공학박사학위논문

불충분한 고장 데이터에 대한 딥러닝 기반 회전 기계 진단기술 학습방법 연구

Training Approaches for Deep Learning Based Fault Diagnosis of Rotating Machinery Overcoming Fault Data Insufficiency

2020 년 2 월

서울대학교 대학원

기계항공공학부

김 현 재

Abstract

Training Approaches for Deep Learning Based Fault Diagnosis of Rotating Machinery Overcoming Fault Data Insufficiency

Hyunjae Kim
Department of Mechanical Engineering
The Graduate School
Seoul National University

Deep Learning is a promising approach for fault diagnosis in mechanical applications. Deep learning techniques are capable of processing lots of data in once, and modelling them into desired diagnostic model. In industrial fields, however, we can acquire tons of data but barely useful including fault or failure data because failure in industrial fields is usually unacceptable. To cope with this insufficient fault data problem to train diagnostic model for rotating machinery, this thesis proposes three research thrusts: 1) filter-envelope blocks in convolution neural networks (CNNs) to incorporate the preprocessing steps for vibration signal; frequency filtering and envelope extraction for more optimal solution and reduced efforts in building diagnostic model, 2) cepstrum editing based data augmentation (CEDA) for diagnostic dataset consist of vibration signals from rotating machinery, and 3) selective parameter freezing (SPF) for efficient parameter transfer in transfer learning. The first research thrust proposes noble types of functional blocks for neural networks in order to learn robust feature to the vibration data. Conventional neural networks including convolution neural network (CNN), is tend to learn biased features when the training data is acquired from small cases of conditions. This can leads to unfavorable performance to the different conditions or other similar equipment. Therefore this research propose two neural network blocks which can be incorporated to the conventional neural networks and minimize the preprocessing steps, filter block and envelope block. Each block is designed to learn frequency filter and envelope extraction function respectively, in order to induce the neural network to learn more robust and generalized features from limited vibration samples. The second thrust presents a new data augmentation technique specialized for diagnostic data of vibration signals. Many data augmentation techniques exist for image data with no consideration for properties of vibration data. Conventional techniques for data augmentation, such as flipping, rotating, or shearing are not proper for 1-d vibration data can harm the natural property of vibration signal. To augment vibration data without losing the properties of its physics, the proposed method generate new samples by editing the cepstrum which can be done by adjusting the cepstrum component of interest. By doing reverse transform to the edited cepstrum, the new samples is obtained and this results augmented dataset which leads to higher accuracy for the diagnostic model. The third research thrust suggests a new parameter repurposing method for parameter transfer, which is used for transfer learning. The proposed SPF selectively freezes transferred parameters from source network and re-train only unnecessary parameters for target domain to reduce overfitting and preserve useful source features when the target data is limited to train diagnostic model.

Keywords: Fault diagnosis
Rotating machinery
Rolling element bearing
Deep learning
Convolution neural networks
Transfer learning
Data augmentation
Diagnosis for different operating conditions

Student Number: 2012-23163

Table of Contents

List of Tables

List of Figures

Nomenclature

CNN	Convolutional neural network
MLP	Multi-layer perceptron
SPF	Selective parameter freezing
RNN	Recurrent neural network
y_i	Output feature of ith neuron
K_i	ith kernel in convolutional layer
$x(j)$	jth local region of input x
D	Output dimension of layer
b_i	Bias corresponding to ith kernel $\mathrm{K_i}$
σ	Activation function
ReLU	rectified linear unit
$\mathrm{p_i}$	ith feature after pooling
SVM	support vector machine
FC	Fully connected layer
VOC	Visual objective classes
BF	Ball fault
IRF	Inner race fault
ORF	Outer race fault
RPM	Revolution per minute
ALT	Accelerated life test
XS	Dataset of source in transfer learning
XT	Dataset of target in transfer learning
YS	Labels of source in transfer learning
YT	Labels of target in transfer learning
α	Hyper parameter of freeze threshold

η	Learning rate
θ_i	ith parameter in neural network
$\mathcal{L}$	Loss of neural network
N_y	Number of classes
CEDA	Cepstrum editing based data augmentation
Grad-CAM	Gradient class activation mapping
FIR	Finite impulse response

Chapter 1

Introduction

1.1 Motivation

As the size of the industry grows, it becomes very important to manage and maintain the system to avoid problems. Particularly in the manufacturing sector, if any part of the process fails, it can disrupt the entire production line and cause great losses. Therefore, there is a need for a technique for continuously measuring the state of the system and diagnosing a defect in the process based thereon. This technique requires basic techniques for diagnosing the state of the basic components of the system. The components of devices mainly used in industries such as manufacturing are concentrated in rotating machinery. For example, a motor for generating power includes a bearing for holding a rotating shaft, and a robot arm includes a gearbox for changing the rotational speed and torque of the motor. Therefore, the importance of developing diagnostic technology for these rotating machinery is increasing.

Fault diagnosis technique is a physics-based approach to identify the cause of a fault by diagnosing the physical principle of the object to be diagnosed and a data-driven technique that generates a diagnosis model based on a large amount of data collected from the object. Are classified. The physics based approach consists mainly of understanding the causes and mechanisms of faults and developing an index that can represent the level of faults. This approach allows the user to accurately understand and solve the cause and solution of the failure. However, because it is developed based on an understanding of the physical behavior of the system and how it works, it is very expensive to develop it for very large industrial systems or various element devices. Therefore, the emerging method is a data-driven approach that collects large amounts of data and analyzes it to model failures and develop diagnostic techniques. This technology is based on various signal processing and machine learning. Zidane technology developed through data-driven works well for very large systems and can dramatically reduce development costs. In particular, since the deep learning technique represented by the artificial neural network was developed, the accuracy and efficiency of the machine learning technique have been greatly increased, and fault diagnosis using deep learning has been very promising.

But the biggest drawback of deep learning-based fault diagnosis is that it requires large amounts of data. Indeed, in the industrial field, a great deal of data is being poured out every day, but it is hard to see all this data useful. Site-measured data may have been measured from the outset, whose purpose is unclear, or may have been collected at a quality that is not sufficient for diagnosis. But the most fatal problem is the lack of samples of fault cases to train deep learning models. The most useful sample for training a diagnostic model is the actual fault data measured from the model's application, but this is difficult to obtain. Because the losses caused by the failure of industrial systems are very large, scheduled maintenance to prevent them is always applied to the system.

Therefore, this thesis aims to study the techniques for fault diagnosis using machine learning including deep learning to diagnosis of the rotating machinery with high accuracy in the situation where the training data is insufficient. After studying and extending existing methods that allow deep learning techniques to be trained from small training data, this thesis finally achieve higher accuracy than existing techniques using physical knowledge of rotating machinery and that result in better and robust performance for fault diagnosis.

1.2 Research Scope and Overview

The goal of this thesis is to overcome the data insufficiency in fault diagnosis problem. For this, three thrusts are proposed. The first thrust is a novel architecture of neural network for more robust feature learning from insufficient fault data. Data augmentation is suggested as a second thrust for the data insufficiency, based on time-series signal processing technique. The third thrust proposes a new approach for the transfer learning, by using a noble scheme of parameter repurposing method. These thrusts are briefly introduced below.

Research Thrust 1: Filter-Envelope Blocks to Learn Robust and Generalized Features for Neural Networks

Research thrust 1 proposes two types of new network blocks which can be incorporated in conventional neural networks and enhance learnability of robust and generalized features from vibration data of rotating machinery.

Many deep learning approaches proposed for fault diagnosis in vibration signal set the input as raw signal which is directly acquired from the sensors. This is because these researches assume the powerful autonomous feature learning ability of the deep neural networks and attempt to minimize the efforts and cost for the building domain-knowledge based hand-crafted features. And many works support this assumption by the high diagnosis performance to the rotating machinery such as rolling element bearings and gearboxes.

Indeed, remarkable generalization ability of the deep neural network is repeatedly reported by tremendous study, especially in visual recognition area, such as image classification, object detection, and image sample generation. However, for specific case of the vibration signals, not much investigation has been done. CNN could have high generalization ability to the image samples because of some structural and parametric regularization, then CNN have highly functional topological feature learning ability.

Thusly, the neural network may generalize better though the proper additive neural structures for the vibration signals. Therefore, we propose two kind of network blocks which function as like frequency filters and envelope extractor. These are designed to have trainable parameters by the network optimizer and intended to learn best spectral features and shape of the outer lines of the signals which contain faulty characteristics.

Research Thrust 2: Cepstrum Editing based Data Augmentation in Vibration Signal for Fault Diagnosis

The motivation of the second research thrust is a lack of data augmentation options that can be used in situations where the training data of vibration signals are insufficient. For example, for colored pictures, people derive several augmentation methods from its nature. CNN achieved translation invariance compared to MLP. That is, CNN can achieve the same performance regardless of the orientation of the object to be recognized in the picture. However, there is no option that is appropriate for data augmentation of vibration signal, because the augmentation for image data can harm the physical properties of vibration signal and that can lead poor performance of the diagnosis model.

In this research thrust, we propose cepstrum editing based data augmentation, a specialized augmentation option for vibration signal. The data augmentation should keep the target data's physical property and even enhance the feature in the data. Therefore, we use cepstrum to extract feature of the fault and edit or modify them in cepstrum domain, and augment new samples by regenerating the vibration signal from edited cepstrum by inverse cepstrum transformation. Augmented samples increase the amount of the training samples and lead to higher performance of the diagnosis model.

Research Thrust 3: Selective Parameter Freezing for Parameter Transfer in Neural Networks

The third research thrust in this thesis is to enhance the diagnosis performance using transfer learning with insufficient training data. Transfer learning is a new branch of machine learning; specifically, it is a technique for transferring knowledge learned from a related source task to improve the performance of a target task[1-3]. Transfer learning is proving to be beneficial in engineering applications and in text and image classification[4, 5]. Transfer learning using a different source of image datasets has also been actively studied[6-8]. One of the most-studied practical examples is Google's Inception model[9]; this model required a large amount of time and computer resources to learn the ILSVRC dataset[10], which consists of 150,000 images and 1000 object categories. Using the Inception model as a starting point, new image classification models in different domains can be trained with less resources and data [11, 12].

The parameter transfer approach freezes the shallow layer near the input layer of the source network and fine-tunes the deep layer near the output layer to the target data. As the source network is larger and more complex than the size of the target dataset, it focuses on freezing more layers and mitigating overfitting and maintaining the generalization ability of the source network. The larger the target dataset size, it is the more likely to freeze the only shallow layers and attempt to increase the adaptability to the target domain. Therefore, parameter transfer is accompanied by several trials and errors to find the depth of the frozen layer which can achieve the best performance[13].

The limitation of this freezing and fine-tuning is to exclude the opportunity to learn the general features from the target data by freezing all the parameters of the shallow. The use of direct source features through freezing is accompanied by the risk of performance degradation as the distribution of source and target data differs. This problem occurs from the fact that parameter transfer approach does not offer any option for adjusting the freezing and fine-tuning inside a layer.

In this research thrust, we propose selective parameter freezing (SPF), an alternative approach to the two conventional parameter repurposing methods, parameter freezing and fine-tuning. SPF is devised to find a compromise for these methods that mitigate overfitting and, at the same time, have adaptability to target domains. The proposed SPF method allows retraining of only unnecessary parameters to the target data, while important source features remain. For SPF, we use parameter sensitivity as the output for determining available parameters. As a result, SPF can obtain better performance, as compared to conventional approaches.

1.3 Structure of the Thesis

The remaining of this thesis is organized as follows. Chapter 2 provides technical background with literature review including fault diagnosis techniques for rotating machinery and neural networks used for deep learning for basic understandings of the proposed approaches. Chapter 3 presents the experimental setting and test-bed for the acquisition of data to validate the proposed approaches. Chapter 4 proposes new types of neural network blocks for the incorporation of preprocessing steps and more optimized diagnostic model. In Chapter 5, cepstrum editing based data augmentation is introduced that is developed to augment an amount of data which from vibration sensor in rotating machinery. Then Chapter 6 suggests a selective parameter freezing method for parameter transfer with small dataset and its validation using bearing diagnostic dataset. Chapter 7 summarize the proposes in this thesis and further suggestions for the future research.

Chapter 2

Literature Review

This section describes the basic knowledge, formulas, and notations about neural networks and transfer learning that are necessary to understand the proposed approach. The scope of the description is not intended to cover all aspects of neural networks and the other techniques, but to provide enough context to understand the approach proposed in this thesis.

2.1 Deep Neural Networks

Neural networks have been proposed as autonomous knowledge learning algorithms that are not task-specific and not programmed. In general, a neural network refers to a technique for finding the relationship between an input and an output by adjusting the weight between connected nodes (so-called neurons). Since so many variations of neural networks have been developed, we focus here only on multilayer perceptron (MLP) and 1-D convolution neural network that used for validation of the proposed method.

1) Multilayer perceptron

The most basic structure, MLP is sometimes referred to as a 'vanilla network', because the latest neural network techniques, such as CNN, RNN, etc., all have their origin in the MLP structure. MLP receives input layer values, linearly projects with weight matrix W and bias vector b, and applies activation functions f to the output layer. The output of the previous layer is used as input to the next layer, deepening the structure. Output layer h, applied to input layer x is expressed by (2-1) and (2-2).

$$z = \mathrm{W}x + b \qquad (2\text{-}1)$$

$$h = f(z) \qquad (2\text{-}2)$$

MLP with L layers representing dataset (x, y) can be expressed as (2-3).

$$\boldsymbol{\hat{y}} = \boldsymbol{f}(\ldots \mathbf{W}^{[\mathbf{2}]}\boldsymbol{f}\left(\mathbf{W}^{[\mathbf{1}]}\boldsymbol{x} + \boldsymbol{b}^{[\mathbf{1}]}\right) + \boldsymbol{b}^{[\mathbf{2}]} \ldots + \boldsymbol{b}^{[\boldsymbol{L}-\mathbf{1}]}) \qquad (2\text{-}3)$$

where $x = (x^{(m)}; m = 1, \ldots, M)$, $x^{(m)} \in \mathbb{R}^{N_x}$, $y = (y^{(i)}; i = 1, \ldots, M)$, $y^{(i)} \in \mathbb{R}^{N_y}$ and one-hot encoded, $\hat{y}^{(i)}$is the prediction of $y^{(i)}$, Nx is the input size, Ny is the number of classes, and M is the number of samples in the dataset.

In MLP, the input and output nodes of each layer are fully-connected by the weight matrix. The activation function $f(\cdot)$ changes the distribution or adds non-linearity to the output, and various types are introduced (e.g., sigmoid, ReLU).

The output layer has a softmax function as an activation function different from other layers in the classification task. The purpose of the softmax function is to make the MLP's output layer a logistic regression layer, which is a classifier. The output of the logistic regression layer with respect to the linear projection z of the previous layer input is shown in (2-4).

$$\hat{y}_i = \frac{\exp\left(z_i^{[L]}\right)}{\sum_{i=1}^{N_y} \exp\left(z_i^{[L]}\right)} \qquad (2\text{-}4)$$

Through the softmax, the final outputs of the MLP are predictions of a multiclass classification problem that satisfies $\hat{y}_i \in [0,1]$ and $\sum_{i=1}^{N_y} \hat{y}_i = 1$.

The training network is set up to solve an optimization problem to find the $\mathrm{W}^{[l]}$ and $\mathrm{b}^{[l]}(l = 1,2,\dots,L)$ that best represent the output y to the input x. The loss function of this optimization takes prediction $\hat{y}$ and real output y and gives an error for the task. Usually, the cross-entropy loss in (2-5) is used to measure this error in the classification problem.

$$\mathcal{L}(y, \hat{y}) = -\sum_{i=1}^{B} y^{(i)} \log\left(\hat{y}^{(i)}\right) \qquad (2\text{-}5)$$

where, B is the batch size of the mini-batch gradient descent method. This is because minimizing the cross-entropy loss and maximizing the log likelihood of the logistic regression are exactly the same.

2) 1-D Convolution neural network

Convolution neural networks are generally used in video and image domains and are mostly used with 2-D inputs. However, in the fault diagnosis domain, the 1-D CNN model for inputting time series vibration data is also widely used. 1-D CNN includes a feature extractor and feature classifier, typically MLP, with multiple stages consisting of convolution layer and pooling layer as in 2-D CNN.

