

## Research Article

# Deep Learning Enabled Fault Diagnosis Using Time-Frequency Image Analysis of Rolling Element Bearings

David Verstraete,<sup>1</sup> Andrés Ferrada,<sup>2</sup> Enrique López Droguett,<sup>1,3</sup>  
Viviana Meruane,<sup>3</sup> and Mohammad Modarres<sup>1</sup>

<sup>1</sup>Department of Mechanical Engineering, University of Maryland, College Park, MD, USA

<sup>2</sup>Computer Science Department, University of Chile, Santiago, Chile

<sup>3</sup>Mechanical Engineering Department, University of Chile, Santiago, Chile

Correspondence should be addressed to David Verstraete; [dbverstr@terpmail.umd.edu](mailto:dbverstr@terpmail.umd.edu)

Received 13 May 2017; Accepted 14 August 2017; Published 9 October 2017

Academic Editor: Matthew J. Whelan

Copyright © 2017 David Verstraete et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Traditional feature extraction and selection is a labor-intensive process requiring expert knowledge of the relevant features pertinent to the system. This knowledge is sometimes a luxury and could introduce added uncertainty and bias to the results. To address this problem a deep learning enabled featureless methodology is proposed to automatically learn the features of the data. Time-frequency representations of the raw data are used to generate image representations of the raw signal, which are then fed into a deep convolutional neural network (CNN) architecture for classification and fault diagnosis. This methodology was applied to two public data sets of rolling element bearing vibration signals. Three time-frequency analysis methods (short-time Fourier transform, wavelet transform, and Hilbert-Huang transform) were explored for their representation effectiveness. The proposed CNN architecture achieves better results with less learnable parameters than similar architectures used for fault detection, including cases with experimental noise.

## 1. Introduction

With the proliferation of inexpensive sensing technology and the advances in prognostics and health management (PHM) research, customers are no longer requiring that their new asset investment be highly reliable; instead they are requiring that their assets possess the capability to diagnose faults and provide alerts when components need to be replaced. These assets often have substantial sensor systems capable of generating millions of data points a minute. Handling this amount of data often involves careful construction and extraction of features from the data to input into a predictive model. Feature extraction relies on some prior knowledge of the data. Choosing which features to include or exclude within the model is a continuous area of research without a set methodology to follow.

Feature extraction and selection has opened a host of opportunities for fault diagnosis. The transformation of a raw

signal into a feature vector allows the learning method to separate classes and identify previously unknown patterns within the data. This has had wide ranging economic benefits for the owners of the assets and has opened new possibilities of revenue by allowing original equipment manufacturers (OEMs) to contract in maintainability and availability value. However, the state of current diagnostics involves a laborious process of creating a feature vector from the raw signal via feature extraction [1–3]. For example, Seera proposes a Fuzzy-Min-Max Classification and Regression Tree (FMM-CART) model for diagnostics on Case Western's bearing data [4]. Traditional feature extraction was completed within both time and frequency domains. An important predictor-based feature selection measure was used to enhance the CART model. multilayer perceptron (MLP) was then applied to the features for prediction accuracies.

Once features are extracted, traditional learning methods are then applied to separate, classify, and predict faults from

learned patterns present within the layers of the feature vector [5, 6]. These layers of features are constructed by human engineers; therefore, they are subject to uncertainty and biases of the domain experts creating these vectors. It is becoming more common that this process is performed on a set of massive multidimensional data. Having prior knowledge of the features and representations within such a data set, relevant to the patterns of interest, is a challenge and is often only one layer deep.

It is in this context that deep learning comes to play. Indeed, deep learning encompasses a set of representation learning methods with multiple layers. The primary benefit is the ability of the deep learning method to learn nonlinear representations of the raw signal to a higher level of abstraction and complexity isolated from the touch of human engineers directing the learning [7]. For example, to handle the complexity of image classification, convolutional neural networks (ConvNets or CNNs) are the dominant method [8–13]. In fact, they are so dominant today that they rival human accuracies for the same tasks [14, 15].

This is important from an engineering context because covariates often do not have a linear effect on the outcome of the fault diagnosis. Additionally, there are situations where a covariate is not directly measured confounding what could be a direct effect on the asset. The ability of deep learning based methods to automatically construct nonlinear representations given these situations is of great value to the engineering and fault diagnosis communities.

Since 2015, deep learning methodologies have been applied, with success, to diagnostics or classification tasks of rolling element signals [2, 16–26]. Wang et al. [2] proposed the use of wavelet scalogram images as an input into a CNN to detect faults within a set of vibration data. A series of  $32 \times 32$  images is used. Lee et al. [20] explored a corrupted raw signal and the effects of noise on the training of a CNN. While not explicitly stated, it appears that minimal data conditioning by means of a short-time Fourier transform was completed and either images or a vector of these outputs, independent of time, was used as the input layer to the CNN. Guo et al. [17] used Case Western's bearing data set [4] and an adaptive deep CNN to accomplish fault diagnosis and severity. Abdeljaber et al. [19] used a CNN for structural damage detection on a grandstand simulator. Janssens et al. [21] incorporated shallow CNNs with the amplitudes of the discrete Fourier transform vector of the raw signal as an input. Pooling, or subsampling, layers were not used. Chen et al. [16] used traditional feature construction as a vector input to a CNN architecture consisting of one convolutional layer and one pooling layer for gearbox vibration data. Although not dealing with rolling elements, Zhang [22] used a deep learning multiobjective deep belief network ensemble method to estimate the remaining useful life of NASA's C-MAPSS data set. Liao et al. [23] used restricted Boltzmann machines (RBMs) as a feature extraction method, otherwise known as transfer learning. Feature selection was completed from the RBM output, followed by a health assessment via self-organizing maps (SOMs). the remaining useful life (RUL) was then estimated on run-to-failure data sets. Babu [24] used images of two PHM competition data sets (C-MAPSS and

PHM 2008) as an input to a CNN architecture. While these data sets did not involve rolling elements, the feature maps were time-based, therefore allowing the piecewise remaining useful life estimation. Guo et al. [18] incorporated traditional feature construction and extraction techniques to feed a stacked autoencoder (SAE) deep neural network. SAEs do not utilize convolutional and pooling layers. Zhou et al. [25] used fast Fourier transform on the Case Western bearing data set for a vector input into a deep neural network (DNN) using 3, 4, and 5 hidden layers. DNNs do not incorporate convolutional and pooling layers, only hidden layers. Liu et al. [26] used spectrograms as input vectors into sparse and stacked autoencoders with two hidden layers. Liu's results indicate there was difficulty classifying outer race faults versus the baseline. Previous deep learning based models and applications to fault diagnostics are usually limited by their sensitivity to experimental noise or their reliance on traditional feature extraction.

In this paper, we propose an improved CNN based model architecture for time-frequency image analysis for fault diagnosis of rolling element bearings. Its main element consists of a double layer CNN, that is, two consecutive convolutional layers without a pooling layer between them. Furthermore, two linear time-frequency transformations are used as image input to the CNN architecture: short-time Fourier transform spectrogram and wavelet transform (WT) scalogram. One nonlinear nonparametric time-frequency transformation is also examined: Hilbert-Huang transformation (HHT). HHT is chosen to compliment the traditional time-frequency analysis of STFT and WT due to its benefit of not requiring the construction of a basis to match the raw signal components. These three methods were chosen because they give suitable outputs for the discovery of complex and high-dimensional representations without the need for additional feature extraction. Additionally, HHT images have not been used as a basis for fault diagnostics.

Beyond the CNN architecture and three time-frequency analysis methods, this paper also examines the loss of information due to the scaling of images from  $96 \times 96$  to  $32 \times 32$  pixels. Image size has significant impact on the CNN's quantity of learnable parameters. Training time is less if the image size can be reduced, but classification accuracy is negatively impacted. The methodology is applied to two public data sets: (1) the Machinery Failure Prevention Technology (MFPT) society rolling element vibrational data set and (2) Case Western Reserve University's Bearing data set [4].

The rest of this paper is organized as follows: Section 2 provides an overview of deep learning and CNNs. Section 3 gives a brief overview of the time-frequency domain analysis incorporated into the image structures for the deep learning algorithm to train. Section 4 outlines the proposed CNN architecture constructed to accomplish the diagnostic task of fault detection. Sections 5 and 6 apply the methodology to two experimental data sets. Comparisons of the proposed CNN architecture against MLP, linear support vector machine (SVM), and Gaussian SVM for both the raw data and principal component mapping data are presented. Additionally, comparisons with Wang et al. [2] proposed

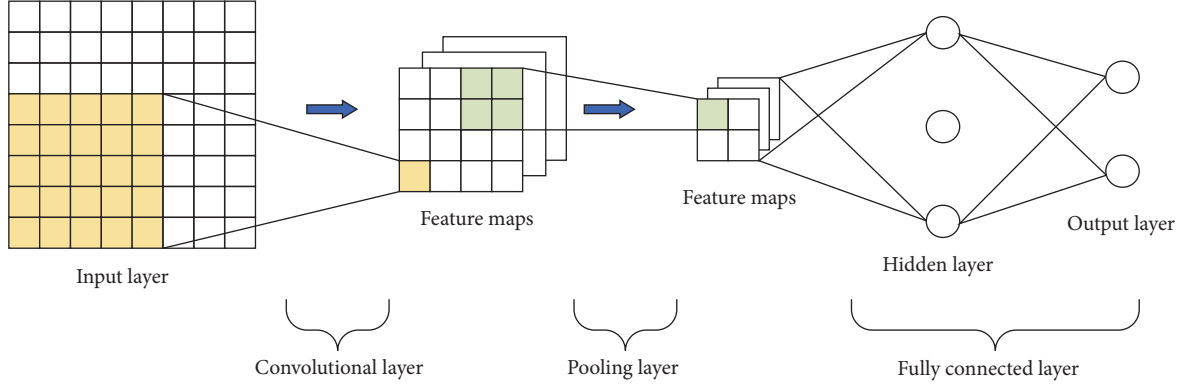


FIGURE 1: Generic CNN architecture.

