 2022. https://pnsn.org/spectrograms/what-is-a-spectrogram. Archived.

Xu, Y., Li, Z., Wang, S., Li, W., Sarkodie-Gyan, T., & Feng, S. (2021). A hybrid deep-learning model for fault diagnosis of rolling bearings. *Measurement, 169*, 108502.

Yin, S., Ding, S. X., Xie, X., & Luo, H. (2014). A review on basic data-driven approaches for industrial process monitoring. *IEEE Transactions on Industrial Electronics, 61*(11), 6418–6428.

Youngberg, J., & Boll, S. (1978). Constant-Q signal analysis and synthesis. In *ICASSP'78. IEEE International Conference on Acoustics, Speech, and Signal Processing* (Vol. 3, pp. 375–378). IEEE.

Yu, Y., Chen, Z. Y., Fang, X., Zhang, D. C., & Wang, Z. (2022). A research of a modified bearing fault diagnosis model based on deformable atrous convolution and squeeze-and-excitation aggregation. In *2022 13th International Conference on Reliability, Maintainability, and Safety (ICRMS)* (pp. 174–179). IEEE.

Yuan, L., Lian, D., Kang, X., Chen, Y., & Zhai, K. (2020). Rolling bearing fault diagnosis based on convolutional neural network and support vector machine. *IEEE Access, 8*, 137395–137406.

Zagoruyko, S., & Komodakis, N. (2016). Wide residual networks. *arXiv preprint arXiv:1605.07146*.

Zhang, Y., Xing, K., Bai, R., Sun, D., & Meng, Z. (2020a). An enhanced convolutional neural network for bearing fault diagnosis based on time–frequency image. *Measurement, 157*, 107667.

Zhang, S., Zhang, S., Wang, B., & Habetler, T. G. (2020b). Deep learning algorithms for bearing fault diagnostics—A comprehensive review. *IEEE Access, 8*, 29857–29881.

Zhang, J., Yi, S., Liang, G. U. O., Hongli, G. A. O., Xin, H. O. N. G., & Hongliang, S. O. N. G. (2020c). A new bearing fault diagnosis method based on modified convolutional neural networks. *Chinese Journal of Aeronautics, 33*(2), 439–447.

Zhang, H., Zhang, S., Wang, Z., Qiu, L., & Zhang, Y. (2022). Signals hierarchical feature enhancement method for CNN-based fault diagnosis. *Advances in Mechanical Engineering, 14*(9), 16878132221125020.

Zhang, S., Zhang, S., Wang, B., & Habetler, T. G. (2019a). Deep learning algorithms for bearing fault diagnostics-a review. In *2019a IEEE 12th International Symposium on Diagnostics for Electrical Machines, Power Electronics and Drives (SDEMPED)* (pp. 257–263). IEEE.

Zhang, S., Ye, F., Wang, B., & Habetler, T. G. (2019b). Semi-supervised learning of bearing anomaly detection via deep variational autoencoders. *arXiv preprint arXiv:1912.01096*.

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

**Bilgin Umut Deveci** is currently working as Research and Development Engineer at TEKNOPAR. He is a graduate of TOBB University of Economics and Technology, Department of Electrical and Electronics Engineering (2022). He conducts research on Industry 4.0 & IoT applications and develops turnkey solutions that include implementation of sensors, data acquisition hardware design, embedded coding, communication hardware & protocols, signal processing, feature engineering and artificial intelligence applications that include ML and DL. He has also worked on fields such as image processing, physical modeling, machine tools and electrical machines.

**Mert Çeltikoğlu** is a master degree student at the faculty of Computer Engineering of Uludağ University. He is currently working as Sofware Development Specialist at CITS R&D Center. He completed his bachelor degree at Kocaeli University, Faculty of Computer Engineering (2016). He worked on human-computer interaction and EEG signals during his bachelor education. Simultaneously, he continued to work on the creation of various mobile and web-based applications. Since he started his business life in the machinery sector, he was able to examine the maintenance processes in the machines and conducted research on machine learning methods for early diagnosis of machine maintenance.

**Dr. Özlem Albayrak** is currently working as Research and Development Manager at TEKNOPAR. She pursued her PhD (2002) from Ankara University, and her MBA in Management Information Systems (1994) and BSc. in Computer Engineering and Information Sciences department of Bilkent University (1992). She conducts eclectic and empirical software engineering studies related to requirements management, software engineering education, gamification and Industry 4.0. Her current research is focused on cognitive digital twins, application of artificial intelligence for various topics, especially anomaly detection and predictive maintenance. Dr. Albayrak has worked both in academia and in the industry. She worked for different international software project development in military, health, disaster management, and Industry 4.0 domains.

**Dr. Perin Ünal** is a graduate of Bilkent University, Industrial Engineering Department. She holds a MSc degree in Economics (specialization in Econometrics) and a Ph.D. degree in Information Systems from METU Informatics Institute and continues post-doc studies at TOBB University Computer Engineering department. She is the Senior Vice President of R&D at TEKNOPAR. where she has been responsible for the technical and project management of various information systems and digitilization projects. Her research interests include IoT, digital twins, big data analytics, machine learning, cognitive systems, and artificial intelligence technologies. She has authored academic publications, including papers in international refereed scientific journals, chapters in books, and papers in conference proceedings, and holds two patents.

**Dr. Pınar Kırcı** is currently an Associate Professor in the Department of Computer Engineering, Faculty of Engineering, Bursa Uludag University. She received her BSc degree from Department of Mathematics-Computer, Faculty of Science and Literature, Beykent University in 2002. Then, she received her MSc degree from Department of Computer Engineering in Istanbul University in 2005 about fiber optical networks. She received her Ph.D. degree from Department of Computer Engineering in Istanbul University in 2011 about multiplexing in fiber optical networks. Lately, she has worked on wireless sensor networks. She worked as an Associate Professor in Department of Engineering Sciences, Faculty of Engineering, Istanbul University-Cerrahpaşa between 2016-2019.