In Convolutional layer, features are extracted through convolution operation with several kernels in one local region of input. The output feature yi for input x by the convolution operation with ith kernel Ki is expressed as:

$$y_i = f\left(\sum_{j=1}^{D} K_i * x(j) + b_i\right) \qquad (2\text{-}6)$$

where, x(j) denotes the jth local region of the input x (j=1,2,…D, D is the output dimension), and bi is a bias corresponding to *i^{th}* kernel K_i, σ is activation function, $*$ is dot product between kernel K_i and local region of input *x(j)*.

The output y^i then passes pooling layer, which is commonly added right after the convolutional layer. The pooling layer is down-sampling of feature y_i in order to reduce the number of parameters and give invariance to locational shift to the feature. The most popular pooling function is max-pooling, which can be expressed as:

$$p_i = \sum_{j=1}^{R} \max_{(j-1)W+1 \leq t \leq jW} y_i(t) \qquad (2\text{-}7)$$

where pi denotes the *i^{th}* feature after pooling, $y^i(t)$ denotes the *t^{th}* unit of the previous convolutional layer output y^i, W is the width of the pooling window, *R* is the output dimension of the pooling layer.

Convolution neural network extracts highly abstracted features by stacking several stages consisting of convolutional and pooling layers. Then, the extracted feature is used to perform the classification operation, in which MLP called a fully connected (FC) layer. At this time, the input of the MLP becomes the output passing through the last pooling layer.

2.2 Transfer Learning and Parameter Transfer

Transfer learning using parameter transfer is a method of obtaining high performance by using a well-learned network as a feature extractor for similar tasks with insufficient data This approach is widely used in the fields of natural and medical image classification, satellite photo classification, and emotion recognition [14-16]. Recent studies have also shown that even in the field of fault diagnosis, a high-accuracy diagnostic model can be developed with fewer fault samples [17, 18].

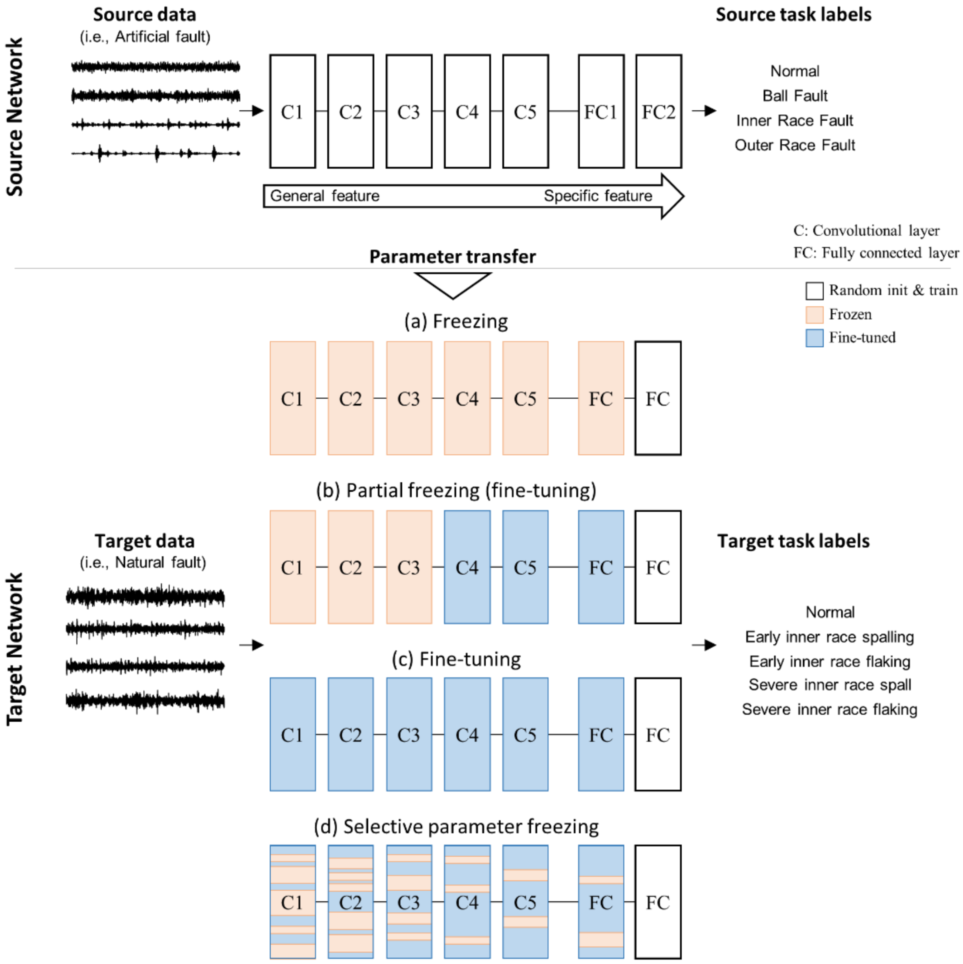


Figure 2-1 Various parameter repurposing method in parameter transfer approach

Parameter repurposing is essential step in parameter transfer methods. The parameter repurposing method is determined by specifying whether the transferred parameter for each layer is frozen or fine-tuned in the training process. All known parameter repurposing methods involve initializing the parameters of the last FC layer of the source network as shown in the Figure 2-1 (a)-(d). This is because the last layer that performs classification must be re-trained to match a new task with a different number or distribution of new label set. Oquab, et al. [16]used a pre-trained network [12] trained to ImageNet dataset [19] for the classification of a relatively small Pascal VOC dataset [19]. In this work, (a) parameter freezing method was used. Ng, et al. [14] and Marmanis, et al. [15] also used parameter freezing method to train emotion recognition and satellite photograph classification using ImageNet based pre-trained network. These works have in common that the source network is large and the source and target domains are similar. In the case of (b) partial freezing (or fine-tuning), the size of the pre-trained network is large, but the difference between the source and target domains is also large (i.e., from images to vibration) as shown in the work by Shao, et al. [17]. (c) fine-tuning is used when the source and target domains are similar

(i.e., vibration with different working conditions) and the size of the network is small, as in [20]. Summarizing the existing studies, the parameter repurposing method should be determined by the size of dataset, the size of network and the domain difference between source and target. Figure 2-1 (d) is the proposed repurposing method, selective parameter freezing which will be described in section III.

Chapter 3

Description of Testbed Data

This section describes the data and the details of its acquisition and the data augmentation method. To validate the performance of the SPF, we prepare two transfer tasks and corresponding two pairs of bearing diagnostic datasets. The first transfer task is transfer to a different fault size domain and the second task is transfer from artificial to natural fault domain.

3.1 Bearing Data I: Case Western Reserve University Data

The first datasets provided by Case Western Reserve University's Bearing Center are vibration data measured from normal and faulty ball bearing operations. The data includes vibrations acquired at the fault size of 0.18mm to 0.54mm, separately in the ball (BF), inner raceway (IRF), and outer raceway(ORF). The bearings with normal and faulty conditions were rotated by a motor of 0 to 3 horsepower (hp). The rotational speed of the bearing is in the range of 1,720 to 1,790 RPM, depending on the load of the motor. The vibration data was measured by an accelerometer attached at 12 o'clock on the drive end of the motor housing. Data were collected for 10 seconds at a sampling rate of 12,000 samples per second. Table 3-1 provides detailed information about the dataset.

Table 3-1 Details of Case western reserve university dataset

Fault type	Normal	Inner raceway fault			Outer raceway fault			Ball fault		
Fault size (mm)	-	0.18	0.36	0.54	0.18	0.36	0.54	0.18	0.36	0.54
Motor load (hp) (Speed (RPM))	1(1771~1774), 2(1748~1754), 3(1722~1729)									
Data size	120,000 points (10 seconds at 12,000 sample/sec rate)									

Table 3-2 Specification of test bearing SKF 7202

Item	Value
Inside diameter	15mm
Outside diameter	35mm
Thickness	11.5mm
Ball diameter	6.35mm
Number of ball	10

3.2 Bearing Data II: Accelerated Life Test Test-bed

The second dataset is also vibration data measured from rolling element bearings during operation. This bearing data was obtained from the ALT test-bed for rolling element bearings. The test bed consists of a motor for driving, two support bearings for supporting the shaft, a load, and the test bearings shown in Table 3-2, and a 3-axis accelerometer mounted vertically on the test bearing housing. In Figure 3-1, the configuration of the test-bed is shown. The data used is acceleration in the vertical direction at the ground. All data were collected at a sampling rate of 10,000 samples per second.

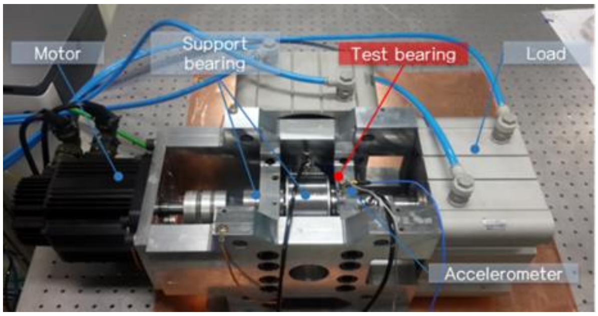


Figure 3-1 Configuration of ALT test-bed

In this ALT test bed, two purpose of data were acquired. First, artificial fault data. As with the case western reserve university dataset, four types of artificially implemented fault data were acquired including normal condition, but data for each health condition were measured for four rotating speeds as shown in the Table 3-3. Second, natural failure data. Two kinds of naturally occurring failure, inner race spall and inner race flaking data were obtained through the accelerated life test. These

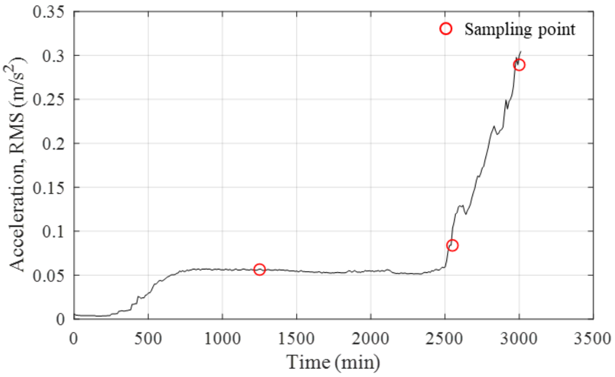


Figure 3-2 Sampling Points for ALT Test-bed

faults occurred at the same rotational speed and load. Also, normal, early failure, and severe failure samples were obtained according to the level of vibration amplitude as shown in the Figure 3-2. Details are given in Table 4.

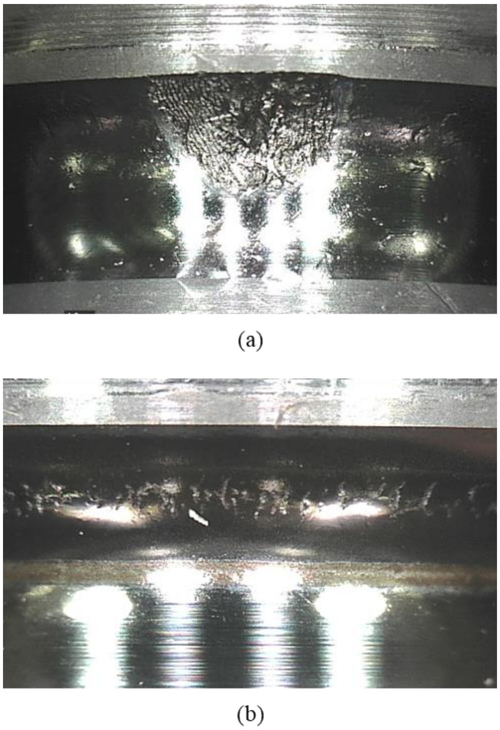
Table 3-3 Details of artificial fault dataset from ALT test-bed

Fault type	Normal	Ball fault	Inner race fault	Outer race fault
Fault size (mm)	-	0.4	0.4	0.4
Motor load (hp)	0.34			
Speed (RPM)	950, 1200, 1450, 1700			
Data size	100,000 points (10 seconds at 10,000 sample/sec rate)			

Table 3-4 Details of natural fault dataset from ALT test-bed

Fault type	Normal	Inner race spall		Inner race flaking	
Fault severity	-	Early	Severe	Early	Severe
Motor load (hp)	0.329 ~ 0.344				
Speed (RPM)	1450~1458				
Data size	100,000 points (10 seconds at 10,000 sample/sec rate)				

Spall and flaking shown in Figure 3-3 are not easy to be distinguish from their vibration signals when they occur in the same subcomponent such as inner raceway, which are presented in Figure 3-4. Because two fault types generate the same characteristic frequency of BPF (ball pass frequency), it has similar resonance frequency components in the enveloped FFT as shown in Figure 3-5. Especially in the early stage, it is not easy to discriminate because of low energy level of fault signal.



(a)

(b)

Figure 3-3 Closeshots of natural faults at inner raceway. (a) spalling (b) flaking

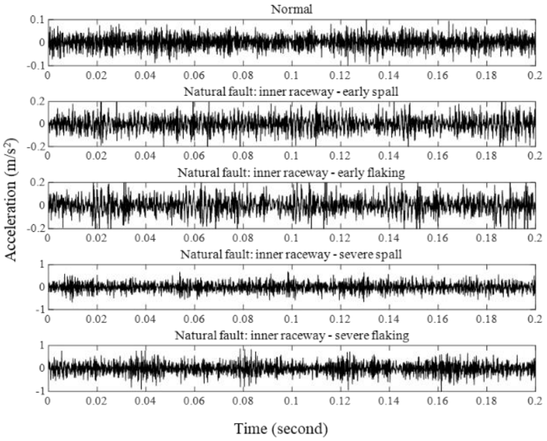


Figure 3-4 Times series plot of natural fault dataset

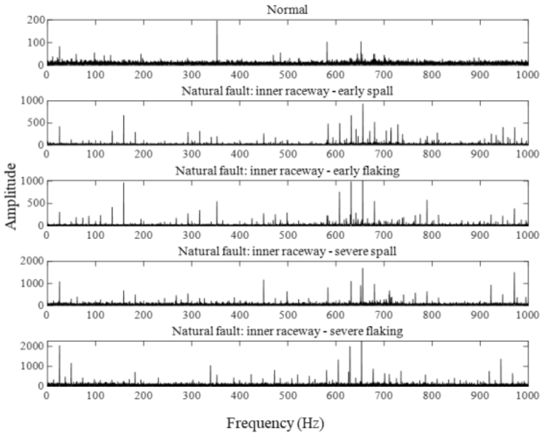


Figure 3-5 Enveloped FFT plot of natural fault dataset

Chapter 4

Filter-Envelope Blocks in Neural Network for Robust Feature Learning

Many studies have been done in deep learning based failure diagnosis models such as rolling element bearings have higher performance while minimizing the development cost of fault features according to the system. However, it is not easy to train a model with high generalization performance with only a few samples due to the nature of the industry where only limited fault samples can be obtained.

In the last study, there have been many studies for extracting fault features from vibration data, which can be classified into 1) time domain features and 2) frequency domain features. First, the time domain features includes root mean square (rms), the overall magnitude of the signal, or kurtosis that measures the magnitude of the impulse component of the signal distribution. Also, envelope of the signal is often used for time-domain feature extraction when an impulse signal by fault modulates another signal, it is difficult to separate by the frequency filter. Envelope analysis is useful for finding the modulating frequency of a shocked signal. In addition, many studies have been done to extract frequency components of a signal periodically generated by fault from rotating machinery to use them as diagnostic features. However, these methods require pre-processing such as frequency filtering and envelope extraction according to the characteristics of the target system, which is expensive because it has to be newly developed or adjusted to the systems.

The motivation of the deep learning-based diagnostic model is to reduce the cost of the development of diagnostic model which demands lots of domain knowledge by introducing end-to-end learning method. It has been reported that high diagnostic performance can be achieved with less preprocess or feature extraction process. However, when the deep learning based diagnostic model is applied to another system or the operating environment from trained system or condition, reduced performance or malfunctioning are reported. Transfer learning and domain adaptation are proposed to ease this problem. However, these approaches have a disadvantage that requires samples from new target systems or operating conditions because these approaches should be re-trained by target systems to knowledge transfer or adaptation.