CNN architecture is presented. Section 7 examines the data set with traditional feature learning. Section 8 explores the addition of Gaussian noise to the signals. Section 9 concludes with discussion of the results.

## 2. Deep Learning and CNN Background

Deep learning is representation learning; however, not all representation learning is deep learning. The most common form of deep learning is supervised learning. That is, the data is labeled prior to input into the algorithm. Classification or regression can be run against these labels, and thus predictions can be made from unlabeled inputs.

Within the computer vision community, there is one clear favorite type of deep, feedforward network that outperformed others in generalizing and training networks consisting of full connectivity across adjacent layers: the convolutional neural network (CNN). A CNN's architecture is constructed as a series of stages. Each stage has a different role. Each role is completed automatically within the algorithm. Each architecture within the CNN construct consists of four properties: multiple layers, pooling/subsampling, shared weights, and local connections.

As shown in Figure 1, the first stage of a CNN is made of two types of layers: convolutional layers which organize the units in feature maps and pooling layers which merge similar features into one feature. Within the convolutional layer's feature map, each unit is connected to a previous layer's feature maps through a filter bank. This filter consists of a set of weights and a corresponding local weighted sum. The weighted sum passed through a nonlinear function such as a rectified linear unit (ReLU). This is shown in (1). ReLU is a half-wave rectifier,  $f(x) = \max(x, 0)$ , and is like the Softplus activation function; that is,  $\text{Softplus}(x) = \ln(1 + e^x)$ . ReLU activation trains faster than the previously used sigmoid/tanh functions [7]:

$$\mathbf{X}_k^{(m)} = \text{ReLU} \left( \sum_{c=1}^C \mathbf{W}_k^{(c,m)} * \mathbf{X}_{k-1}^{(c)} + \mathbf{B}_k^{(m)} \right), \quad (1)$$

where  $*$  represents the convolutional operator;  $\mathbf{X}_{k-1}^{(c)}$  is input of convolutional channel  $c$ ;  $\mathbf{W}_k^{(c,m)}$  is filter weight matrix;  $\mathbf{B}_k^{(m)}$  is bias weight matrix; ReLU is rectified linear unit.

An important aspect of the convolutional layers for image analysis is that units within the same feature map share the same filter bank. However, to handle the possibility that a feature map's location is not the same for every image, different feature maps use different filter banks [7]. For image representations of vibration data this is important. As features are extracted to characterize a given type of fault represented on the image, it may be in different locations on subsequent images. It is worth noting that feature construction happens automatically within the convolutional layer, independent of the engineer constructing or selecting them, which gives rise to the term *featureless learning*. To be consistent with the terminology of the fault diagnosis community, one could liken the convolutional layer to a feature construction, or extraction, layer. If a convolutional layer is similar in respect to feature construction, the pooling layer in a CNN could be related to a feature selection layer.

The second stage of a CNN consists of a pooling layer to merge similar features into one. This pooling, or subsampling, effectively reduces the dimensions of the representation. Mathematically, the subsampling function  $f$  is [27]

$$\mathbf{X}_k^{(m)} = f \left( \beta_k^{(m)} \text{down}(\mathbf{X}_k^{(m-1)}) + b_k^{(m)} \right), \quad (2)$$

where  $\text{down}(\bullet)$  represents the subsampling function.  $\beta_k^{(m)}$  is multiplicative bias.  $b_k^{(m)}$  is additive bias.

After multiple stacks of these layers are completed, the output can be fed into the final stage of the CNN, a multilayer perceptron (MLP) fully connected layer. An MLP is a classification feedforward neural network. The outputs of the final pooling layer are used as an input to map to labels provided for the data. Therefore, the analysis and prediction of vibration images are a series of representations of the raw signal. For example, the raw signal can be represented in a sinusoidal form via STFT. STFT is then represented graphically via a spectrogram, and finally a CNN learns and classifies the spectrogram image features and representations that best predict a classification based on a label. Figure 2 outlines how deep learning enabled feature learning differs from traditional feature learning.

Traditional feature learning involves a process of constructing features from the existing signal, feature searching

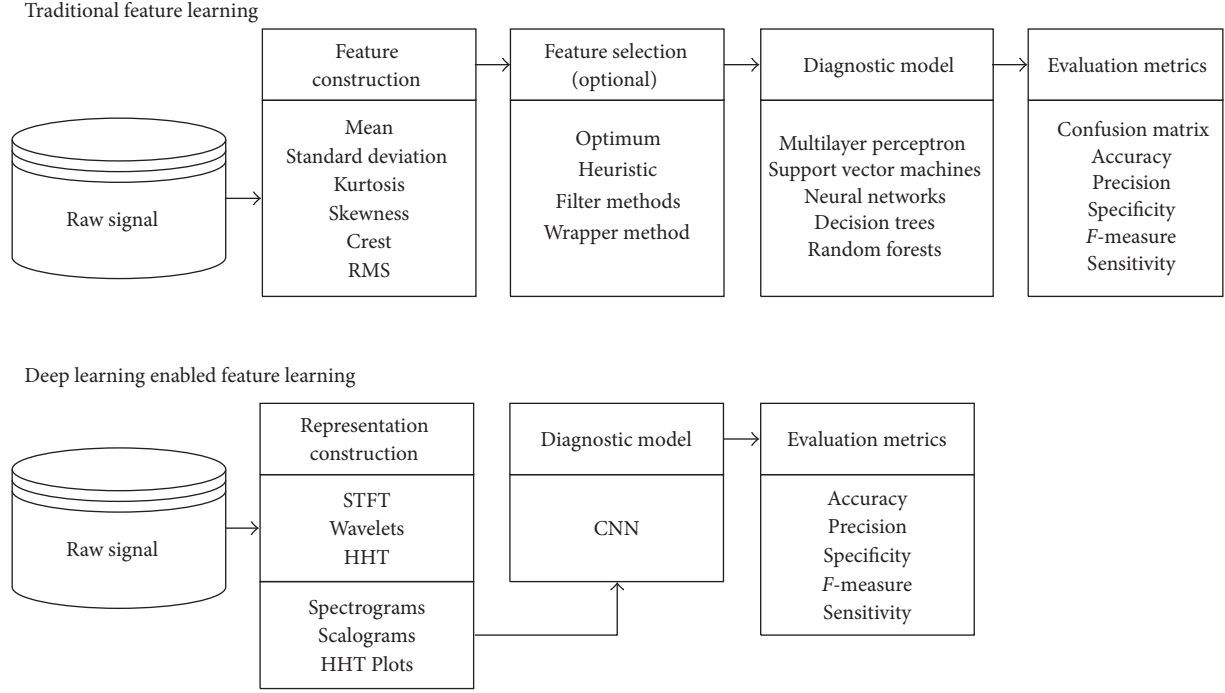


FIGURE 2: Process of representations for time-frequency analysis.

via optimum or heuristic methods, feature selection of relevant and important features via filter or wrapper methods, and feeding the resulting selected features into a classification algorithm. Deep learning enabled feature learning has the advantage of not requiring a feature construction, search, and selection sequence. This is done automatically within the framework of the CNN. The strength of a CNN in its image analysis capabilities. Therefore, an image representation of the data as an input into the framework is ideal. A vector input of constructed features misses the intent and power of the CNN. Given that the CNN searches spatially for features, the sequence of the vector input can affect the results. Within this paper spectrograms, scalograms, and HHT plots are used as the image input to leverage the strengths of a CNN as shown in Figure 2.

### 3. Time-Frequency Methods Definition and Discussion

Time frequency represents a signal in both the time and frequency domains simultaneously. The most common time-frequency representations are spectrograms and scalograms. A spectrogram is a visual representation in the time-frequency domain of a signal using the STFT, and a scalogram uses the WT. The main difference with both techniques is that spectrograms have a fixed frequency resolution that depends on the windows size, whereas scalograms have a frequency-dependent frequency resolution. For low frequencies, a long window is used to observe enough of the slow alternations in the signal and at higher frequency values a shorter window is

used which results in a higher time resolution and a poorer frequency resolution. On the other hand, the HHT does not divide the signal at fixed frequency components, but the frequency of the different components (IMFs) adapts to the signal. Therefore, there is no reduction of the frequency resolution by dividing the data into sections, which gives HHT a higher time-frequency resolution than spectrograms and scalograms. In this paper, we examine the representation effectiveness of the following three methods: STFT, WT, and HHT. These representations will be graphically represented as an image and fed into the proposed CNN architecture in Section 4.

**3.1. Spectrograms: Short-Time Fourier Transform (STFT).** Spectrograms are a visual representation of the STFT where the  $x$ - and  $y$ -axis are time and frequency, respectively, and the color scale of the image indicates the amplitude of the frequency. The basis for the STFT representation is a series of sinusoids. STFT is the most straightforward frequency domain analysis. However, it cannot adequately model time-variant and transient signal. Spectrograms add time to the analysis of FFT allowing the localization of both time and frequency. Figure 3 illustrates a spectrogram for the baseline condition of a rolling element bearing vibrational response.

**3.2. Scalograms: Wavelet Transform.** Scalograms are a graphical image of the wavelet transform (WT). WTs are a linear time-frequency representation with a wavelet basis instead of sinusoidal functions. Due to the addition of a scale variable along with the time variable, the WT is effective for nonstationary and transient signals.



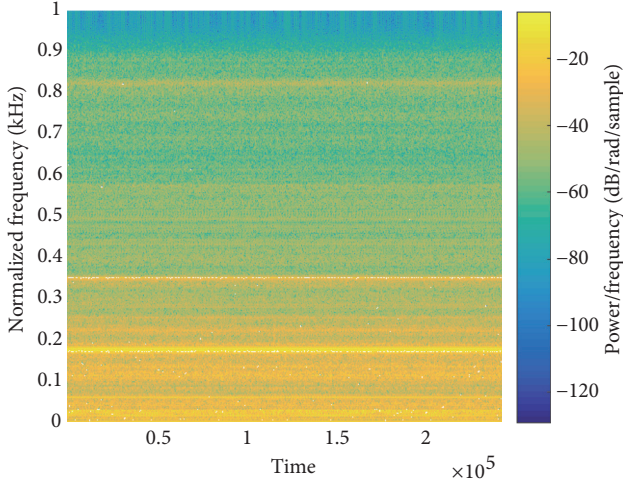


FIGURE 3: STFT spectrogram of baseline raw signal.

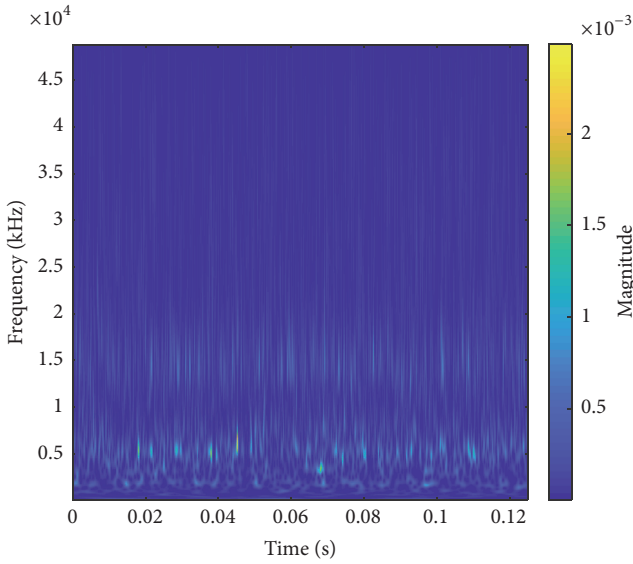


FIGURE 4: Wavelet transform scalogram of baseline raw signal.

For a wavelet transform,  $WT_x(b, a)$ , of a signal which is energy limited  $x(t) \in L^2(R)$ ; the basis for the transform can be set as

$$WT_x(b, a) = \frac{1}{\sqrt{a}} \int_{-\infty}^{+\infty} x(t) \psi\left(\frac{t-b}{a}\right) dt, \quad (3)$$

where  $a$  is scale parameter;  $b$  is time parameter;  $\psi$  is analyzing wavelet.

Figure 4 illustrates a scalogram with a Morlet wavelet basis for the baseline condition of a rolling element bearing vibrational response. There have been many studies on the effectiveness of individual wavelets and their ability to match a signal. One could choose between the Gaussian, Morlet, Shannon, Meyer, Laplace, Hermit, or the Mexican Hat wavelets in both simple and complex functions. To date there is not a defined methodology for identifying the proper wavelet to be used and this remains an open question within

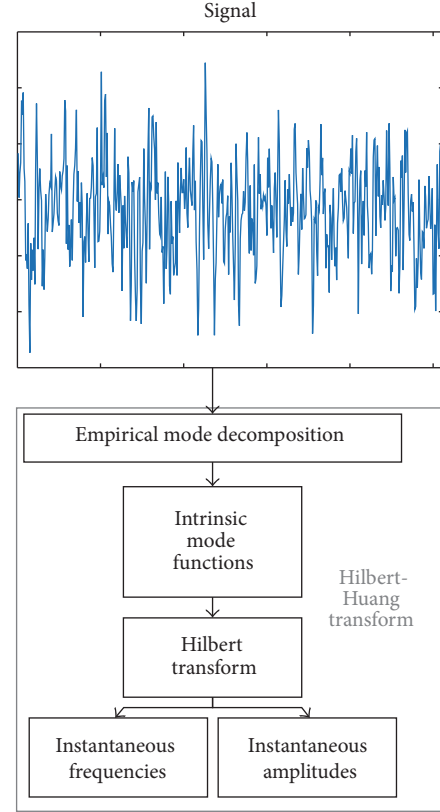


FIGURE 5: Overview of HHT adapted from Wang (2010).

the research community [28]. For the purposes of this paper, the Morlet wavelet,  $\Psi_\sigma(t)$ , is chosen because of its similarity to the impulse component of symptomatic faults of many mechanical systems [29] and is defined as

$$\Psi_\sigma(t) = c_\sigma \pi^{-(1/4)} e^{-(1/2)t^2} (e^{i\sigma t} - K_\sigma), \quad (4)$$

where  $c$  is normalization constant.  $K_\sigma$  is admissibility criterion.

Wavelets have been extensively used for machinery fault diagnosis. For the sake of brevity, those interested can refer to Peng and Chu [30] for a comprehensive review of the wavelet transform's use within condition monitoring and fault diagnosis.

**3.3. Hilbert-Huang Transform.** Feng et al. [28] refer to the time-frequency analysis method, Hilbert-Huang transform (HHT), as an adaptive nonparametric approach. STFT and WT are limited in the sense that they are a representation of the raw signal on a predefined set of basis function. HHT does not make predefined assumptions on the basis of the data but employs the empirical mode decomposition (EMD) to decompose the signal into a set of elemental signals called intrinsic mode functions (IMFs). The HHT methodology is depicted in Figure 5.

The HHT is useful for nonlinear and nonstationary time series analysis which involves two steps: EMD of the time series signal and Hilbert spectrum construction. It is

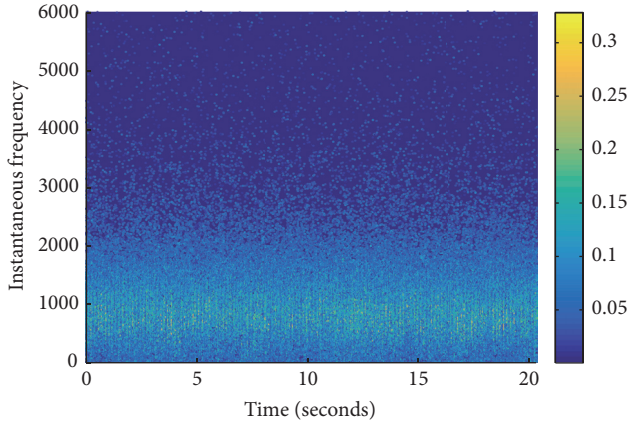


FIGURE 6: HHT image of baseline raw signal.

an iterative numerical algorithm which approximates and extracts IMFs from the signal. HHTs are particularly useful for localizing the properties of arbitrary signals. For details of the complete HHT algorithm, the reader is directed towards Huang [31].

Figure 6 shows an HHT image of the raw baseline signal used in Figures 3 and 4. It is not uncommon for the HHT instantaneous frequencies to return negative values. This is because the HHT derives the instantaneous frequencies from the local derivatives of the IMF phases. The phase is not restricted to monotonically increasing and can therefore decrease for a time. This results in a negative local derivative. For further information regarding this property of HHT, the reader is directed to read Meeson [32].

The EMD portion of the HHT algorithm suffers from possible mode mixing. Intermittences in signal can cause this. Mode mixing within signals containing instantaneous frequency trajectory crossings is inevitable. The results of mode mixing can result in erratic or negative instantaneous frequencies [33]. This means for such signals HHT does not outperform traditional time-frequency analysis methods such as STFT.

#### 4. Proposed CNN Architecture for Fault Classification Based on Vibration Signals

The primary element of the proposed architecture consists of a double layer CNN, that is, two consecutive convolutional layers without a pooling layer between them. The absence of a pooling layer reduces the learnable parameters and increases the expressivity of the features via an additional nonlinearity. However, a pooling layer is inserted between two stacked double convolutional layers. This part of the architecture makes up the automatic feature extraction process that is then followed by a fully connected layer to accomplish rolling element fault detection.