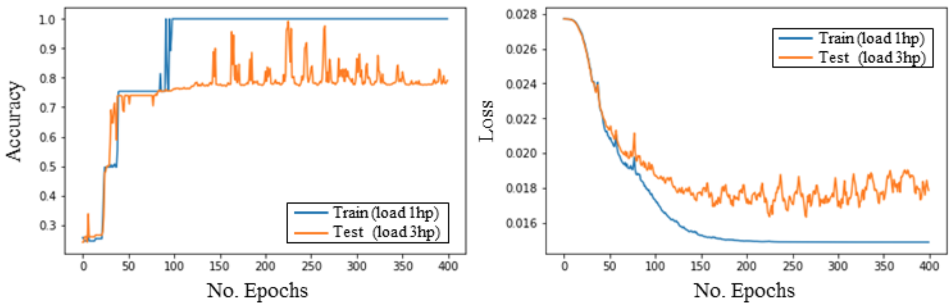
This chapter proposes Filter-Envelope Blocks in Neural Network for Robust Feature Learning to solve the weak generalization ability to vibration data of CNN and to improve the diagnostic performance of various driving environments or systems. The proposed approach includes two types of neural network blocks to prevent CNNs from learning local time series patterns while using raw time series signal as inputs and to better learn the time series patterns and frequency characteristics of signals. Finally, the proposed method is validated using bearing failure signals measured in various environments and systems, and to visualize the results and latent spaces of the learned shape filters.

4.1 Preliminary Study of Problems In Use of CNN for Vibration Signals

This subsection provides brief understandings of motivations of robust feature learning techniques for neural network by introducing some cases which the conventional deep learning approach fails in fault diagnosis for vibration signals. These are mainly demonstrated to how the neural network easily get trained 100% accuracy for the few samples and have poor generalization ability to the samples from different conditions.

4.1.1 Class Confusion Problem of CNN Model to Different Conditions

The problem we introduce in this subsection is that there is a tendency for diagnostic performance to be decreased to test data which is measured under different operating condition to the training data. Though the deep neural network have outstanding performance to the feature generalization, the network could learn only limited features when the given training data is limited. In visual recognition area, high quality large datasets are exist, therefore the deep neural network can extract well-generalized features for image recognition. However, in vibration based fault diagnosis, each application has different characteristics of system and training data can be acquired is limited to the case of test-bed or rare fault case in fields. Therefore, even for the well-trained model to the specific condition, the model give poor performance to the different condition, i.e., different operating load, different fault size. For example, for CWRU bearing dataset, the two model trained to the different load condition cannot give high accuracy to the other data. In Figure #, an example for the aforementioned case is given.



(a)

(b)

Figure 4-1 Results for the different load conditions as a training and test samples

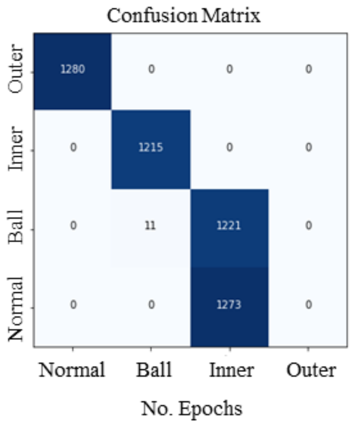
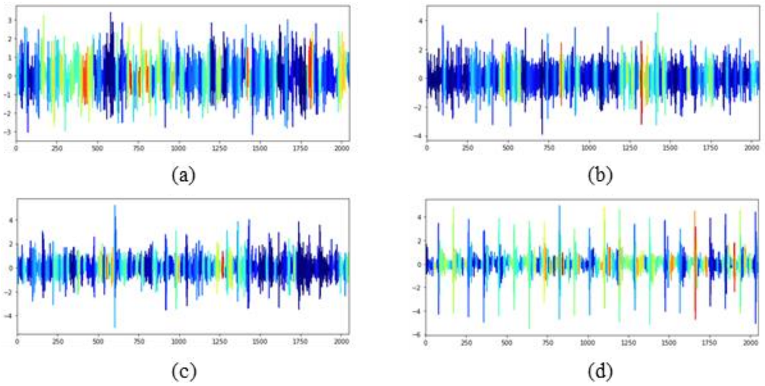


Figure 4-2 Confusion matrix for the different load conditions as a training and test samples

Figure 4-1 (a), (b), and Figure 4-2 show the training results when the level 3 fault size data of the CWRU bearing dataset is divided into operating load condition 1hp and 3hp and used as training and test data respectively. Each training and test data has four classes: normal, ball fault, inner race fault, and outer race fault. Looking at the accuracy trend according to the number of epoch, for training data, accuracy is achieved in order of 25%, 50%, 75%, and 100% within 100 epochs, which can be seen as steps of learning four classes sequentially. However, for the test data, even if the training continues after achieving 75% accuracy, no further improvement can be found, but rather the loss increases. In confusion matrix, the normal, ball fault, and inner race fault are well diagnosed, but the outer race fault is confused with the inner race fault. Depending on training approach or network structure, the above results may be different. However, the case that a class is completely misdiagnosed as another class occurs frequently in vibration signal based fault diagnosis. And this is called class confusion in this thesis.

The cause of this class confusion can be thought of as overfitting caused by too low variance within the dataset. Grad-CAM (gradient-class activation mapping[21]) allows to observe where the trained network is looking for vibration signal to diagnosis the signal. This is shown in Figure 4-3.



(a)

(b)

(c)

(d)

Figure 4-3 Visualization using Grad-CAM for the rolling element bearing diagnostic model

Figure (a), (b), (c), and (d) show the fractions of the vibration data that the model considers important to diagnosis fault for normal, ball fault, inner race fault, and outer race fault. It is represented in red color. Unlike human, the model is not seeing the impulse component do diagnose inner race and outer race fault, or the frequency component to classify normal and ball fault signal, but through local features of very narrow view range and this can be considered as overfitting because of lacking variance of the training data. Class confusion also is considered as this overfitting.

4.1.2 Benefits of Frequency Filtering and Envelope Extraction for Fault Diagnosis in Vibration Signals

CNN is the most widely used network structure for vibration-based fault diagnosis[22-30]. In fact, it has been adopted as a diagnostic model with a much higher frequency than MLP (Multi-layer Perceptron) and RNN (Recurrent neural network). The reason is the high performance that CNN has shown in many image-based applications. Therefore, many studies take an approach of diagnosing vibration data after converting it into an image through STFT[31-35] (Short time Fourier transform). In general, the features to be extracted from the vibration data for diagnosis include the frequency component and the outer line of the signal[36-40] (envelope). However, these features are difficult to extract with normal CNN structure. Because CNN learns local feature through convolution or cross-correlation operation in kernel area, it is difficult to find good feature from oscillation type data such as vibration data.

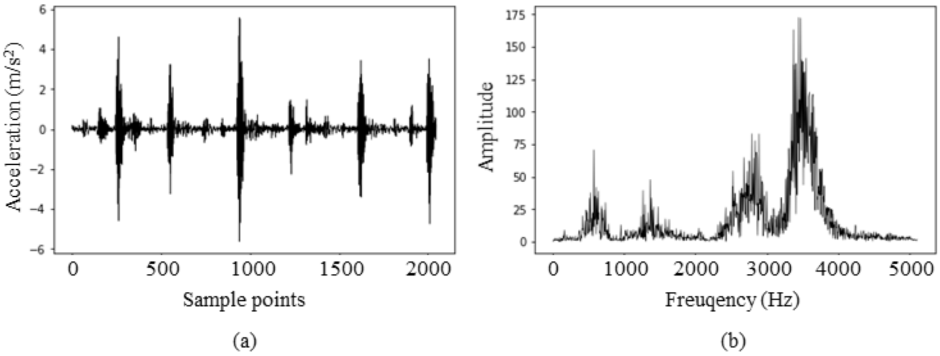


Figure 4-4 Outerrace fault samples (a) time-series plot (b) frequency spectrum

Therefore, vibration data needs preprocessing such as frequency filtering and envelope extraction in order to be learned by CNN. In this subsection, the benefits that these pretreatments bring to fault diagnosis are shown experimentally using the example used in 4.1.1. Figure 4-4 (a) is part of the falsely diagnosed outer race fault sample in the example of 4.1.1, and (b) is the corresponding frequency spectrum. As is known in many literatures, the feature of the outer race fault can be extracted from the resonance frequencies in the mid-range. Therefore, band-pass filtering was performed on the 2000-4000Hz region (high bimodal frequency region), which is the resonance frequency region of a given sample, and the result is shown in Figure 4-5.

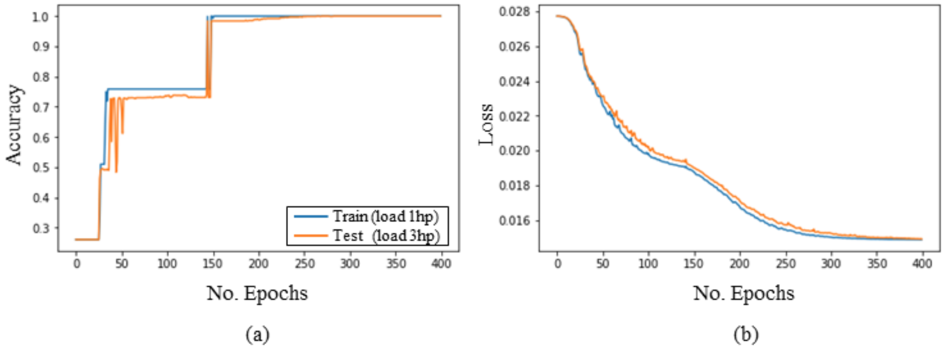


Figure 4-5 Training result after bandpass filter processing

The training results and signals after frequency preprocessing are shown in Figure 4-5. And learning results using preprocessed signals are very similar in learning trend of training and test data. The reason is that by extracting only the signals in the resonance frequency region, it eliminates fault-free signals and the impulse waveform due to the impact on the race surface becomes clear, making the network easier to learn this feature. Figures (a) and (b) show enlarged pictures of impulse waveforms before and after filtering, respectively.

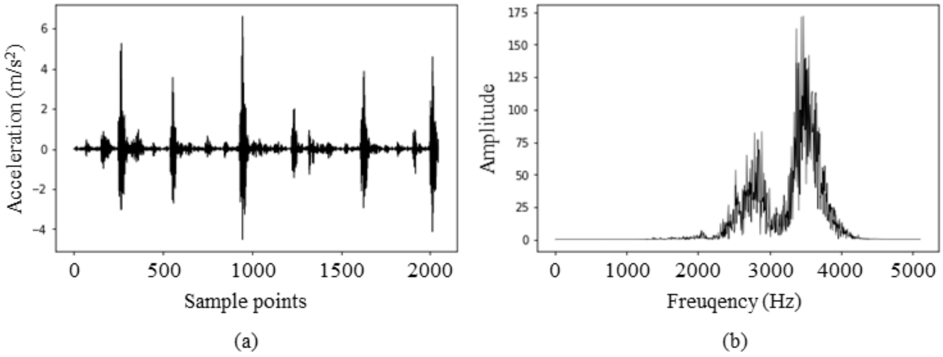


Figure 4-6 Outerrace fault samples after bandpass filtering for the resonance region

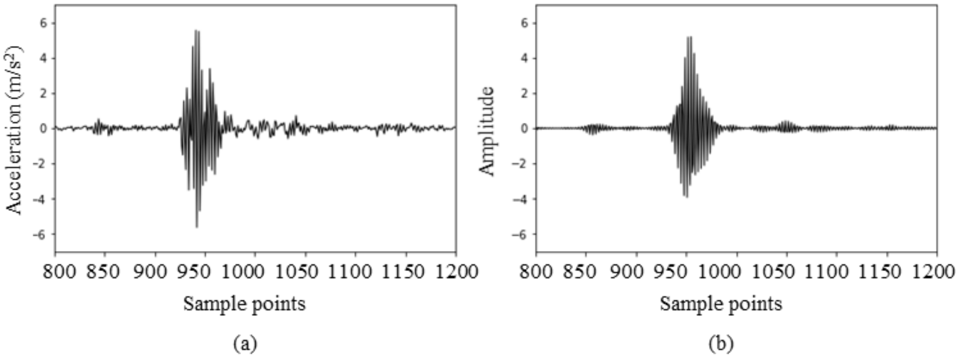


Figure 4-7 Enlarged pictures of outerrace fault impulse (a) before and (b) after bandpass filtering

Extracting the envelope of the signal is also an important preprocess for learning a good diagnostic model. As shown in Figure 4-8, the envelope is very important to find fault features and identify the type of the fault.

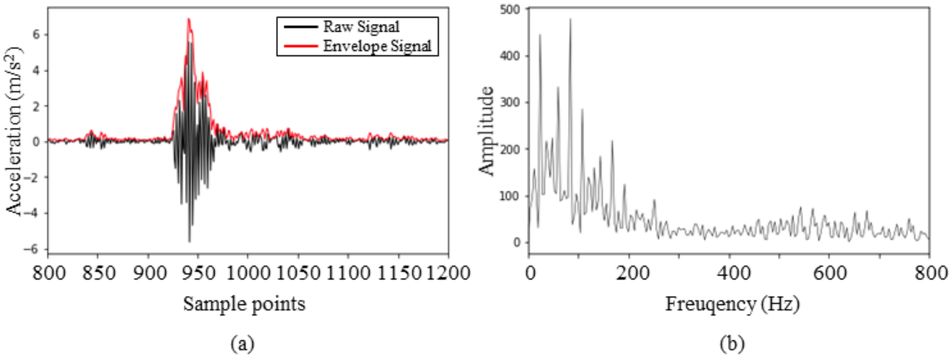


Figure 4-8 Raw and enveloped signals (a) time-series plot (b) frequency spectrum of enveloped signal

The envelope of the signal is an excellent feature in itself, but by Fourier transform, the frequency of impulse component appearing periodically can be used to estimate the type of fault. The result after preprocessing with envelope is shown in Figure 4-9.

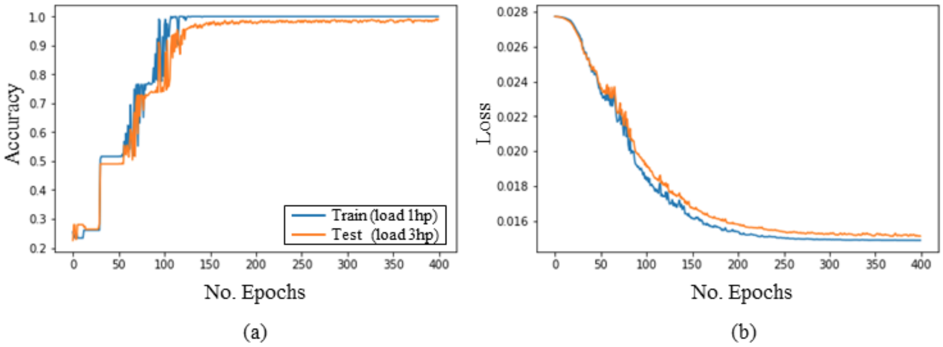


Figure 4-9 Results for the enveloped signals as training samples.

Similarly, high performance is achieved by the preprocessing, and faster accuracy convergence is achieved than frequency filtering. As such, the preprocessing reflecting the characteristics of the vibration signal has a great influence on the failure diagnosis performance. However, this preprocessing increases the hyper-parameters in the whole network training process. This is because, in frequency filtering, the cut-off range or window size selection for envelope extraction still needs to be determined by the user, so several cycles of preprocessing and model training have to be repeated to obtain optimal performance.

4.2 Proposed Network Block 1: Filter Block

In this subsection, we propose a filter block, a new type of network block that allows neural networks to construct their own band-filter and find the optimal band width. The convolution layer included in the proposed network block is designed to have the transfer function of the band-pass finite impulse response (FIR) filter as the weight. This filter block has two parameters for low-cut and high-cut frequency. These two parameters, along with the other parameters of the network, are learned by back-propagation to the values optimized for diagnosis.