The first convolutional layer consists of 32 feature maps of  $3 \times 3$  size, followed by second convolutional layer of 32 feature maps of  $3 \times 3$  size. After this double convolutional layer, there is a pooling layer of 32 feature maps of  $2 \times 2$  size. This makes up the first stage. The second stage consists

of two convolutional layers of 64 feature maps each, of  $3 \times 3$  size, followed by subsampling layer of 64 feature maps of  $2 \times 2$  size. The third stage consists of two convolutional layers of 128 feature maps each, of  $3 \times 3$  size, followed by subsampling layer of 128 feature maps of  $2 \times 2$  size. The last two layers are fully connected layers of 100 features. Figure 7 depicts this architecture. The intent of two stacked convolutional layers before a pooling layer is to get the benefit of a large feature space via smaller features. This convolutional layer stacking has two advantages: (1) reducing the number of parameters the training stage must learn and (2) increasing the expressivity of the feature by adding an additional nonlinearity.

Table 1 provides an overview of CNN architectures that have been used for fault diagnosis, where C's are convolutional layers, P's are pooling layers, and FCs are fully connected layers. The number preceding the C, P, and FC indicates the number of feature maps used. The dimensions  $[3 \times 3]$  and  $[2 \times 2]$  indicate the pixel size of the features.

Training the CNN involves the learning of all of the weights and biases present within the architectures. These weights and biases are referred to as learnable parameters. The quantity of learnable parameters for a CNN architecture can radically improve or degrade the time to train of the model. Therefore, it is important to optimize the learnable parameters by balancing training time versus prediction accuracy. Table 2 outlines the quantity of learnable parameters for the proposed CNN architecture as well as a comparison to architectures 1 and 2 presented in Table 1.

Beyond the learnable parameters, the CNN requires the specification and optimization of the hyperparameters: dropout and learning rate. Dropout is an essential property of CNNs. Dropout helps to prevent overfitting and reduce training error and effectively thins the network (Srivastava, 2015). The remaining connections are comprised of all the units that survive the dropout. For this architecture, dropout is set to 0.5. For the other hyperparameter, learning rate, the adapted moment estimation (ADAM) algorithm was used for optimization. It has had success in the optimizing the learning rate for CNNs faster than similar algorithms. Instead of hand picking learning rates like similar algorithms, the ADAM learning rate scale adapts through different layers [34].

Part of the reason for deep learning's recent success has been the use of graphics processing unit (GPU) computing [7]. GPU computing was used for this paper to increase the speed and decrease the training time. More specifically, the processing system used for the analysis is as follows: CPU Core i7-6700 K 4.2 GHz with 32 GB ram and GPU Tesla K20.

#### 5. Case Study 1: Machinery Failure Prevention Technology

This data set was provided by the Machinery Failure Prevention Technology (MFPT) Society [35]. A test rig with a NICE bearing gathered acceleration data for baseline conditions at 270 lbs of load and a sampling rate of 97,656 Hz for six seconds. In total, ten outer-raceway and seven inner-raceway fault conditions were tracked. Three outer race faults included 270 lbs of load and a sampling rate of 97,656 Hz for six

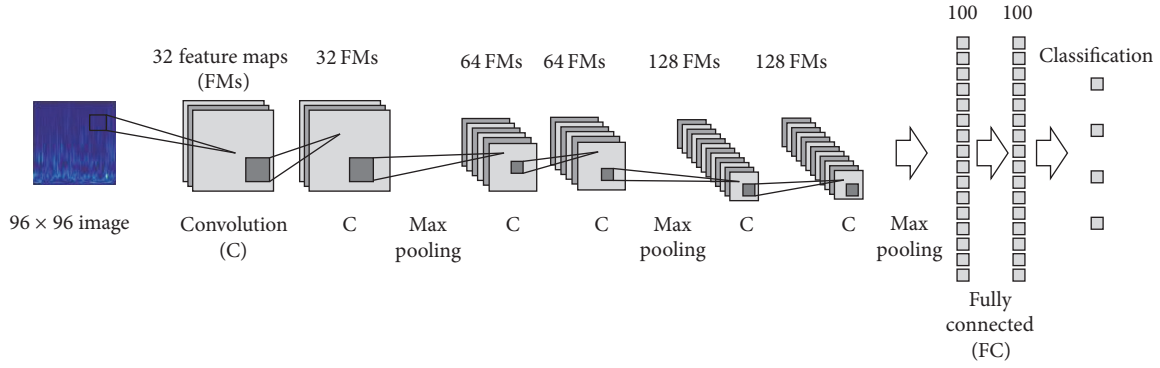


FIGURE 7: Proposed CNN architecture.

TABLE 1: Overview of CNN architectures used for fault diagnosis.

Proposed model	CNN architecture
Architecture 1 [2]	Input[32 × 32]–64C[3 × 3]–64P[2 × 2]–64C[4 × 4]–64P[2 × 2]–128C[3 × 3]–128P[2 × 2]–FC[512]
Architecture 2, Chen et al. [16]	Input[32 × 32]–16C[3 × 3]–16P[2 × 2]–FC[10]
Proposed architecture	Input[32 × 32]–32C[3 × 3]–32C[3 × 3]–32P[2 × 2]–64C[3 × 3]–64C[3 × 3]–64P[2 × 2]–128C[3 × 3]–128C[3 × 3]–128P[2 × 2]–FC[100]–FC[100]
Proposed architecture	Input[96 × 96]–32C[3 × 3]–32C[3 × 3]–32P[2 × 2]–64C[3 × 3]–64C[3 × 3]–64P[2 × 2]–128C[3 × 3]–128C[3 × 3]–128P[2 × 2]–FC[100]–FC[100]
Guo et al. [17, 18]	Input[32 × 32]–5C[5 × 5]–5P[2 × 2]–10C[5 × 5]–10P[2 × 2]–10C[2 × 2]–10P[2 × 2]–FC[100]–FC[50]
Abdeljaber et al. [19]	Input[128]–64C[41]–64P[2]–32C[41]–32P[2]–FC[10–10]

TABLE 2: Overview of learnable parameters for the CNN architectures.

CNN model	32 × 32 image	96 × 96 image
Architecture 2	41,163	368,854
Proposed CNN	501,836	2,140,236
Architecture 1	1,190,723	9,579,331

seconds. Seven additional outer race faults were assessed at varying loads: 25, 50, 100, 150, 200, 250, and 300 lbs. The sample rate for the faults was 48,828 Hz for three seconds. Seven inner race faults were analyzed with varying loads of 0, 50, 100, 150, 200, 250, and 300 lbs. The sample rate for the inner race faults was 48,848 Hz for three seconds. Spectrogram, scalogram, and HHT images were generated from this data set with the following classes: normal baseline (N), inner race fault (IR), and outer race fault (OR). The raw data consisted of the following data points: N with 1,757,808 data points, IR with 1,025,388 data points, and OR with 2,782,196 data points. The total images produced from the data set are as follows: N with 3,423, IR with 1,981, and OR with 5,404.

From MFPT, there was more data and information on the outer race fault conditions; therefore more images were generated. This was decided due to the similarities between the baseline images and the outer race fault images as shown in Tables 5 and 7. It is important to note that functionally the CNN looks at each pixel's intensity value to learn the features.

Therefore, based on size and quantity, the 96 × 96-pixel and 32 × 32-pixel images result in 99,606,528 and 11,067,392 data points, respectively.

Once the data images were generated, bilinear interpolation [36] was used to scale the image down to the appropriate size for training the CNN model. From this image data a 70/30 split was used for the training and test sets. These images are outlined in Tables 3, 4, and 5.

Within the MFPT image data set, a few things stand out. Although the scalogram images of the outer race faults versus the baseline are similar, the scalogram images had the highest prediction accuracy from all the modeling techniques employed in Tables 6 and 7. The information loss of the HHT images when reducing the resolution from 96 × 96 to 32 × 32 pixels could be relevant because of the graphical technique used to generate the images.

Depending upon the modeling technique used, the prediction accuracies are higher or lower in Tables 6 and 7. The CNN modeling had a significant shift between 96 and 32 image resolutions. Support vector machines (SVM) had a

TABLE 3: MFPT baseline images.





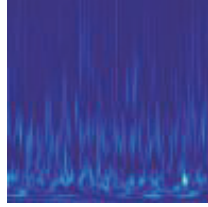

Image size (pixels)	Spectrogram	Scalogram	HHT
$32 \times 32$			
$96 \times 96$			

TABLE 4: MFPT inner race images.




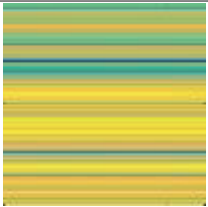
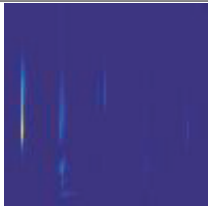

Image size (pixels)	Spectrogram	Scalogram	HHT
$32 \times 32$			
$96 \times 96$			

TABLE 5: MFPT outer race images.