4.2.1 Spectral Feature Learning in Neural Network

In order to show that the kernel, which acts as a filter of the existing CNN, is vulnerable to learning the characteristics of the frequency domain, the response of the frequency domain was examined by extracting the kernel weight of the first CNN convolution layer. An example of an analysis plot is shown in Figure 4-10 to look at the shape of the kernel and the frequency response to the input signal. Figure (a) shows the shape of the kernel trained in the first convolution layer of the network, (b) shows the input signal through the kernel, (c) shows the frequency spectrum of the signal passed through, and (d) shows white noise in the kernel. This is the frequency spectrum after passing.

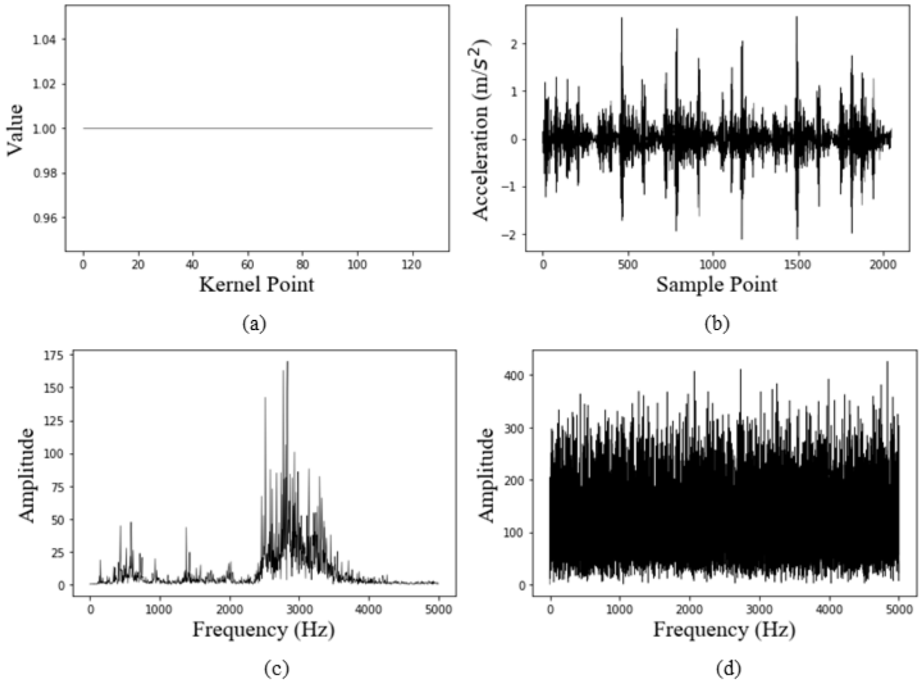


Figure 4-10 Example of (a) Kernel, (b) Kernel Filtered Signal, (c) Spectrum of Filtered Signal, and (d) Frequency Response of Filter

The performance of CNNs on raw signals is generally low, but sometimes high, so look at how the kernel was trained at high accuracy (95% ~).

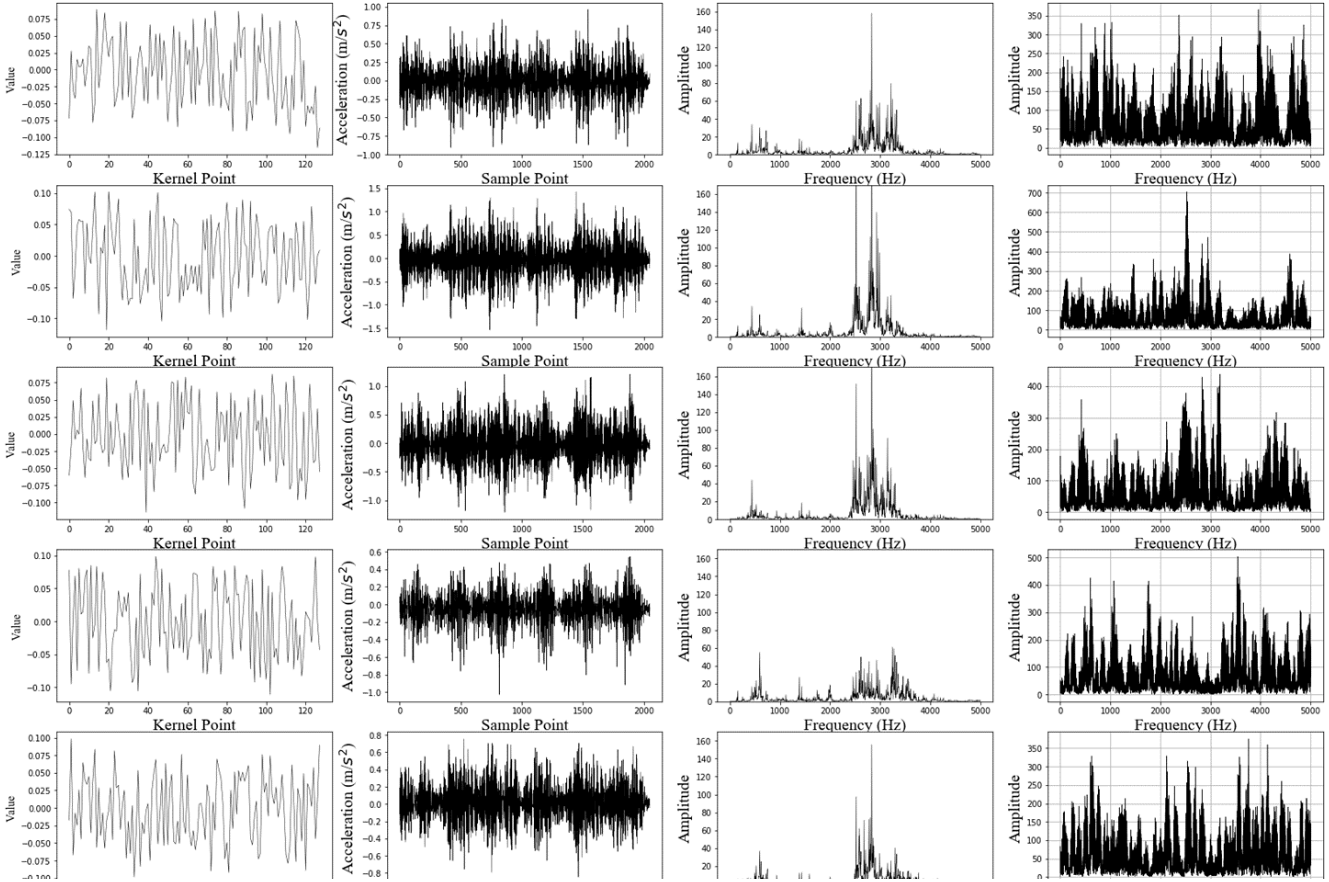


Figure 4-11 Examples of Trained Kernels from Vanilla CNN

Figure 4-11 shows the frequency response for the five kernels learned from the CNN. The training and test accuracy of the CNN is more than 95%, and the kernel of the model with the best accuracy among several trainings is extracted. First of all, the shape of the kernel is a form of noise that is difficult to find patterns in time series. The signal passing through the kernel is more impulse-reduced than the original signal, but there is no significant difference in the frequency spectrum from the original signal. However, a slight change in the resonance frequency region was found. The frequency filters of these kernels take the form of intermittent, intensive complex frequency amplifiers. These kernels can be advantageous for troubleshooting by passing patterns in the time series, but these dense and diverse frequency filters can cause overfitting on the training data.

4.2.2 FIR Band-pass Filter in Neural Network

The key idea of the filter block proposed in this subsection is to prevent overfitting by having one kernel learn to act as a band-pass filter. For example, the more complex the frequency response of a filter, the less likely the feature of the test data will deviate from the frequency response of the filter. However, if the kernel learns the pass response for successive bands, it can cope with failure characteristics in the slightly different frequency domain.

The kernel modification to be learned is designed to take the form of an FIR filter[41, 42]. CNN convolutes the kernel with input to produce output. This is the same as Equation (4-1).

$$y(n) = \sum_{k=0}^{N-1} h(k)x(n-k) \qquad (4\text{-}1)$$

where h is the weight of the convolution layer. CNN skips the flip of the input and cross-correlation, which reduces the amount of computation but gives the same performance. If the kernel has the frequency response of a band-pass filter, a finite impulse response (FIR) filter can be applied. The frequency response of the filter with respect to the input x [n] in the frequency domain can be expressed by equations (4-2) and (4-3) by the convolution theorem.

$$Y(\omega) = \mathcal{F}\{x * h\} = \mathcal{F}\{\dot{x}\} \cdot \mathcal{F}\{h\} = X(\omega) \cdot H(\omega) \quad (4\text{-}2)$$

$$\mathrm{y[n]} = \mathrm{x[n]} * \mathrm{h[n]}\mathcal{F}^{-1}\{X(\omega) \cdot H(\omega)\} \quad (4\text{-}3)$$

where $\mathcal{F}$ and $\mathcal{F}^{-1}$ denote the discrete-time Fourier Transform (DTFT) and its reverse. Hence, $H(\omega)$ is the filter’s frequency response. This is defined by Equation (4-4) by Z-transform.

$$H_{2\pi}(\omega) = \widehat{H}(e^{j\omega}) \quad (4\text{-}4)$$

where $\omega = \frac{2\pi f}{f_s}$ is normalized frequency.

If you want the filter, that is, the kernel's frequency response to have a frequency response like the one in Figure 4-12, you can do this by inverse Fourier transform, as shown in Equation (4-5).

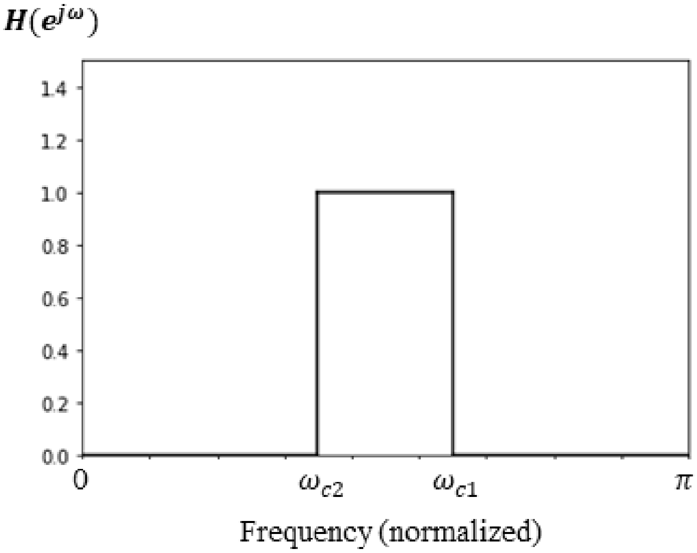


Figure 4-12 Frequency Response of Ideal Band-pass Filter

$$h[n] = \begin{cases} \frac{\sin(\omega_{c2}(n-M))}{\pi(n-M)} - \frac{\sin(\omega_{c1}(n-M))}{\pi(n-M)}, & (n \neq M) \\ \omega_{c2} - \omega_{c1}, \quad (n = M) \end{cases} \tag{4-5}$$

The FIR Band-pass filter form of Equation (6-4) is shown in Figure 4-13.

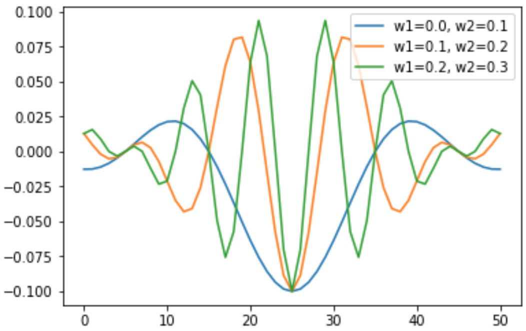


Figure 4-13 FIR band-pass filter in time-series domain for various cut-off frequencies

If the width of one kernel is w, the kernel has two parameters regardless of w. A typical convolution kernel has as many parameters as it is wide. Thus, replacing a 128-kernel layer with a filter block results in a 64x reduction in the number of parameters. This limits the functionality of the layer to frequency filtering rather than feature extraction, but it is difficult to expect good convolution layers to obtain features from raw vibration signals. And this substitution allows for strong structural regulation.

4.2.3 Result and Discussion

The diagnostic accuracy and learning results for the problem of 4.1.1 are shown in Figure 4-14.

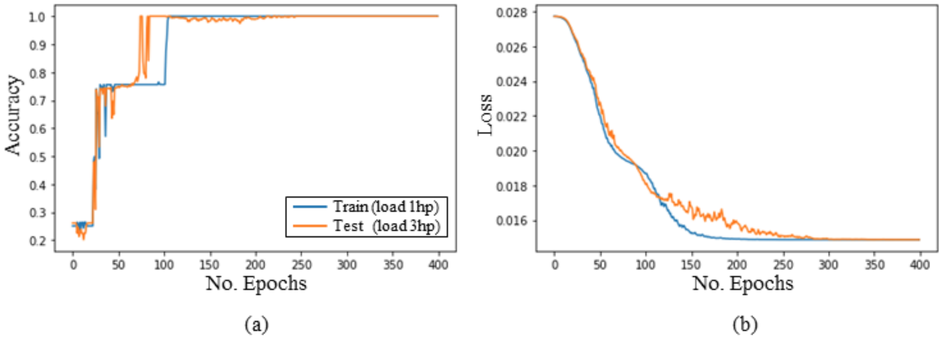


Figure 4-14 Training Result for Bearing Diagnosis

Similar to the pretreatment through the frequency filter, the filter block gave fast convergence and high accuracy. In addition, the learned kernels all take the form of FIR band-pass filters and are trained to filter different frequency domains.

4.3 Proposed Neural Block 2: Envelope Block

In this subsection, we propose an envelope block that allows the network to learn the modification of the vibration signal on its own and to learn the fault features from it. Hilbert transforming a continuous case signal to obtain an analytic signal and extracting the envelope from this absolute value does not increase the cost of the network learning process since no hyper-parameters are needed. However, the length of the discrete Hilbert filter exists as a hyper-parameter to extract the envelope of discrete signals acquired in the field. There are also several empirical approaches for envelope extraction. Empirical envelopes include moving rms (root mean square) and peak-spline methods[43, 44]. These empirical approaches yield different types of envelopes, which can lead to different diagnostic results depending on the characteristics of the vibration signal. The cost of preprocessing is therefore due to the choice of envelope method and the resulting hyper-parameters. The proposed envelope block aims to reduce this preprocessing cost by learning how to extract the envelope by itself.

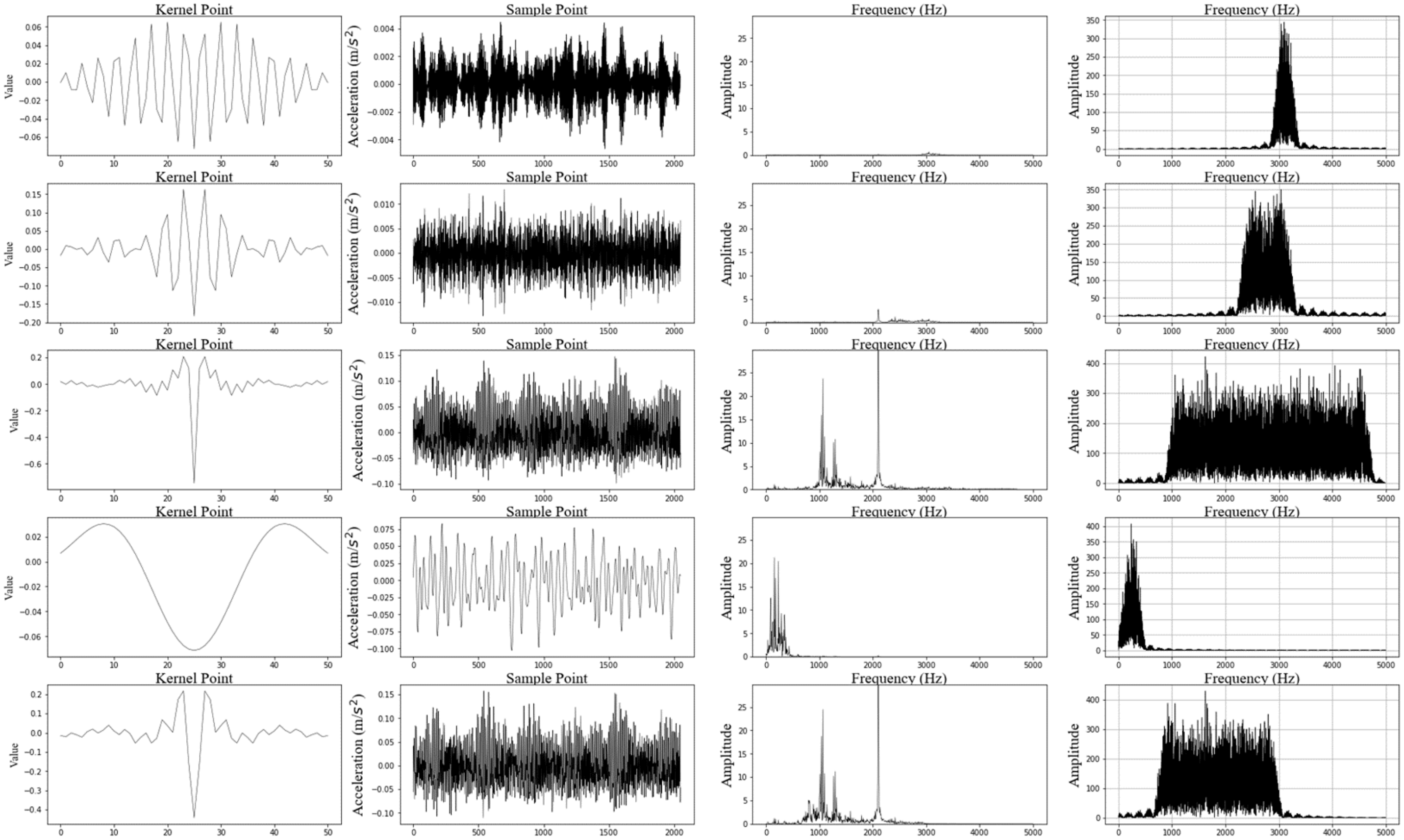


Figure 4-15 Trained Kernels from Filter Block

4.3.1 Max-Average Pooling Block for Envelope Extraction

The most theoretical way for a network to extract envelopes is to implement a discrete Hilbert transform in the network. Convoluting the Hilbert transfer function to a signal and obtaining the root of squared sum of the signal yields an analytic envelope. The disadvantage of this method, however, is that it is difficult to find a derivative for the back-propagation of the root of squared sum. Therefore, this study uses a similar but slightly modified method to EMD[45-47] (empirical mode decomposition). The EMD calculates the local maxima of the signal with the peak detector to obtain the empirical envelope and spline interpolates these maxima. This approach, however, needs to be further simplified because back detection of peak detection and spline operations is difficult.

In this study, empirical envelope is implemented using max-pooling and average-pooling which are used in the existing convolution layer. This method uses max-pooling to find the local maxima of the signal, and smooths the signal with average-pooling to extract the envelope. This is shown in Figure 4-16.

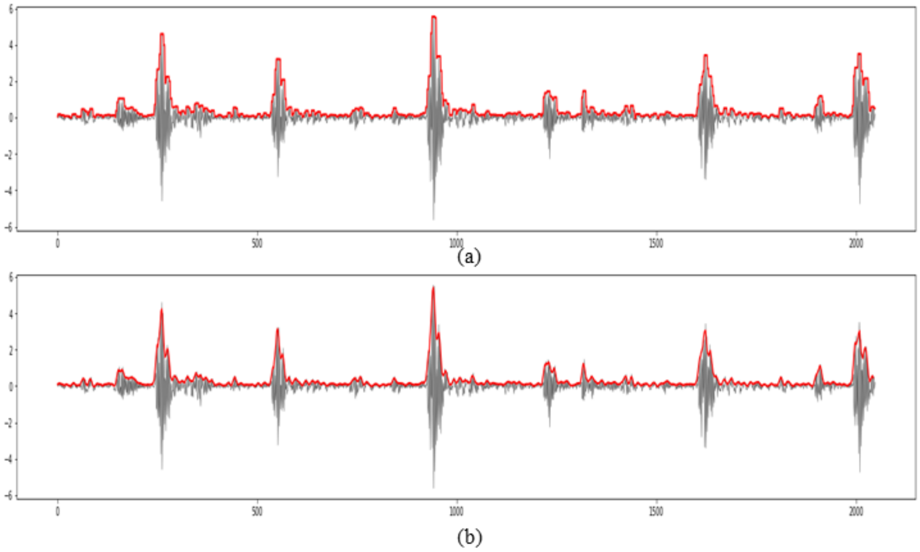


Figure 4-16 Example of Empirical Envelope using Max and Average Pooling

Figure 4-16 (a) shows the local maxima using the max-pooling layer of width 8, and (b) shows the envelope obtained by smoothing using the average-pooling of width 8.

4.3.2 Adaptive Average Pooling for Learnable Envelope Extractor

The disadvantage of max-average-pooling proposed in 4.3.1 is that the network cannot learn by adjusting the characteristics of the envelope itself. Therefore, in this paper, we propose an adaptive average-pooling method and adjust the smoothness of the envelope by changing the average window size. For this purpose, a dual-sigmoid window is proposed. The formula and shape are shown in Equation (4-6) and Figure 4-17.

$$\sigma_{dual}(n) = \sigma(n + a * s) + -\sigma(n - a * s) \qquad (4\text{-}6)$$

where, a is a network parameter for window size, n is kernel point, and s is slope of the sigmoid.

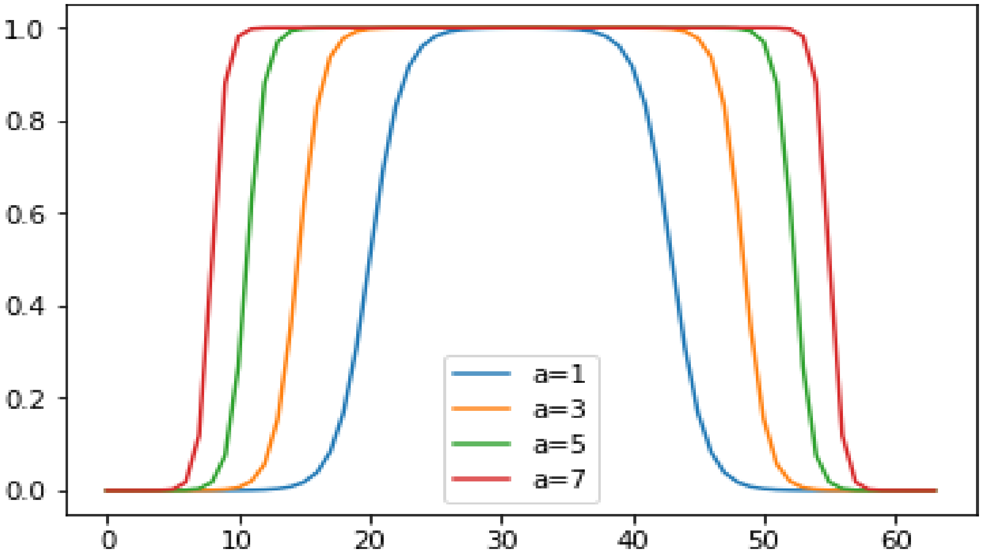


Figure 4-17 Adaptive Average Window for Pooling Layer

As shown in Figure 4-17, you can adjust the parameter a to size the window. By convoluting the dual-sigmoid with the max-pooled signal, an envelope of varying smoothness can be obtained, which is shown in Figure 4-18.

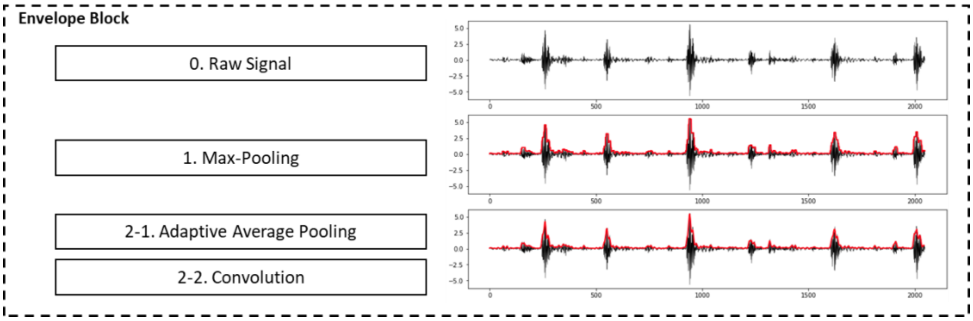


Figure 4-18 Process of Envelope Block

4.3.3 Result and Discussion

The diagnostic accuracy and learning results for the problem of 4.1.1 using the Envelope Block are shown in Figure 4-19.

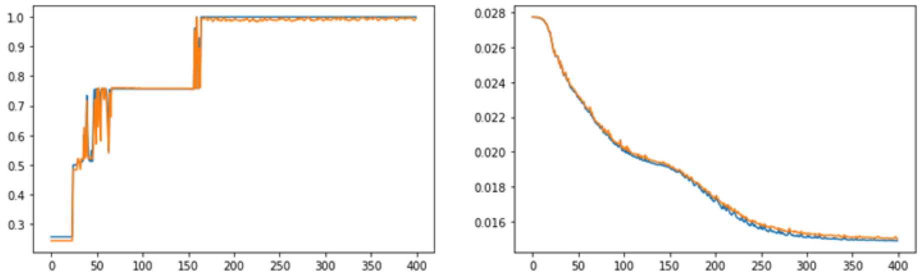


Figure 4-19 Training Result of CNN using Envelop Block

The accuracy of Figure 4-19 shows that the training and test data show almost the same accuracy and convergence trend even though they are acquired under different operating load conditions. This is because the envelope plays an important role in diagnosis. In addition, since the convolution layer learns that signal and kernel similarity, that is, cross-correlation, is inherently high, it is difficult to learn the outer line shape of oscillation data.

Figure 4-20 also shows the result of an envelope block using five kernels. The first column shows the adaptive average window for each kernel. Each kernel derives various envelopes by learning windows of different widths. As a result, a number of outputs are derived while adjusting the degree of smoothness from the same max-pooling result. Therefore, the parameter a is determined to be good enough for the subsequent convolution layer to learn its characteristics.

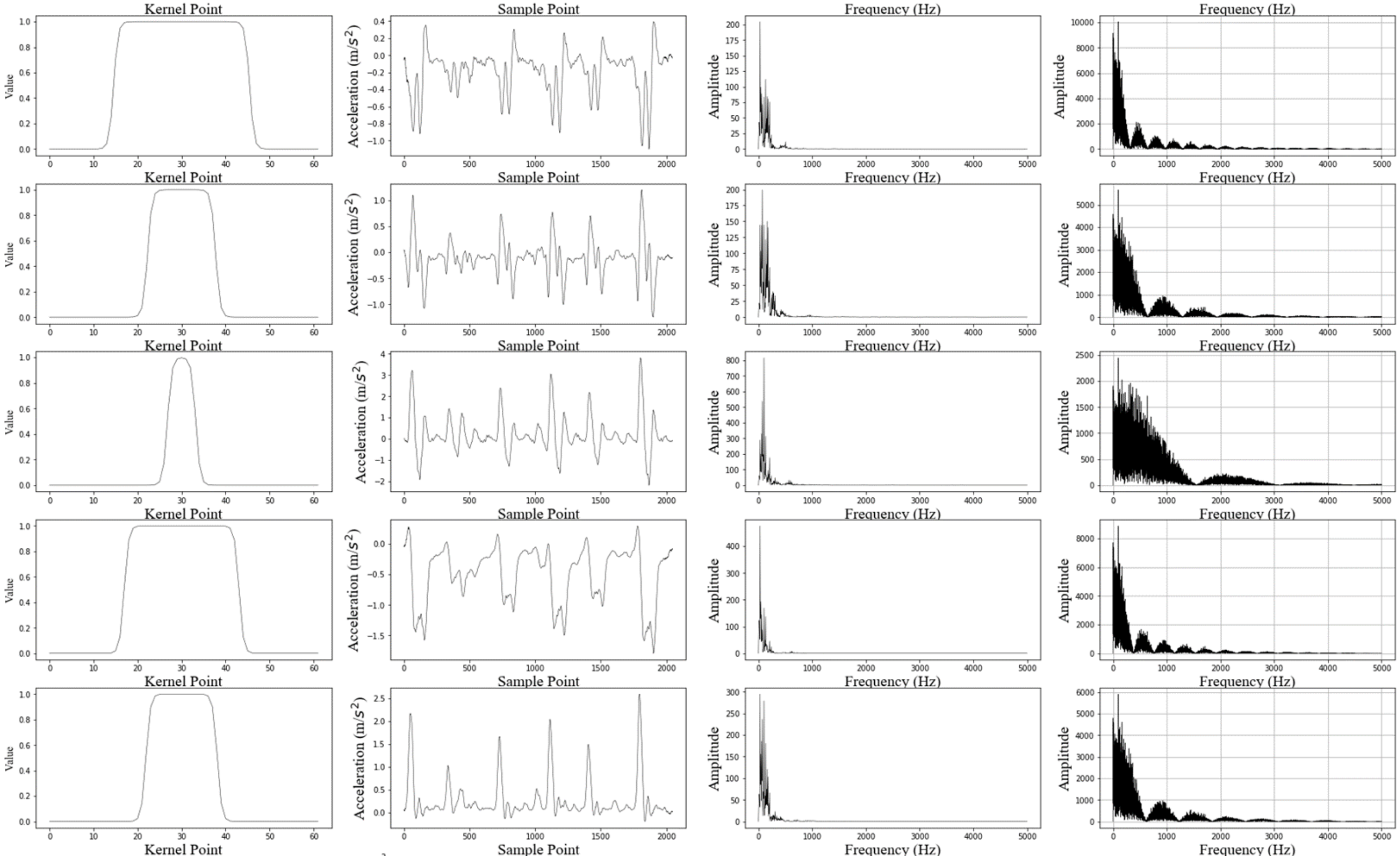


Figure 4-20 Result of Trained Envelope Block

4.4 Filter-Envelope Network for Fault Diagnosis

By combining the two types of blocks proposed above, it is possible to diagnose the failure of the rotor. As an example of rolling element bearing, band-pass filtering is performed by finding a resonance frequency in which a failure characteristic exists. In addition, frequency analysis is performed by extracting the envelope of the filtered signal to find the fault characteristic frequency to determine the kind of failure.

4.4.1 Combinations of Filter-Envelope Blocks for the use of Rolling Element Bearing Fault Diagnosis

The proposed method mimics the process of extracting the envelope from the aforementioned process, and does not find the fault characteristic frequency, but distinguishes the fault type by looking at the topological characteristic of the envelope, that is, its shape. This process is shown in Figure 4-21.

First, we take input and learn various types of FIR band-pass filter with filter block. The filtered signal is passed to the envelope block and is sent to the fully connected, sorted phase through global average pooling. The reason is that the size of the learned filter area itself can be an important feature. So this connection helps determine the band-pass for extracting the envelope, leading to faster learning. Next, we learn the envelope of various smoothness from the filtered signal.