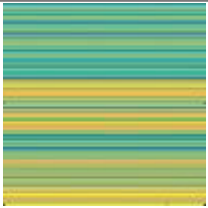
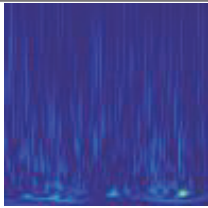

Image size (pixels)	Spectrogram	Scalogram	HHT
$32 \times 32$			
$96 \times 96$			

TABLE 6: Prediction accuracies for  $32 \times 32$ -pixel image inputs.

Model	Spectrogram	Scalogram	HHT
MLP flat	70.3%	94.0%	49.2%
LSVM flat	63.6%	91.8%	50.0%
SVM flat	73.9%	92.7%	58.5%
MLP PCA	62.3%	95.3%	56.7%
LSVM PCA	48.8%	89.9%	45.8%
SVM PCA	51.3%	92.5%	56.4%
Architecture 2	77.3%	92.4%	68.9%
Architecture 1	80.6%	99.8%	74.5%
Proposed CNN architecture	81.4%	99.7%	75.7%



TABLE 7: Prediction accuracies for  $96 \times 96$ -pixel image inputs.

Model	Spectrogram	Scalogram	HHT
MLP flat	80.1%	81.3%	56.8%
LSVM flat	77.1%	91.9%	52.8%
SVM flat	85.1%	93.3%	57.8%
MLP PCA	81.5%	96.4%	69.2%
LSVM PCA	74.1%	92.0%	51.4%
SVM PCA	49.6%	70.0%	68.8%
Architecture 2	81.5%	97.0%	74.2%
Architecture 1	86.2%	99.9%	91.8%
Proposed CNN architecture	91.7%	99.9%	95.5%

TABLE 8: MFPT paired two-tailed  $t$ -test  $p$  values.

Image type	Architecture 1	Architecture 1	Architecture 2	Architecture 2
	$32 \times 32$	$96 \times 96$	$32 \times 32$	$96 \times 96$
Scalogram	0.080	0.344	0.049	0.108
Spectrogram	0.011	0.037	0.058	0.001
HHT	0.031	0.410	0.000	0.000

difficult time predicting the faults for both the raw data (flat pixel intensities) and principal component analysis (PCA).

Flat pixel data versus PCA of the pixel intensities varied across different modeling and image selection. Scalograms outperformed spectrograms and HHT. However, the optimal modeling method using traditional techniques varied. For both the HHT and spectrogram images, SVM on the flat data was optimal. For scalograms, MLP on the PCA data was optimal.

Resolution loss from the reduction in image from  $96 \times 96$  to  $32 \times 32$  influenced the fault diagnosis accuracies. There was a slight drop in the scalogram accuracies between the two images sizes except for SVM PCA modeling. Spectrograms suffered a little from the resolution drop; however, HHT was most affected. This is due to the image creation method. Scatter plots were used due to the point estimates of the instantaneous frequencies and amplitudes.

With regard to the CNN architectures, the proposed deep architecture outperformed the shallow one. The shallow CNN architecture outperformed the traditional classification methodologies in the  $96 \times 96$  image sizes except for spectrograms. With a  $32 \times 32$  image size, the shallow CNN outperformed the traditional methods except for the scalogram images. The proposed CNN architecture performed better overall for the four different image techniques and resolution sizes except for  $32 \times 32$  scalograms.

To measure the similarity between the results of the proposed CNN architecture versus architectures 1 and 2, the model accuracies were compared with a paired two tail  $t$ -test. Table 8 outlines the  $p$  values with a null hypothesis of zero difference between the accuracies. A  $p$  value above 0.05 means the results are statistically the same. A  $p$  value less than 0.05 indicates the models are statistically distinct.

From the results in Table 8, one can see that the proposed architecture has the advantage of outperforming or achieving statistically identical accuracies with less than half

TABLE 9: Confusion matrices for MFPT (a)  $96 \times 96$  and (b)  $32 \times 32$  scalograms for the proposed architecture.

(a)			
	N	IR	OR
N	<b>99.9%</b>	0.0%	0.1%
IR	0.0%	<b>100%</b>	0.0%
OR	0.1%	0.0%	<b>99.9%</b>
(b)			
	N	IR	OR
N	<b>99.6%</b>	0.1%	0.3%
IR	0.0%	<b>100%</b>	0.0%
OR	0.5%	0.0%	<b>99.5%</b>

the amount of the learnable parameters. Table 9 outlines the confusion matrices results for the MFPT data set on  $96 \times 96$  and  $32 \times 32$  scalograms. The values are horizontally normalized by class. From this, the following 4 metrics were derived: precision, sensitivity, specificity, and  $F$ -measure (see [37] for details on these metrics).

From the results shown in Tables 10–13, the precision, sensitivity, specificity, and  $f$ -measures of the proposed architecture outperform the other two CNN architectures when dealing with spectrograms and HHT images of both  $96 \times 96$  and  $32 \times 32$  sizes and are statistically identical to architecture 1 in case of scalograms. Precision assessment is beneficial for diagnostics systems as it emphasizes false positives, thus evaluating the model's ability to predict actual faults. To measure the precision for the model, one must look at each class used in the model. For the MFPT data set, three classes were used. Table 10 outlines the average precision of the three classes for the three architectures. Sensitivity is another effective measure for a diagnostic system's ability to

TABLE 10: Precision for MFPT data set.

Model	Proposed CNN architecture	Architecture 1	Architecture 2
Scalogram $32 \times 32$	99.7%	99.8%	91.9%
Scalogram $96 \times 96$	99.9%	99.9%	95.8%
Spectrogram $32 \times 32$	82.0%	81.4%	78.8%
Spectrogram $96 \times 96$	91.3%	85.0%	81.7%
HHT $32 \times 32$	75.9%	74.6%	71.0%
HHT $96 \times 96$	92.9%	89.7%	74.1%

TABLE 11: Sensitivity for MFPT data set.

Model	Proposed CNN architecture	Architecture 1	Architecture 2
Scalogram $32 \times 32$	99.7%	99.8%	89.6%
Scalogram $96 \times 96$	99.9%	100.0%	96.5%
Spectrogram $32 \times 32$	79.7%	77.8%	73.6%
Spectrogram $96 \times 96$	90.8%	82.1%	74.8%
HHT $32 \times 32$	76.2%	74.4%	68.0%
HHT $96 \times 96$	95.3%	92.3%	67.7%

TABLE 12: Specificity for MFPT data set.

Model	Proposed CNN architecture	Architecture 1	Architecture 2
Scalogram $32 \times 32$	99.8%	99.9%	94.9%
Scalogram $96 \times 96$	95.7%	89.6%	85.3%
Spectrogram $32 \times 32$	89.8%	89.0%	87.0%
Spectrogram $96 \times 96$	100.0%	100.0%	97.6%
HHT $32 \times 32$	89.3%	88.3%	85.1%
HHT $96 \times 96$	97.9%	96.6%	83.5%

TABLE 13:  $F$ -measure for MFPT data set.

Model	Proposed CNN architecture	Architecture 1	Architecture 2
Scalogram $32 \times 32$	99.8%	99.8%	90.2%
Scalogram $96 \times 96$	99.9%	99.9%	96.1%
Spectrogram $32 \times 32$	80.3%	78.5%	74.2%
Spectrogram $96 \times 96$	90.9%	81.5%	73.9%
HHT $32 \times 32$	74.0%	71.9%	65.4%
HHT $96 \times 96$	93.9%	90.1%	62.6%

classify actual faults. However, sensitivity emphasizes true negatives. Table 11 outlines the average sensitivity of the three classes. Specificity, or true negative rate, emphasizes false positives and is therefore effective for examining false alarm rates. Table 12 outlines the average specificity. The  $f$ -measure metric assesses the balance between precision and sensitivity. It does not take true negatives into account and illustrates a diagnostic system's ability to accurately predict true faults. Table 13 outlines the average  $f$ -measure for the three classes.

Overall, the proposed architecture outperforms or is statistically identical to the other CNN architectures for diagnostic classification tasks with far fewer learnable parameters. As shown from the images, the MFPT data set appears like it has more noise in the measurements from the baseline and outer race fault conditions. Under these conditions, the proposed architecture outperforms the other architectures

due to the two convolutional layers creating a more expressive nonlinear relationship from the images. Additionally, the proposed CNN can better classify outer race faults versus the baseline (normal) condition even with very similar images.

## 6. Case Study 2: Case Western Reserve University Bearing Data Center

The second experimental data set used in this paper was provided by Case Western Reserve (CWR) University Bearing Data Center [4]. A two-horsepower reliance electric motor was used in experiments for the acquisition of accelerometer data on both the drive end and fan end bearings, as shown in Figure 8. The bearings support the motor shaft. Single point artificial faults were seeded in the bearing's inner raceway (IR), outer raceway (OR), and rolling element (ball) (BF)

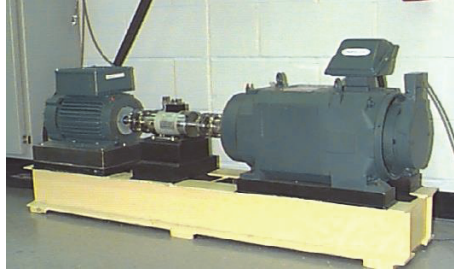


FIGURE 8: Test stand for roller bearing accelerometer data.

TABLE 14: CWR baseline images.

Image size	Spectrogram	Scalogram	HHT
$32 \times 32$			
$96 \times 96$			

with an electrodischarge machining (EDM) operation. These faults ranged in diameter and location of the outer raceway. The data includes a motor load of 0 to 3 horsepower. The accelerometers were magnetically attached to the housing at the 12 o'clock position.

For the purposes of this paper, the speed and load on the motor were not included as a classifier. Additionally, the fault sizes were grouped together as predicting the size of the fault was beyond the scope of this paper. A 70/30 split was used for the training and test data. Spectrogram, scalogram, and HHT images were generated from this data. The raw data consisted of the following data points: N had 1,691,648, BF had 1,441,792, IR had 1,440,768, and OR had 1,443,328 data points. The total images produced from the data set are as follows: N 3,304, BF 2,816, IR 2,814, and OR 2,819. From CWR, there was more balanced set of data between the baseline and faults. Again, based on size and quantity, the  $96 \times 96$  and  $32 \times 32$  images result in 108,315,648 and 12,035,072 data points, respectively. This data is used by the CNN to learn the features of the data.

Deep learning algorithms hold promise to unlock previously unforeseen relationship within explanatory variables; however, it is important to keep this in context. The value of these algorithms is as much as they can outperform much simpler fault diagnosis techniques. If envelope analysis, MLP, SVM, or other traditional approaches can achieve the same results, then there is no value in spending the extra time and resources to develop a deep learning algorithm to perform the analysis. Smith and Randall [38] outline this benchmark study for the Case Western Reserve data set for envelope analysis. Appendix B within that paper outlines the potential

areas within the data set where a more sophisticated analysis must be used to diagnose certain faults. From these results, analysis including the ball faults within the fault diagnosis requires more sophisticated techniques. These include data sets 118 to 121, 185 to 188, 222, 224, and 225. These data sets are used within this paper; therefore, there is potential value of the computational expense of the methodology proposed within this paper. These data sets incorporated the small injected faults at  $0.007''$  (data sets 118 to 121) to the larger injected faults of  $0.028''$  (data sets 3001 to 3004).

To be more explicit, the following data sets were used within the analysis: for the baseline, data sets 97 to 100; for the inner race, 105 to 108, 169 to 172, 209 to 212, and 3001 to 3004; for the ball faults, 118 to 121, 185 to 188, 222 to 225, and 3005 to 3008; for the outer race faults, 130 to 133, 197 to 200, 234 to 237, and 144 to 147.

Bilinear interpolation [36] was used to scale the image down to the appropriate size for training the CNN model. A 70/30 split was used for the training and test sets. These images are outlined in Tables 14, 15, 16, and 17.

The CWR image data set is different than the MFPT images. Even though the scalogram images of the ball faults versus the inner race faults are similar, all four image sets look easier to classify. The scalogram images had the highest prediction accuracy for modeling techniques employed in Tables 18 and 19. The information loss of the HHT images when reducing the resolution from  $96 \times 96$  to  $32 \times 32$  did not affect the predictions as much as the MFPT data had, possibly due to the lower noise levels in the case of the CWR data set.

Overall, spectrograms performed much better on the CWR data set than the MFPT data set. Flat pixel data versus

TABLE 15: CWR inner race images.





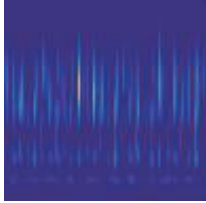
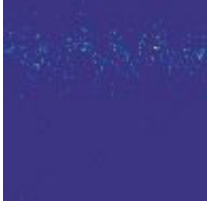
Image size	Spectrogram	Scalogram	HHT
$32 \times 32$			
$96 \times 96$			

TABLE 16: CWR ball fault images.




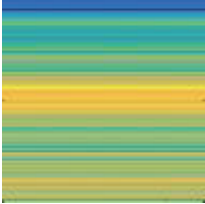
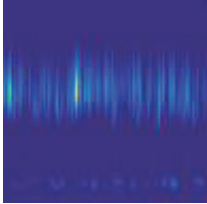
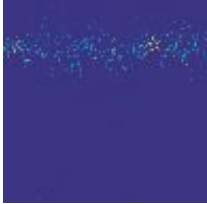
Image size	Spectrogram	Scalogram	HHT
$32 \times 32$			
$96 \times 96$			

TABLE 17: CWR outer race images.





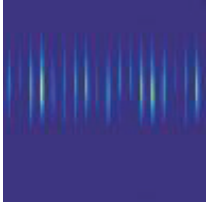

Image size	Spectrogram	Scalogram	HHT
$32 \times 32$			
$96 \times 96$			

TABLE 18: Prediction accuracies for  $32 \times 32$  image inputs.

Model	Spectrogram	Scalogram	HHT
MLP flat	92.7%	83.6%	59.6%
LSVM flat	88.6%	80.8%	59.7%
SVM flat	97.3%	89.3%	72.5%
MLP PCA	89.4%	94.7%	76.0%
LSVM PCA	77.9%	69.3%	59.7%
SVM PCA	74.4%	90.0%	80.0%
Architecture 2	95.9%	92.6%	78.0%
Architecture 1	98.4%	99.2%	88.9%
Proposed CNN architecture	98.1%	98.8%	86.5%

TABLE 19: Prediction accuracies for  $96 \times 96$  image inputs.

Model	Spectrogram	Scalogram	HHT
MLP flat	96.7%	91.7%	68.0%
LSVM flat	95.4%	84.4%	71.4%
SVM flat	98.7%	92.1%	69.0%
MLP PCA	96.3%	97.6%	85.0%
LSVM PCA	87.1%	74.5%	65.4%
SVM PCA	28.6%	84.4%	93.1%
Architecture 2	96.0%	96.0%	79.5%
Architecture 1	99.7%	99.8%	97.4%
Proposed CNN architecture	99.5%	99.5%	97.6%

TABLE 20: CWR paired two-tailed  $t$ -test  $p$  values.

Image type	Architecture 1	Architecture 1	Architecture 2	Architecture 2
	$32 \times 32$	$96 \times 96$	$32 \times 32$	$96 \times 96$
Scalogram	0.001	0.004	0.040	0.221
Spectrogram	0.022	0.000	0.000	0.211
HHT	0.005	0.784	0.000	0.000

PCA of the pixel intensities varied across different modeling and image selection. Spectrograms outperformed scalograms except for SVM PCA. The optimal modeling method using traditional techniques varied. HHT's optimum was SVM PCA, spectrograms were SVM flat, and for scalograms, MLP PCA was optimal.

Like the MFPT results, resolution loss from the reduction in image from  $96 \times 96$  to  $32 \times 32$  influenced the classification accuracies. Like the MFPT results, there was a slight drop in the scalogram accuracies between the two images sizes except for SVM PCA modeling. All methods suffered a little from the resolution drop; however, HHT again was the most affected.

The proposed architecture either outperformed or had statistically identical results with the other architectures. Table 20 outlines the results of the  $t$ -test values for the CWR data. The same hypothesis test as the MFPT data set was used for comparison.

Table 21 outlines the confusion matrix results for the CWR data set on  $96 \times 96$  scalograms. The values are horizontally normalized by class. From this, the following four tables of metrics were derived.

From the results for accuracy (Tables 18 and 19) and precision, sensitivity, specificity, and  $F$ -measure (Tables 22–25, resp.), one can say that, overall, the proposed architecture outperforms or is compatible with the other CNN architectures for diagnostic classification tasks with far fewer learnable parameters. The benefits of the additional nonlinear expressivity provided by the double layer approach in the proposed architecture are still present, but the images show the CWR data set has an overall better quality of measurement with far less noise.

## 7. Scalograms with Noise

To evaluate the robustness of the CNN architectures, white Gaussian noise was injected into the signals to evaluate how

TABLE 21: Confusion matrix for CWR (a)  $96 \times 96$  and (b)  $32 \times 32$  scalograms for the proposed architecture.

(a)				
	N	BF	IR	OR
N	<b>98.4%</b>	0.6%	0.0%	1.0%
BF	0.0%	<b>99.8%</b>	0.0%	0.2%
IR	0.0%	0.0%	<b>100%</b>	0.0%
OR	0.2%	<u>0.0%</u>	0.0%	<b>99.7%</b>
(b)				
	N	BF	IR	OR
N	<b>97.0%</b>	2.0%	0.0%	1.0%
BF	0.5%	<b>99.1%</b>	0.0%	<u>0.3%</u>
IR	0.0%	0.0%	<b>100%</b>	0.0%
OR	0.6%	0.6%	0.0%	<b>98.8%</b>

the deep learning framework handles the noise within a scalogram. Five and ten percent (20 and 10 signal to noise ratio (SNR), resp.) Gaussian noise was used via the wgn function on the raw signal within Matlab. Additionally, the noisy images were randomly sampled without replacement to generate a 50 : 50 mix with images of the raw signal (zero noise). The MFPT data set was chosen for this analysis as it had a higher amount of noise in the baseline and outer race images. Examples of those images can be seen in Table 26.