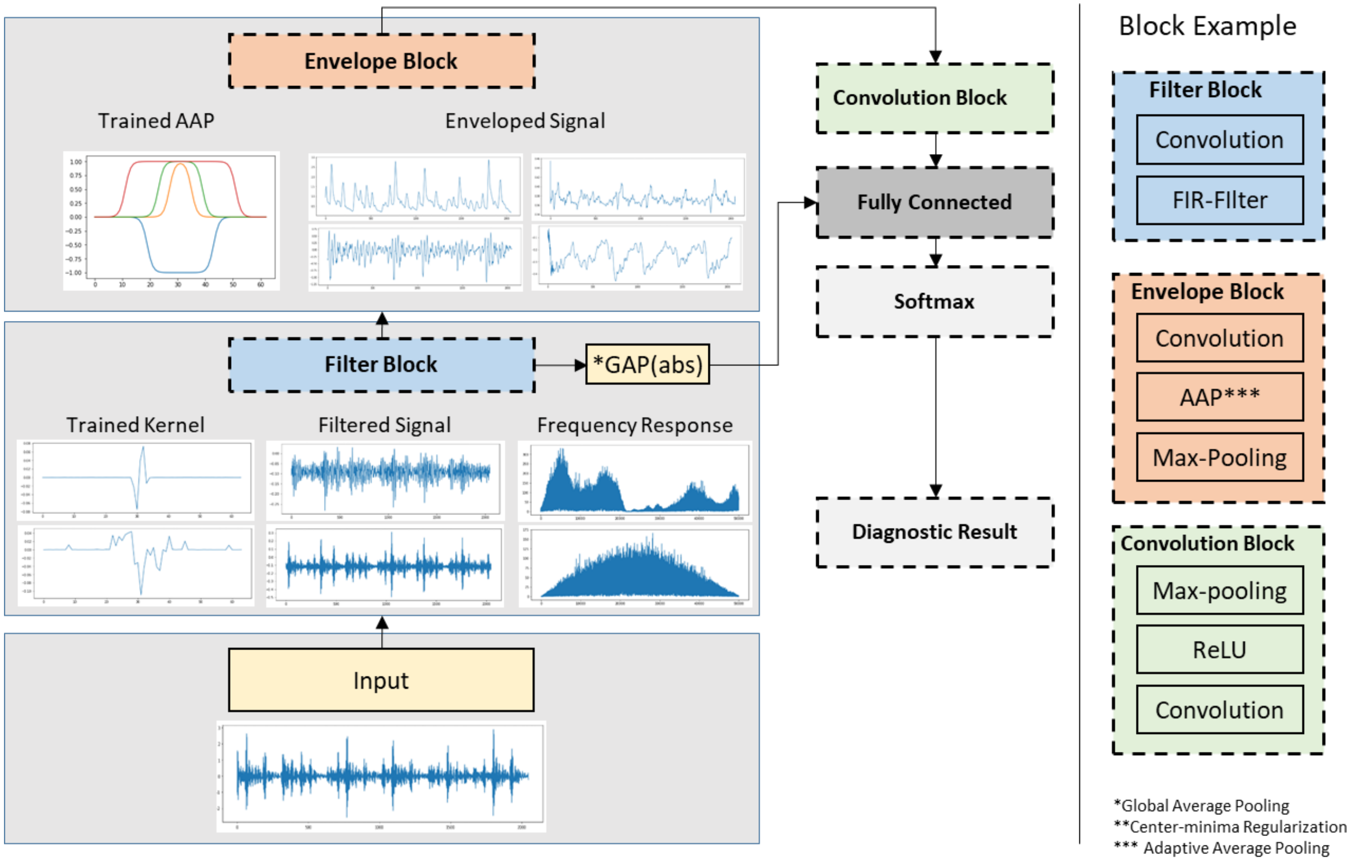


Figure 4-21 Overall Process of Filter-Envelope Network for Fault Diagnosis of Vibration Signal

4.4.2 Summary and Discussion

This chapter introduces studies to stabilize the diagnostic results and improve the diagnostic performance by learning generalized features from limited samples. The proposed two types of blocks can also function individually, and can be implemented as a feature extractor by being implemented at the front of the network regardless of MLP, CNN or RNN. The proposed method not only showed that high generalization performance can be achieved through strong structural regulation, but also suggests that it is important to design the network using knowledge of mechanical engineering and signal processing domain.

Chapter 5

Cepstrum Editing Based Data Augmentation for Vibration Signals

5.1 Brief Review of Data Augmentation for Deep Learning

This Chapter Covers a brief review of the data augmentation that motivates the proposed method, cepstrum editing based data augmentation (CEDA). First, we will cover examples in visual recognition field where data augmentation is well established, then introduced to other fields.

5.1.1 Image Augmentation to Enlarge Training Dataset

It is well known that machine learning technique requires more data to get better performance and operate efficiently. Even when the data is of low quality, the algorithm works better as long as usable data can be extracted from the original data. Google's Inception[48, 49] Network, for example, uses several data augmentation techniques to reduce the model's top-5 error rate by 2-3%. Collecting high quality data is difficult to obtain in large quantities because of the high cost of acquisition and labeling. Therefore, the method of collecting large amounts of unfiltered data using web crawling is mainly used, which is mostly unstructured data and it is known that structuring it is very difficult. Instead, a small amount of well-structured, well-labeled, high-quality data can be used. However, this is against the fact that machine learning requires a lot of data to get good performance. Therefore, many studies have been conducted to augment a small amount of high quality data[50-53].

Colored pictures, a representative visual recognition data, derive several augmentation methods from its nature. CNN achieved translation invariance compared to MLP. That is, CNN can achieve the same performance regardless of the orientation of the object to be recognized in the picture. However, even if the picture is taken with the same object, different data is generated according to the direction, distance, and lighting. So we can modify the existing photo to get more data as shown in Figure 5-1.

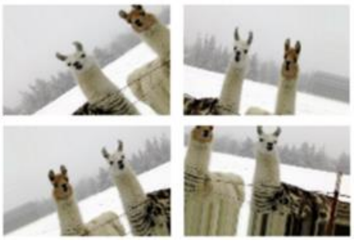


Figure 5-1 Examples of Data Augmentation in Images

Popular image augmentation methods include shifting, zooming in / out, rotate, flip, distort, or shaded with a hue. These images have essentially the same information as the existing images, but help the network learn more features from the data[54, 55].

5.1.2 Data Augmentation for Vibration Signal

The data augmentation of the vibration signal is limited compared to the image. This is because vibration signals are one-dimensional data of time, and false augmentation may harm its physical properties. An example of an outer race fault sample of rolling element bearing is shown in Figure 5-2.

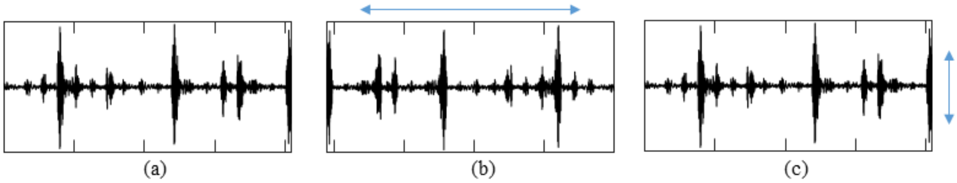


Figure 5-2 Affine Transforms to Augment Vibration Signal

Figure 5-2 (a) shows the original sample of the vibration signal, (b) shows the augmented examples of horizontal flipping and (c) shows vertical flipping. The outer race fault of the bearing takes the form of a sudden increase in the magnitude of vibration when the ball and the outer race dent collide, and then decrease due to damping of the system. Horizontal flipping, however, produces data in which vibrations increase slowly and then decrease rapidly. This does not properly reflect the physical characteristics of the vibration and can lead to incorrect learning of the network. In addition, in the case of vertical flipping, no significantly new sample is generated due to the symmetric property of the vibration signal about the center axis of the bearing. Therefore, it is necessary to consider a method that can augment more realistic data by reflecting the vibration signal of the data and the characteristics of the failure.

5.2 Cepstrum Editing based Data Augmentation

A cepstrum is considered as Fourier transformation of the logarithm of a spectrum. The name ‘Cepstrum’ is derived from reversed first four letters of word ‘spectrum’. The x-axis of the spectrum is frequency, and that of cepstrum is quefrency, which is derived from the word frequency from reversed first three letters from frequency. Given a real signal x(n), different cepstrum forms can be expressed as follow,

The real cepstrum of a signal x(n):

$$\mathrm{c(n)} = \frac{1}{2\pi}\int_{-\pi}^{\pi} \log X \left|e^{j\omega}\right| e^{j\omega n} d\omega \qquad (5\text{-}1)$$

The complex cepstrum of a signal x(n):

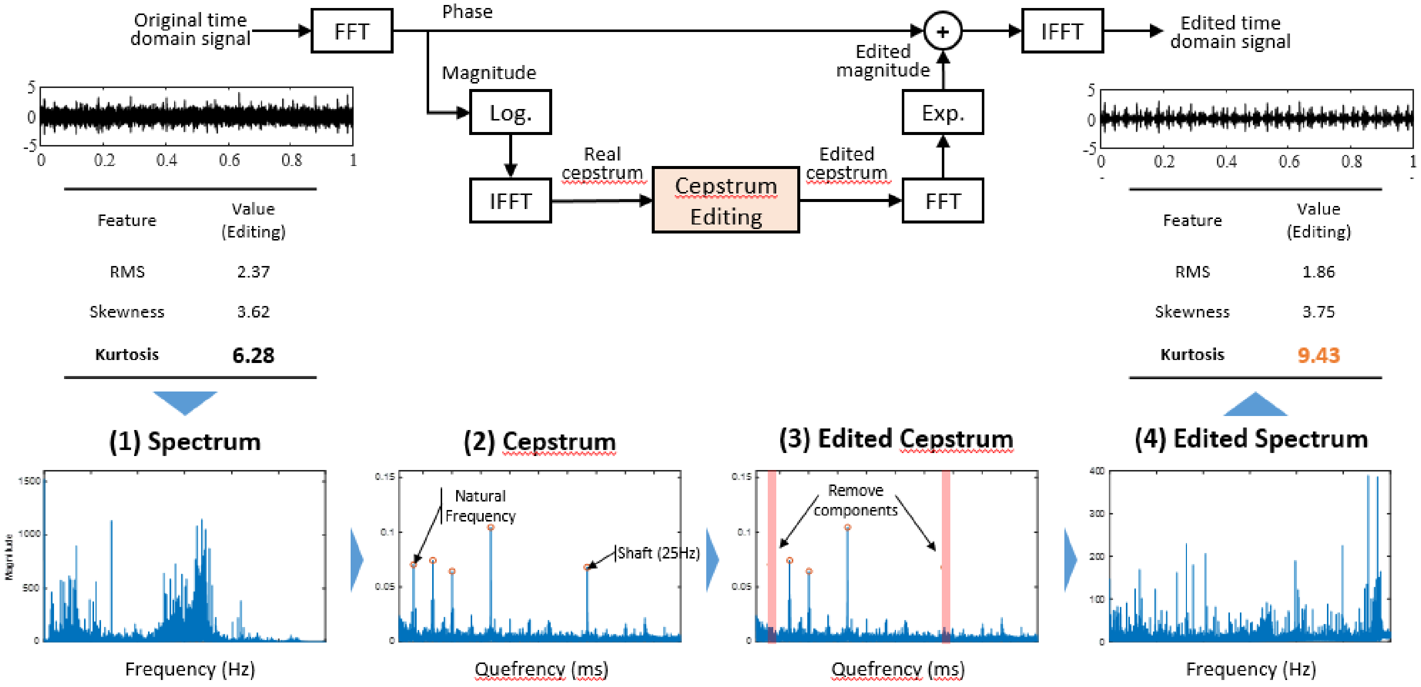
$$\mathrm{c(n)} = \frac{1}{2\pi}\int_{-\pi}^{\pi} \log X \, [(\omega)] e^{j\omega n} d\omega \qquad (5\text{-}2)$$

The power cepstrum of a signal x(n):

$$\mathrm{c(t)}^2 = \frac{1}{2\pi}\int_{-\pi}^{\pi} \log X \, [(\omega)] \left|e^{j\omega n} d\omega\right|^2 \qquad (5\text{-}3)$$

5.2.1 Cepstrum Editing as a Signal Preprocessing

The cepstrum is very useful when periodic spectral components (e.g.harmonics) need to be observed as concentrated information, into fewer peaks in ”rahmonics”. By eliminating these peaks, which is called ”liftering”, corresponding periodic spectral components are decreased in the signal. It was recently proposed and real cepstrum can be used to refine vibration signals. Recombining this edited amplitude spectrum with the original phase produces the edited time domain signal. Cepstrun editing can be implemented to separate deterministic signals from random signals by selecting and editing the peaks among the rahmonics. A general scheme of the CEP method is shown in Figure 5-3.



Feature	Value (Editing)
RMS	2.37
Skewness	3.62
Kurtosis	**6.28**

Feature	Value (Editing)
RMS	1.86
Skewness	3.75
Kurtosis	**9.43**

Figure 5-3 Process of Cepstrum Editing for Fault Diagnosis

5.2.2 Cepstrum Editing based Data Augmentation

The proposed approach in this chapter, cepstrum editing based data augmentation (CEDA) manipulate the original signal to direction of having more prominent features from fault. As we saw in the previous chapter, cepstrum allows us to distinguish between deterministic and random components from the original signal. This is shown in Figure 5-4.

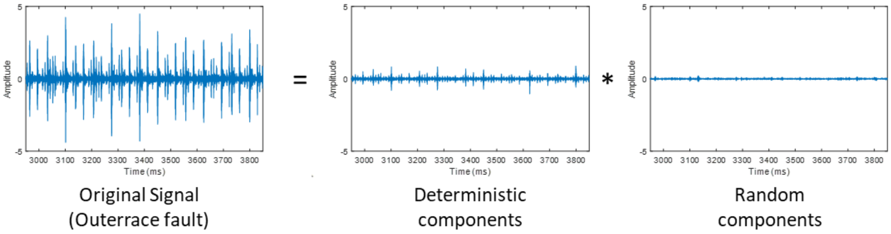


Figure 5-4 Separation of Discrete and Random Components using Cepstrum

The following describes the proposed CEDA process in order. First, Fourier transform the obtained signal. After dividing the result of the Fourier transform by magnitude and phase, the phase is stored in memory. Then take log in magintude. The reason for taking the log is that the low frequency components of the original signal are so large that they are offset by taking the log and finding the best discrete component. And if you take an invert Fourier transform again, the real part of the result is the cepstrum. To edit this cepstrum, first find the peak from the cepstrum. The number of peaks to be found here depends on the data and can be defined by the parameters of the algorithm. These cepstrum peaks contain information about the demodulation frequency due to the fundamental nature or failure of the system.

However, lack of data makes it difficult for the network to learn that this information is about failure. Thus, some perturbation of this information allows the network to focus more on the features exhibited by the failure. At this time, there are various ways to give perturbation, mainly by multiplying the comb filter composed of random variables of a specific distribution.

One comb filter corresponds to one new augment sample. It should be noted that One vibration sample is increased by N times when augmented by N comb filters. The cepstrum modified by the comb filter is returned to the frequency domain through the Fourier transform and exponential functions, and the augmentation of a sample is completed by taking the inverse Fourier transform in combination with the stored phase information.

5.3 Results and Discussion

5.3.1 Performance validation to rolling element bearing diagnosis

First, cepstrum is performed for four modes of rolling element bearing (normal, ball fault, inner race fault, outer race fault), and the result of removing all peaks found in cepstrum is shown in Figure 5-5.