From these images the models were trained and assessed. Those results can be found in Table 27. Both architectures 1 and 2's prediction accuracy suffered from the injection of noise. This is due in part to only having one convolutional layer before pooling, therefore limiting the richness of the features for the final predictions. The inclusion of an additional convolutional layer within the proposed architecture prior to the pooling layer results in a much richer feature and



TABLE 22: Precision for CWR data set.

Model	Proposed CNN architecture	Architecture 1	Architecture 2
Scalogram $32 \times 32$	98.6%	99.2%	93.0%
Scalogram $96 \times 96$	99.4%	99.8%	96.7%
Spectrogram $32 \times 32$	98.0%	98.4%	95.8%
Spectrogram $96 \times 96$	99.5%	99.7%	96.7%
HHT $32 \times 32$	84.1%	85.4%	74.5%
HHT $96 \times 96$	97.0%	97.2%	82.5%

TABLE 23: Sensitivity for CWR data set.

Model	Proposed CNN architecture	Architecture 1	Architecture 2
Scalogram $32 \times 32$	98.7%	99.2%	92.7%
Scalogram $96 \times 96$	99.5%	99.8%	96.2%
Spectrogram $32 \times 32$	98.0%	98.3%	95.8%
Spectrogram $96 \times 96$	99.5%	99.7%	96.2%
HHT $32 \times 32$	84.2%	85.5%	74.4%
HHT $96 \times 96$	97.1%	97.3%	82.0%

TABLE 24: Specificity for CWR data set.

Model	Proposed CNN architecture	Architecture 1	Architecture 2
Scalogram $32 \times 32$	99.6%	99.7%	97.4%
Scalogram $96 \times 96$	99.8%	99.9%	98.7%
Spectrogram $32 \times 32$	99.3%	99.4%	98.6%
Spectrogram $96 \times 96$	99.8%	99.9%	98.7%
HHT $32 \times 32$	94.2%	94.7%	90.1%
HHT $96 \times 96$	99.0%	99.0%	93.4%

TABLE 25:  $F$ -measure for CWR data set.

Image type	Proposed CNN architecture	Architecture 1	Architecture 2
Scalogram $32 \times 32$	98.7%	99.2%	92.8%
Scalogram $96 \times 96$	99.5%	99.8%	96.4%
Spectrogram $32 \times 32$	98.0%	98.4%	95.8%
Spectrogram $96 \times 96$	99.5%	99.7%	96.4%
HHT $32 \times 32$	84.0%	85.4%	74.4%
HHT $96 \times 96$	97.0%	97.2%	82.1%

the increased nonlinearity helps the architecture handle noise better than the other architectures here examined.

## 8. Traditional Feature Extraction

To have a direct comparison with the standard fault diagnostic approach that relies on manually extracted features, we now examine the use of extracted features as an input to the CNN architectures discussed in this paper. The architectures were modified slightly to accommodate the vector inputs; however, the double convolutional layer followed by a pooling layer architecture was kept intact.

**8.1. Description of Features.** The vibration signals were divided into bins of 1024 samples each with an overlapping of 512 samples. Each of these bins was further processed to extract the following features from the original, derivative, and integral signals [39, 40]: maximum amplitude, root mean square (RMS), peak-to-peak amplitude, crest factor, arithmetic mean, variance ( $\sigma^2$ ), skewness (normalized 3rd central moment), kurtosis (normalized 4th central moment), and fifth to eleventh normalized central moments. Additionally, the arithmetic mean of the Fourier spectrum [41], divided into 25 frequency bands along with the RMS of the first five IMFs (empirical mode decomposition), were used as features.

TABLE 26: MFPT  $96 \times 96$  scalogram images with noise injected.

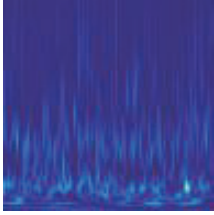
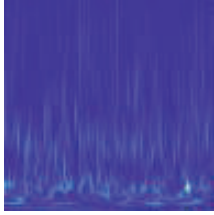
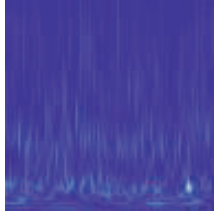
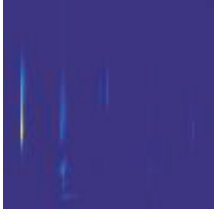


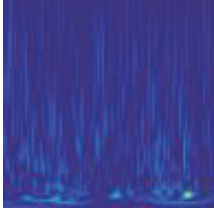

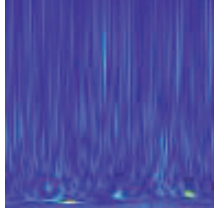
Data set	Baseline	5% noise	10% noise
Normal			
Inner race			
Outer race			

TABLE 27: Prediction accuracies for MFPT scalograms with injected noise.

Noisy image set	Architecture 2	Architecture 1	Proposed CNN architecture
$96 \times 96$ w/5% noise	96.6%	99.9%	99.9%
$96 \times 96$ w/10% noise	88.6%	91.8%	99.9%

TABLE 28: Prediction accuracies for CWR.

Model	20 epochs with early stopping	30 Epochs with no early stopping	No overlap
Architecture 2	75.2%	86.7%	67.2%
Architecture 1	90.4%	95.7%	87.2%
Proposed CNN architecture	83.1%	98.5%	93.6%

TABLE 29: Prediction accuracies for MFPT.

Model	20 epochs with early stopping	30 epochs with no early stopping	No overlap
Architecture 2	79.1%	80.9%	75.2%
Architecture 1	82.9%	75.1%	75.1%
Proposed CNN architecture	96.4%	93.8%	87.3%

In total, seventy-five features per bin were computed and each of the features was normalized using the mean and standard deviation of the first baseline condition.

**8.2. Application to CNN Architecture.** To evaluate the full set of features, the architecture of the CNN was changed slightly to incorporate all the features. The following iteration of the proposed architecture was used: Input[ $75 \times 15$ ] - 32C[ $75 \times 3$ ] - 32C[ $1 \times 3$ ] - 32P[ $2 \times 2$ ] - 64C[ $1 \times 3$ ] - 64C[ $1 \times 3$ ] - 64P[ $2 \times 2$ ] - FC[100]. Three different scenarios were examined: (1) twenty

epochs with early stopping and a stride of fifteen time steps with an overlap of eight times steps, (2) thirty epochs with no early stopping and stride of fifteen time steps with an overlap of eight times steps, and (3) twenty epochs with a stride of fifteen time steps with no overlap.

Tables 28 and 29 illustrate the difficulties the CNN architectures had when dealing with the manually constructed features: the prediction accuracies considerably dropped for all the CNN architectures for both MFPT and CWR data sets. Additional epochs without early stopping improved the

results; however, they are still well below the results of the image representations. For the MFPT data, early stopping and data overlap helped the accuracies. For the CWR data, the opposite is true for early stopping. The CWR data benefited from more epochs; however, the MFPT data suffered slightly from increased epochs.

The CNN strength is images and it has spatial awareness; therefore, the ordering of the features within the vector could influence the output predictions. It should be said that the sizes of the vectors and filters were chosen on the input and convolutional layers to minimize this effect.

CNNs are very good when the data passed through them is as close to the raw signal as possible, as the strength of the convolutional and pooling layers is their ability to learn features which are inherent representation of the data. If one manipulates the data too much by engineering features in the traditional sense, the CNNs do not perform as well. As illustrated from the results in Tables 28 and 29, the CNN architectures had difficulties in all scenarios. Moreover, even in this unfavorable scenario, the proposed architecture outperformed the others. The stacked convolutional layers, as in the case with infused noise, result in more expressive features to better capture the nonlinearity of the data. Thus, one can argue that, for CNNs, it is optimal to use an image representation of the raw signal instead of a vector of extracted features.

## 9. Concluding Remarks

Fault diagnosis of rolling element bearing is a significant issue in industry. Detecting faults early to plan maintenance is of great economic value. Prior applications of deep learning based models tended to be limited by their sensitivity to experimental noise or their reliance on traditional feature extraction. In this paper, a novel CNN architecture was applied to the time-frequency and image representations of raw vibration signals for use in rolling element bearing fault classification and diagnosis. This was done without the need for traditional feature extraction and selection and to exploit the deep CNNs strength for fault diagnosis: automatic feature extraction.

To determine the ability for the proposed CNN model to accurately diagnose a fault, three time-frequency analysis methods (STFT, WT, and HHT) were compared. Their effectiveness as representations of the raw signal were assessed. Additionally, information loss due to image scaling was analyzed which had little effect on the scalogram images, a slight effect on the spectrograms, and larger effect on the HHT images. In total, 189,406 images were analyzed.

The proposed CNN architecture showed it is robust against experimental noise. Additionally, it showed featureless learning and automatic learning of the data representations were effective. The proposed architecture delivers the same accuracies for scalogram images with lower computational costs by reducing the number of learnable parameters. The architecture outperforms similar architectures for both spectrograms and HHT images. The manual process of feature extraction and the delicate methods of feature selection

can be substituted with a deep learning framework allowing automated feature learning, therefore removing any confirmation biases surrounding one's prior experience. Overall, the CNN transformed images with minimal manipulation of the signal and automatically completed the feature extraction and learning resulting in a much-improved performance.