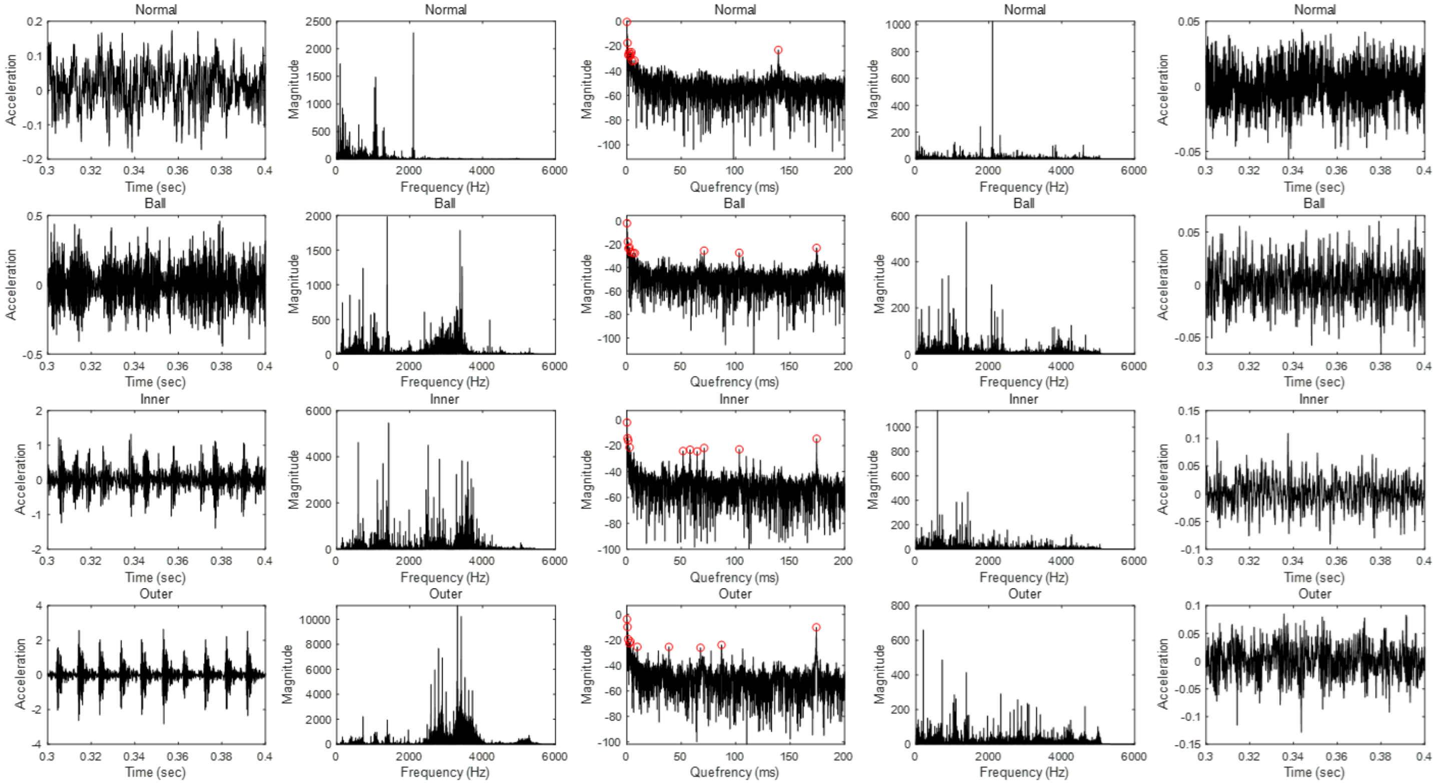


Figure 5-5 Data Augmentation Process using Cepstrum Editing for Rolling Element Bearing

First, look at the spectrum of the normal signal, where most of the components are below 2000 Hz, and the cepstrum is also concentrated in a very short quefrency. This is because there is hardly any harmonic or modulation due to a failure. In contrast, the spectrums of the fault signals all show resonance frequencies in the 2000 – 4000 Hz range. Therefore, in the cepstrum, a peak is found at a quefrency of 50-200 ms. These peaks are likely to contain fault signals such as modulation. In order to see how the information contained in the Cepstrum peak appears in the time domain, if all the peaks are removed and the signal is augmented, it can be seen that all three faults have disappeared. An example of an augment signal regenerated with a random comb filter is shown in Figure 5-6.

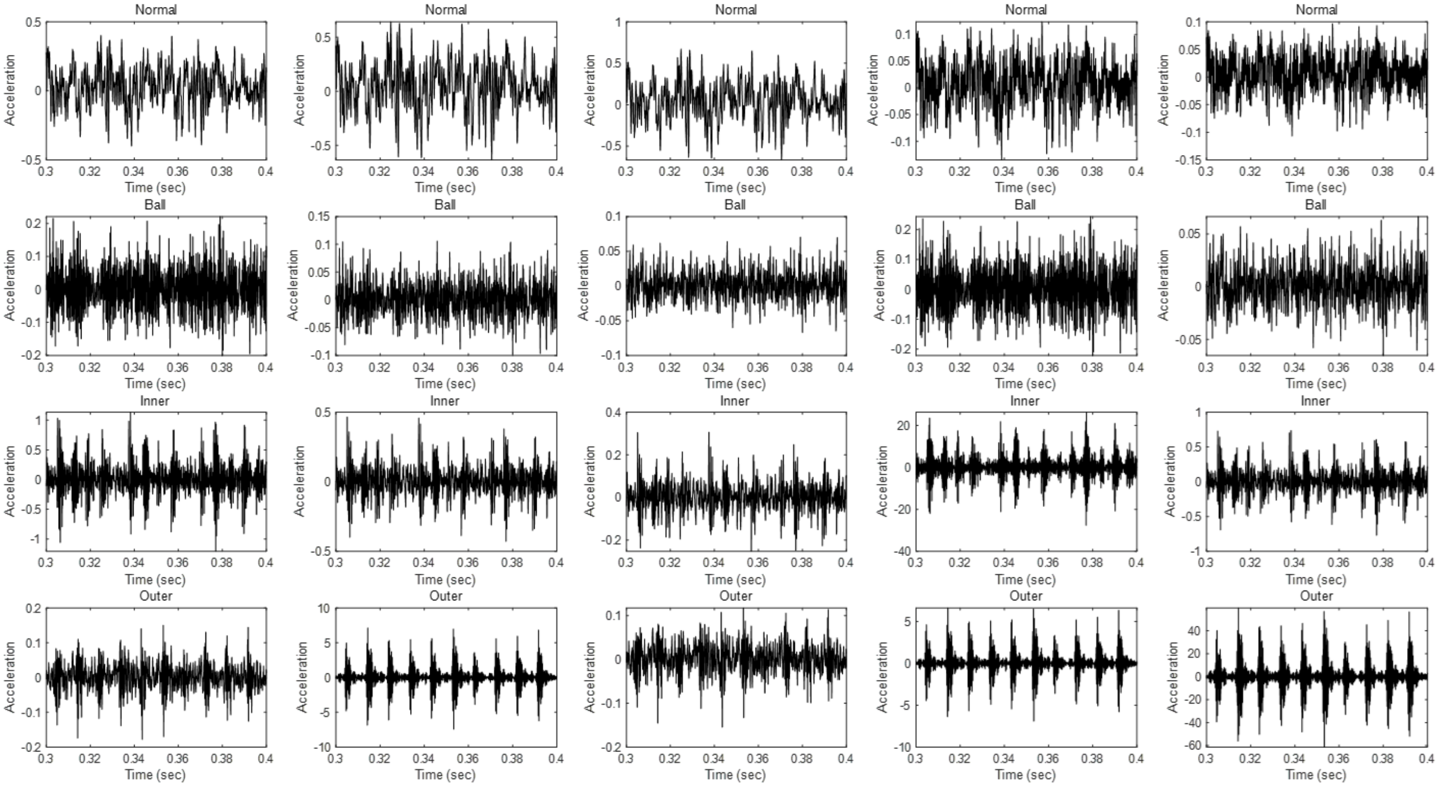


Figure 5-6 Examples of Augmented Samples from Cepstrum Editing

By the inspection of the augmented samples, it can be seen that in the case of a normal signal, the augmented sample exhibits similar behavior as the original signal. This is because there is almost no change due to peak editing because there are few cepstrum peaks due to fault modulation at the top. Alternatively, changes are found in case of faulty samples. It can be seen that the magnitude of the modulated signal, which appears repeatedly in all three types of faults, increases or decreases in size and changes in shape. The accuracy results for these and no augmentations are shown in Figure 5-7

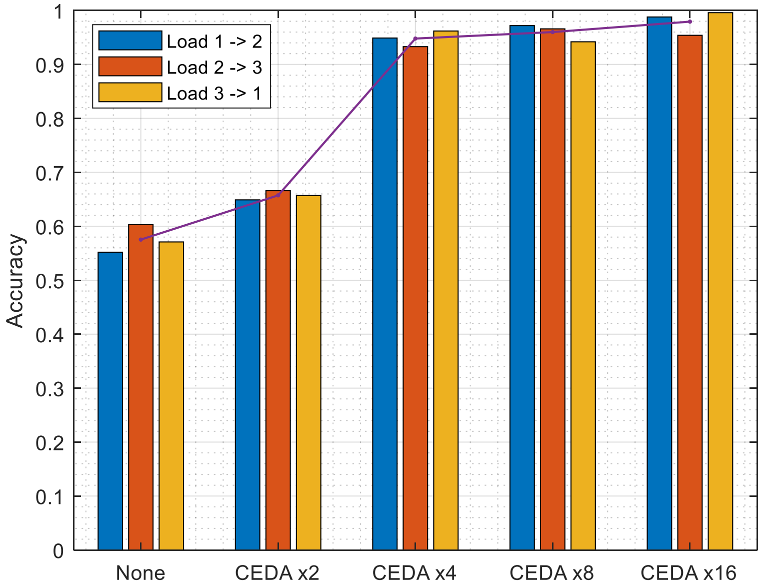


Figure 5-7 Accuracy of CEDA in Rolling Element Vibration Signals

The experiment acquires the signal generated when the rolling element bearing is operated for three levels of operating load conditions, and then analyzes the accuracy of diagnosing the failure with the signal acquired under another load condition using the network learned under one load condition. It proceeded by measuring. This experiment simulates a situation where only fault signals of limited environment can be obtained in the industrial field. Looking at the results, it can be seen that the accuracy increases as the number of samples increased by CEDA increases.

Chapter 6

Selective Parameter Freezing for Parameter Transfer with Small Dataset

The parameter transfer approach freezes the shallow layer near the input layer of the source network and fine-tunes the deep layer near the output layer to the target data. As the source network is larger and more complex than the size of the target dataset, it focuses on freezing more layers and mitigating overfitting and maintaining the generalization ability of the source network. The larger the target dataset size, it is the more likely to freeze the only shallow layers and attempt to increase the adaptability to the target domain. Therefore, parameter transfer is accompanied by several trials and errors to find the depth of the frozen layer which can achieve the best performance[13].

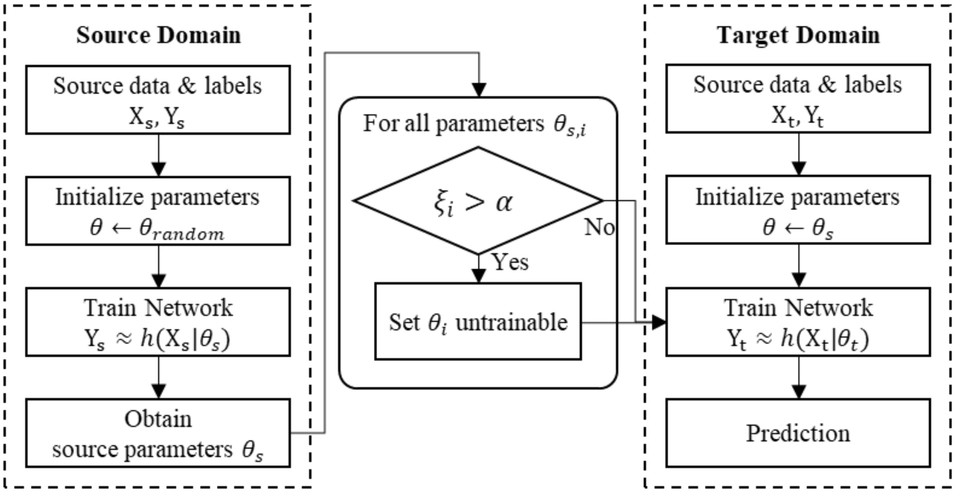
The limitation of this freezing and fine-tuning is to exclude the opportunity to learn the general features from the target data by freezing all the parameters of the shallow. The use of direct source features through freezing is accompanied by the risk of performance degradation as the distribution of source and target data differs. This problem occurs from the fact that parameter transfer approach does not offer any option for adjusting the freezing and fine-tuning inside a layer.

In this paper, we propose selective parameter freezing (SPF), an alternative approach to the two conventional parameter repurposing methods, parameter freezing and fine-tuning. SPF is devised to find a compromise for these methods that mitigate overfitting and, at the same time, have adaptability to target domains. The proposed SPF method allows retraining of only unnecessary parameters to the target data, while important source features remain. For SPF, we use parameter sensitivity as the output for determining available parameters. As a result, SPF can obtain better performance, as compared to conventional approaches.

The remainder of this paper is organized as follows. Section 6.1 provides background knowledge for neural networks and transfer learning. In Section 6.2, the proposed SPF method is introduced, and details are described. In Section 6.3, the experimental setup for the validation of the proposed approach is described. Section 6.4 and 6.5 contains the results of two case studies for bearing diagnosis. Section 6.6 summarizes and concludes this paper.

6.1 Overall Procedure of Selective Parameter Freezing

Figure 6-1 describes the overall procedure for SPF-based transfer learning. First, the network is trained to the source task $(\mathrm{X_s}, Y_s)$ and the source parameters θ_s= $(\theta_s^{(i)}; i = 1,2, \dots, N_p, N_p$ is number of parameters) are obtained. Some of the source parameters are highly sensitive with respect to the result of the network, which means contain features useful to the source task. In CNN case, the average sensitivity within a kernel determines whether the kernel should be frozen or not. The sensitivity of parameter θ_i is denoted as ξ_i. Here, we introduce a hyperparameter α which is a threshold value for the parameter sensitivity ξ_i. If the ξ_i. Is higher than α, the parameter θ_i is set to be untrainable in the training. Third, the target task $(\mathrm{X_t}, Y_t)$ is trained using only trainable parameters that have low sensitivity to the source task; this is so that they can provide more adaptability to the target task by allowing learn new features from the target task.



Source Domain
Source data & labels
$\mathrm{X}_s, \mathrm{Y_s}$
Initialize parameters
$\theta \leftarrow \theta_{random}$
Train Network
$\mathrm{Y_s} \approx h(\mathrm{X_s}|\theta_s)$
Obtain
source parameters θ_s
For all parameters $\theta_{s,i}$
$\xi_i > \alpha$
No
Yes
Set θ_i untrainable
Target Domain
Source data & labels
$\mathrm{X_t}, \mathrm{Y_t}$
Initialize parameters
$\theta \leftarrow \theta_s$
Train Network
$\mathrm{Y_t} \approx h(\mathrm{X_t}|\theta_t)$
Prediction

Figure 6-1 The overall procedure for SPF-based transfer learning

In order to train a network using SPF, different gradients must be delivered to the gradient descent update rule, based on their sensitivity. The gradient descent update rule is written as follows:

$$\theta_i := \theta_i - \eta \frac{\partial \mathcal{L}}{\partial \theta_i} \qquad (6\text{-}1)$$

where η is the learning rate.

Here, we introduce SPF for the update rule:

$$\frac{\partial \mathcal{L}}{\partial \theta_i} := \begin{cases} \frac{\partial \mathcal{L}}{\partial \theta_i}, & \xi_i \leq \alpha \quad (Fine\ tuning) \\ 0, & \xi_i > \alpha \quad (Freezing) \end{cases} \tag{6-2}$$

If ξ_i corresponding to θ_i is higher than the threshold α, the gradient is set to zero and θ_i is not updated to be frozen. If α is given to be larger than the highest sensitivity among all parameters, the training works the same as in the fine-tuning method. If α is 0 or smaller than 0, all parameters are frozen and not updated. In other words, the threshold α changes the balance between freezing and fine-tuning.

Table 6-1 Details of CNN model used in experiments

Layer	Type	Kernel size	Stride	Channel	Parameter	Repurposing
Input	Window	1024	-	-	-	-
Conv1	Convolutional	64	16	16	1,024	Freezing/ Fine-tuning/ SPF
	Max pooling	2	2	16		
Conv2	Convolutional	3	1	32	1,536	Freezing/ Fine-tuning/ SPF
	Max pooling	2	2	32		
Conv3	Convolutional	3	1	32	6,144	Freezing/ Fine-tuning/ SPF
	Max pooling	2	2	32		
Conv4	Convolutional	3	1	64	12,288	Fine-tuning
	Max pooling	2	2	64		
Conv5	Convolutional	3	1	64	12,288	Fine-tuning
	Max pooling	2	2	64		
FC	Fully connected	100	-	1	6400	Fine-tuning
	Classification	No. class	-	1	1000	Initialize

6.2 Determination Sensitivity of Source Network Parameters

Research for defining features from trained network parameters has been attempted; approaches proposed to date include active hidden unit screening[56], and visual inspection depending on domain knowledge, among others. These methods are limited because they do not include direct quantification of the influence of the parameters to the task. The feature should be defined by their level of impact on the performance of the task. Thus, we determined whether the parameter is a feature or not by using the derivative of the output with respect to the parameter; we call this the parameter sensitivity ξ. The parameter sensitivity implies that when a parameter is changed, the larger the value of the output change, the more sensitive the parameter. In our study, the parameter sensitivity ξ for parameter θ_i is expressed as (6-3).

$$\xi_i = \sqrt{\frac{1}{N_y}\sum_k \left(\frac{\partial \hat{y}_k}{\partial \theta_i}\right)^2} \qquad (6\text{-}3)$$

Equation (10) is a root mean square of the derivative of the $\hat{y}_k$ with respect to the parameter θ_i. An arbitrary $\hat{y}_k(k = 1, \ldots, N_y)$ with respect to the weight w_{ji} and bias b_j of the final layer can be expressed as (6-4) and (6-5):

$$\hat{y}_k = \mathrm{softmax}(\mathrm{z_k}) = \frac{e^{z_k}}{\sum_k e^{z_k}} \qquad (6\text{-}4)$$

$$\mathrm{z_j} = w_{ji}x_i + b_j \qquad (6\text{-}5)$$

where, i and j is an index of the column and row of weight matrix, respectively.