Fault diagnosis is a continually evolving field that has vast economic potential for automotive, industrial, aerospace, and infrastructure assets. One way to eliminate the bias and requirement of expert knowledge for feature extraction and selection is to implement deep learning methodologies which learn these features automatically. Industries could benefit from this approach on projects with limited knowledge, like innovative new systems.

## Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

## Acknowledgments

The authors acknowledge the partial financial support of the Chilean National Fund for Scientific and Technological Development (Fondecyt) under Grant no. 1160494.

## References

- [1] Z. Huo, Y. Zhang, P. Francq, L. Shu, and J. Huang, *Incipient Fault Diagnosis of Roller Bearing Using Optimized Wavelet Transform Based Multi-Speed Vibration Signatures*, IEEE Access, 2017.
- [2] J. Wang, J. Zhuang, L. Duan, and W. Cheng, "A multi-scale convolution neural network for featureless fault diagnosis," in *Proceedings of the International Symposium on Flexible Automation, (ISFA '16)*, pp. 1–3, Cleveland, OH, USA, August 2016.
- [3] M. Seera and C. P. Lim, "Online motor fault detection and diagnosis using a hybrid FMM-CART model," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 4, pp. 806–812, 2014.
- [4] K. A. Loparo, Loparo, K. A., Bearing Data Center, Case Western Reserve University, <http://csegroups.case.edu/bearingdatacenter/pages/welcome-case-western-reserve-university-bearing-data-center-website>, 2013.
- [5] A. Sharma, M. Amarnath, and P. Kankar, "Feature extraction and fault severity classification in ball bearings," *Journal of Vibration and Control*, 2014.
- [6] P. K. Wong, J. Zhong, Z. Yang, and C. M. Vong, "Sparse Bayesian extreme learning committee machine for engine simultaneous fault diagnosis," *Neurocomputing*, vol. 174, pp. 331–343, 2016.
- [7] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [8] C. Szegedy, W. Liu, Y. Jia et al., "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '15)*, pp. 1–9, Boston, Mass, USA, June 2015.
- [9] J. Tompson, R. Goroshin, A. Jain, Y. LeCun, and C. Bregler, "Efficient object localization using Convolutional Networks," in *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 648–656, IEEE, Boston, MA, USA, June 2015.

- [10] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "DeepFace: closing the gap to human-level performance in face verification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '14)*, pp. 1701–1708, Columbus, Ohio, USA, June 2014.
- [11] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "OverFeat: integrated recognition, localization and detection using convolutional networks," *Computer Vision and Pattern Recognition*, 2013.
- [12] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the 27th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '14)*, pp. 580–587, Columbus, Ohio, USA, June 2014.
- [13] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," in *Proceedings of the International Conference on Learning Representations (ICRL)*, p. 14, 2015.
- [14] D. Cires and U. Meier, "Multi-column Deep Neural Networks for Image Classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3642–3649, 2012.
- [15] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proceedings of the 26th Annual Conference on Neural Information Processing Systems (NIPS '12)*, pp. 1–9, Lake Tahoe, Nev, USA, December 2012.
- [16] Z. Chen, C. Li, and R.-V. Sanchez, "Gearbox fault identification and classification with convolutional neural networks," *Shock and Vibration*, vol. 2015, Article ID 390134, 10 pages, 2015.
- [17] X. Guo, L. Chen, and C. Shen, "Hierarchical adaptive deep convolution neural network and its application to bearing fault diagnosis," *Measurement: Journal of the International Measurement Confederation*, vol. 93, pp. 490–502, 2016.
- [18] L. Guo, H. Gao, H. Huang, X. He, and S. Li, "Multifeatures fusion and nonlinear dimension reduction for intelligent bearing condition monitoring," *Shock and Vibration*, vol. 2016, Article ID 4632562, 10 pages, 2016.
- [19] O. Abdeljaber, O. Avci, S. Kiranyaz, M. Gabbouj, and D. J. Inman, "Real-time vibration-based structural damage detection using one-dimensional convolutional neural networks," *Journal of Sound and Vibration*, vol. 388, pp. 154–170, 2017.
- [20] D. Lee, V. Siu, R. Cruz, and C. Yetman, "Convolutional neural net and bearing fault analysis," in *Proceedings of the International Conference on Data Mining series (ICDM) Barcelona*, pp. 194–200, San Diego, CA, USA, 2016.
- [21] O. Janssens, V. Slavkovikj, B. Vervisch et al., "Convolutional neural network based fault detection for rotating machinery," *Journal of Sound and Vibration*, vol. 377, pp. 331–345, 2016.
- [22] C. Zhang, P. Lim, A. K. Qin, and K. C. Tan, "Multiobjective Deep Belief Networks Ensemble for Remaining Useful Life Estimation in Prognostics," *IEEE Transactions on Neural Networks and Learning Systems*, vol. PP, no. 99, pp. 1–13, 2016.
- [23] L. Liao, W. Jin, and R. Pavel, "Enhanced Restricted Boltzmann Machine with Prognosability Regularization for Prognostics and Health Assessment," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 11, pp. 7076–7083, 2016.
- [24] G. S. Babu, P. Zhao, and X.-L. Li, "Deep convolutional neural network based regression approach for estimation of remaining useful life," *Lecture Notes in Computer Science (including sub-series Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9642, pp. 214–228, 2016.
- [25] F. Zhou, Y. Gao, and C. Wen, "A novel multimode fault classification method based on deep learning," *Journal of Control Science and Engineering*, Article ID 3583610, Art. ID 3583610, 14 pages, 2017.
- [26] H. Liu, L. Li, and J. Ma, "Rolling bearing fault diagnosis based on STFT-deep learning and sound signals," *Shock and Vibration*, vol. 2016, Article ID 6127479, 12 pages, 2016.
- [27] J. Bouvrie, "Notes on convolutional neural networks," Defense Technical Information Center, Center for Biological and Computational Learning, 2006.
- [28] Z. Feng, M. Liang, and F. Chu, "Recent advances in time-frequency analysis methods for machinery fault diagnosis: a review with application examples," *Mechanical Systems and Signal Processing*, vol. 38, no. 1, pp. 165–205, 2013.
- [29] J. Lin and L. Qu, "Feature extraction based on morlet wavelet and its application for mechanical fault diagnosis," *Journal of Sound and Vibration*, vol. 234, no. 1, pp. 135–148, 2000.
- [30] Z. K. Peng and F. L. Chu, "Application of the wavelet transform in machine condition monitoring and fault diagnostics: a review with bibliography," *Mechanical Systems and Signal Processing*, vol. 18, no. 2, pp. 199–221, 2004.
- [31] N. E. Huang, "Introduction to the hilbert–huang transform and its related mathematical problems," in *Hilbert–Huang Transform and Its Applications*, vol. 16 of *Interdisciplinary Mathematical Sciences*, pp. 1–26, World Scientific, Singapore, 2nd edition, 2014.
- [32] R. N. Meeson, "HHT sifting and filtering," in *Hilbert-Huang Transform And Its Applications*, vol. 5, pp. 75–105, Institute for Defense Analyses, Washington, DC, USA, 2005.
- [33] J. S. Smith, "The local mean decomposition and its application to EEG perception data," *Journal of the Royal Society Interface*, vol. 2, no. 5, pp. 443–454, 2005.
- [34] D. P. Kingma and J. Ba, "Adam: a method for stochastic optimization," arXiv preprint, 2014, <https://arxiv.org/abs/1412.6980>.
- [35] Bechhoefer, E., A Quick Introduction to Bearing Envelope Analysis, MFPT Data, <http://www.mfpt.org/FaultData/Fault-Data.htm.Set>, 2016.
- [36] H. Raveendran and D. Thomas, "Image fusion using LEP filtering and bilinear interpolation," *International Journal of Engineering Trends and Technology*, vol. 12, no. 9, pp. 427–431, 2014.
- [37] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Information Processing and Management*, vol. 45, no. 4, pp. 427–437, 2009.
- [38] W. A. Smith and R. B. Randall, "Rolling element bearing diagnostics using the Case Western Reserve University data: a benchmark study," *Mechanical Systems and Signal Processing*, vol. 64–65, pp. 100–131, 2015.
- [39] B. Samanta, K. R. Al-Balushi, and S. A. Al-Araimi, "Artificial neural networks and support vector machines with genetic algorithm for bearing fault detection," *Engineering Applications of Artificial Intelligence*, vol. 16, no. 7, pp. 657–665, 2003.
- [40] B. Samanta, "Gear fault detection using artificial neural networks and support vector machines with genetic algorithms," *Mechanical Systems and Signal Processing*, vol. 18, no. 3, pp. 625–644, 2004.
- [41] A. K. Nandi, C. Liu, and M. D. Wong, "Intelligent Vibration Signal Processing for Condition Monitoring," *Surveillance*, vol. 7, pp. 29–30, 2013.