By using a chain rule, the derivatives of $\hat{y}_k$ with respect to the w_{ji} and b_j are expressed as:

$$\frac{\partial \hat{y}_k}{\partial w_{ji}} = \frac{\partial \hat{y}_k}{\partial z_j} \cdot \frac{\partial z_j}{\partial w_{ji}} \qquad (6\text{-}6)$$

$$\frac{\partial \hat{y}_k}{\partial b_j} = \frac{\partial \hat{y}_k}{\partial z_j} \cdot \frac{\partial z_j}{\partial b_j} \qquad (6\text{-}7)$$

Using the quotient rule of derivatives, the derivative of (6-4) is:

$$\frac{\partial \hat{y}_k}{\partial z_j} = \hat{y}_j\left(\delta_{kj} - \hat{y}_k\right) \qquad (6\text{-}8)$$

where δ_{kj} is a Kronecker delta.

Also, the derivatives of the z_j to the parameters are:

$$\frac{\partial z_j}{\partial w_{ji}} = x_i \qquad (6\text{-}9)$$

$$\frac{\partial z_j}{\partial b_j} = 1 \qquad (6\text{-}10)$$

Thus, the sensitivities with respect to weight and bias for the final layer are, respectively:

$$\xi_{w_{ji}} = \frac{\partial \hat{y}_k}{\partial w_{ji}} = \sqrt{\frac{1}{N_y} \sum_k \left(\hat{y}_j\left(\delta_{kj} - \hat{y}_k\right) \cdot x_i\right)^2} \qquad (6\text{-}11)$$

$$\xi_{b_j} = \frac{\partial \hat{y}_k}{\partial b_j} = \sqrt{\frac{1}{N_y} \sum_k \left(\hat{y}_j\left(\delta_{kj} - \hat{y}_k\right) \right)^2} \qquad (6\text{-}12)$$

6.3 Case Study 1: Transfer to Different Fault Size

In order to compare the performance of the proposed method and the conventional approaches, parameter transfer was performed using a case western reserve university bearing dataset. Input size N is set to 1,024, and the first layer of the network is designed to have a wide kernel as proposed in Zhang, et al. [57]. The network has five convolutional-ReLU-Batch-Norm-Max Pooling Blocks and two Fully-connected layers with a total of 41,510 parameters. 1-3th convolution layers of the network are frozen/fine-tuned or selectively frozen. Details are in Table 6-1 . In addition, three transfer tasks were used to derive SPF performance. Each set is set so that the source data is sufficient for one type of fault size and the target data is given only a small amount for the other two fault sizes as described in Table 6-2.

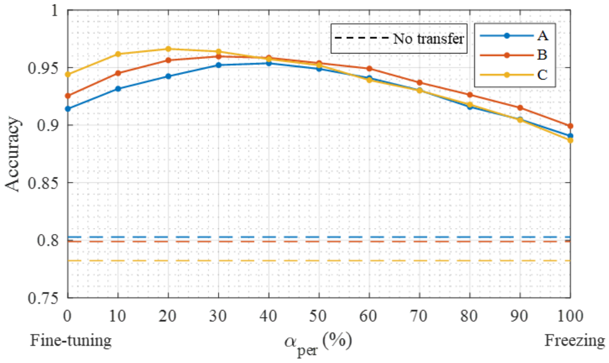


Figure 6-2 Performances for various transfer conditions by the change of hyperparameter α

6.3.1 Performance by hyperparameter α

The key feature of the SPF is an adaptation to the small number of target data by introducing a hyperparameter α to change the balance between freezing and fine-tuning within a layer. Therefore, we can achieve both advantages of fine-tuning and freezing by adjusting the α appropriately. For the transfer tasks, the result of the accuracy evaluation by varying the α is shown in Figure 6-2.

Figure 6-2, α_{per} is percentile α, which is the α corresponding to the upper α_{per} % of the overall parameter sensitivity. When α_{per}=0, parameters belonging to all SPF layers become trainable regardless of its sensitivity. Likewise, when α_{per} =100, all layers are frozen.

First, when training the target data from scratch, the accuracy of the transfer set was only about 80%. And the accuracy from the parameter transfer using fine-tuning was more than 11 ~ 16% higher than that. On the other hand, the freezing showed 8 ~ 10% accuracy improvement. In other words, the accuracy of the fine-tuning method for transfer sets A, B, and C was 2.36%, 2.62%, and 5.73% higher than freezing, respectively. By applying the SPF method, we can freeze only a portion of parameters within a layer by changing α and fine-tune only the rest. In this experiment, the parameters were further frozen at 10% intervals. The highest performance which is 3.96% higher than that of fine-tuning was obtained at α_{per}=40 in the transfer set A. To transfer set B and C, accuracies were also improved by 3.42% and 2.22%, respectively.

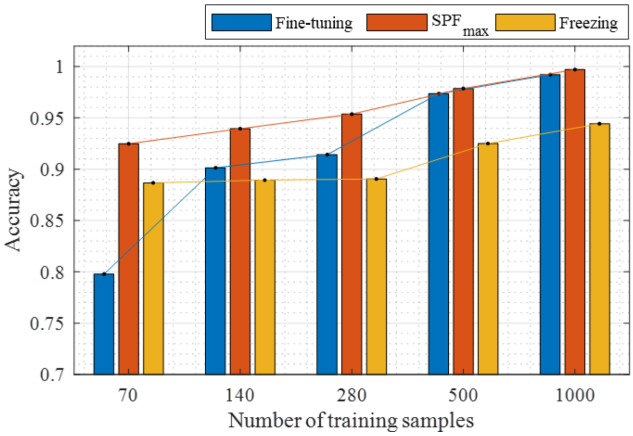


Figure 6-3 Performances of three parameter repurposing methods by size of dataset for transfer set A

6.3.2 Effect of the number of training samples and network size

We used 70, 140, 280, 500 and 1000 training samples in the same network to determine the relationship between the size of the dataset and the performance enhancement in the SPF. Figure 6-3 is the diagnostic accuracy according to the amount of training sample derived for transfer A. When the number of samples was very small, fine-tuning the network resulted in poor accuracy. However, as the number of samples increased, the fine-tuning method gave higher accuracy. In the case of freezing, the accuracy was about 9% higher than the fine-tuning method with 70 samples. However, the fine-tuning was superior given sufficient samples were available. InFigure 6-3, the performance of the SPF is plotted with the highest accuracy from the αper of 10-90 with 10 intervals. The SPF has better performance than conventional methods with all tested sample numbers. In particular, the improvement in accuracy was higher with less samples.

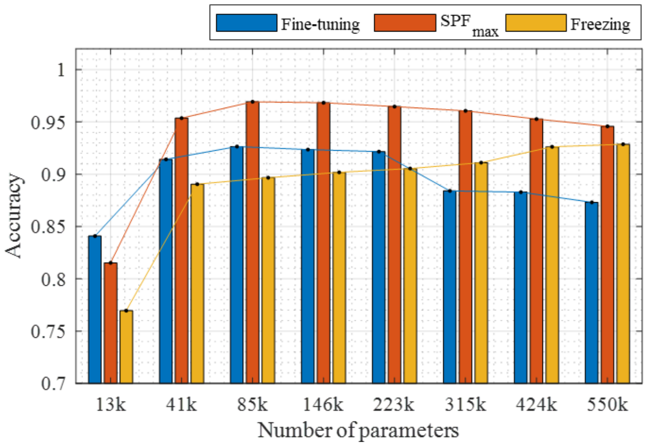


Figure 6-4 Performances of three parameter repurposing methods by size of network for transfer set A

Table 6-3 and Table 6-4 summarize absolute performances and SPF performance improvements (compared to the higher performance between fine-tuning and freezing) for transfer sets A, B and C, respectively. A notable increase in the performance of the SPF method was observed as the number of samples less than 500. SPF always gives a performance similar to that of a better method of fine-tuning and freezing, because accuracy depending on α occur continuously between the performance of fine-tuning and freezing.

Next, we examined the performance of SPF the parameter repurposing methods including the proposed depending on the size of the network. The number of samples in the experiment was set at 280, and the number of kernels belonging to the convolutional layers were multiplied by × 0.5, × 1, × 1.5, ..., ×4 to change the size of the network. A network with four times the number of kernels have 14 times more parameters to the original network.

As can be seen in Figure 6-4, when the size of the network is very small (about 13,000), the overall diagnostic performance is greatly decreased. SPF also showed low performance and fine-tuning showed the highest performance. It should be noted that as the size of the network increases, the performance of freezing increases, while the performance of

fine-tuning decreases. This is because the number of trainable parameters is very large when the network size is large, so the fine-tuning method quickly reduces the training loss but not with the validation error. Freezing, on the other hand, has increased accuracy as the network size larger. This seems to be due to sufficient data and well-considered regularization to ensure good test accuracy to source data (generally close to 100%) when increasing the size of the source network. SPF performed better than other methods for all network size settings except for the smallest network. In the largest network, the performance gap to the freezing method is the smallest, when SPF has αper of 90, which has 90% of frozen parameters.

6.4 Case Study 2: Transfer from Artificial to Natural Fault

Industrial fields are suffering from a small amount of data to develop a fault diagnosis model. Industries are extremely sensitive to equipment faults or failures and have a conservative management system to prevent failures in advance. Nevertheless, the development of fault diagnosis models is important for the construction of cost-effective management systems. Obtaining fault data from equipment and applications in the field is difficult therefore it is challenging to develop diagnostic models. Transfer learning using parameter transfer can be an appropriate solution to this problem.

In this case study, we will develop a model to diagnose natural faults from the accelerated life test of rolling element bearings using parameter transfer of network trained with artificial fault data. Table 6-7 describes the details of the transfer task. The source network was trained for 16 categories of data, normal and three types of artificial faults obtained at four rotational speeds. The target data consists of normal and four kinds of fault, inner race spalling and flaking depending on the severity of failure, and relatively small number of samples are provided for training. All training samples were augmented with a window size of 2,048.

6.4.1 Diagnostic performance for proposed method

Table 12 shows the accuracy of various diagnostic methods for comparison with the proposed SPF method. For comparison, 1) SVM, MLP, CNN trained from scratch without transfer learning, 2) CNN with frozen or fine-tuned layers were used. The SVM was trained with 1,025 FFT coefficients of a sample as an input. CNN has been tested with several levels of freezing and fine-tuning, and its subscripts indicate how many shallow layers are frozen. For example, CNN_0 means fine-tuning all layers, CNN_3 means that three shallow layers are frozen.

As a result, the methods trained from scratch have lower performance than the parameter transfer methods. CNN, however, showed up to 97% accuracy when the number of samples was sufficient. Parameter transfer methods have different accuracy depending on degree of freezing and CNN_3 method has showed the highest accuracy with smaller dataset. And when the training samples were sufficient, the best results were obtained when the first layer was frozen or all fine-tuned, and the best accuracy was 99.8% in the fine-tuning method with 2000 samples. The proposed method, SPF, showed generally high accuracy. In particular, when the number of samples is 100 or 200, the diagnostic performance is about 4-5% higher than the freezing and fine-tuning method. However, when 1000 or 2000 samples were used for training, the performance was 1-2% lower. The result of this case study from artificial faults to natural faults shows that the parameter transfer is useful to develop diagnostic models with small natural fault samples and SPF can achieve higher performance for this problem.

6.4.2 Visualization of frozen parameters by hyperparameter α

Figure 6-5 (a) shows the map of parameter sensitivity of the source network trained for various artificial faults data. Each box represents the weight matrix of the convolutional layers, and since the second layer is a three-dimensional matrix, it represents the average in the input depth direction. Therefore, the horizontal direction is the kernel width, and the vertical direction is the number of the kernel. The first thing to look at is that not all parameters are related to the output, even though the source network has been trained for a lot of data. Therefore, the parameters expressed in blue in Figure 6-5 (a) do not affect the performance of the source task even if their values change.

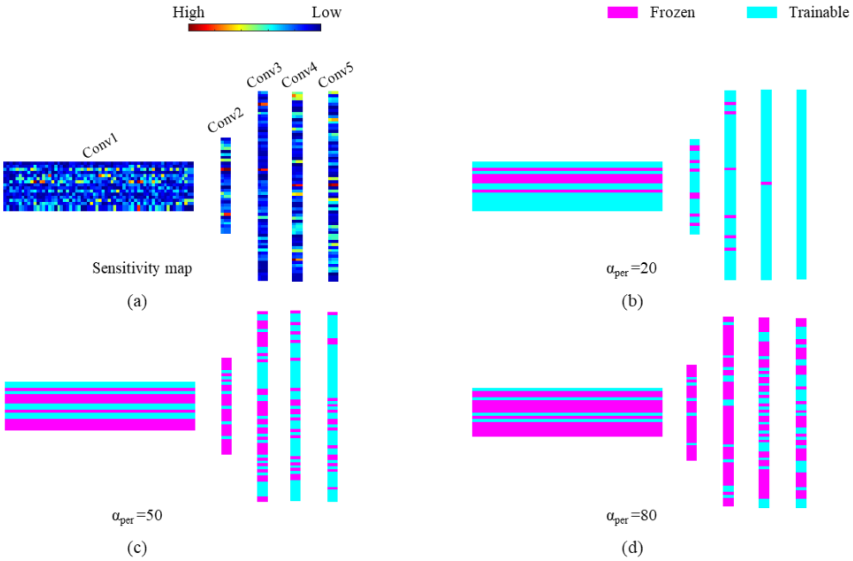
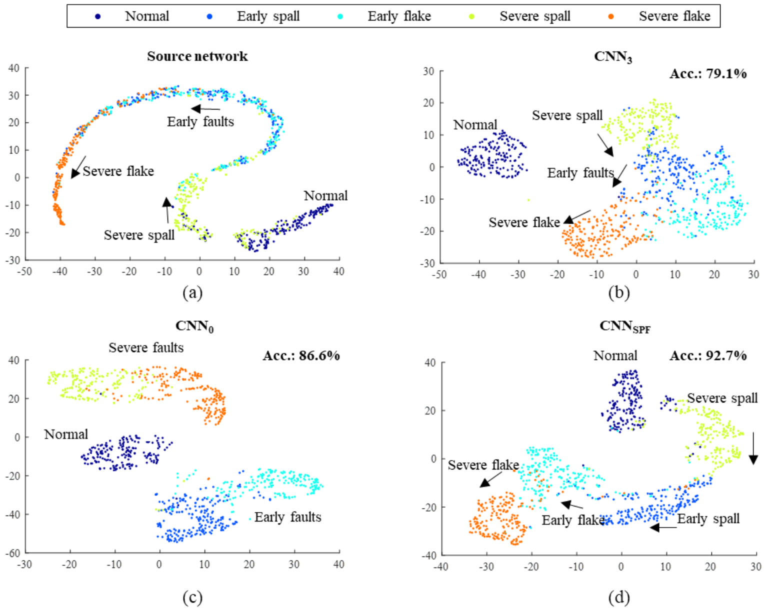


Figure 6-5 Parameter sensitivity and corresponding selective parameter freezing by various value of α

The SPF method allows only these parameters to be trainable. Figure 6-5 (b)-(d) show the freezing state of the parameters according to α. As mentioned in Section III, SPF freezes a kernel if the average sensitivity of the kernel is greater than a given α. The pink horizontal lines in Figure 6-5 (b)-(d) represent that the kernels are frozen. In this figure, it can be assumed that while the frozen shallow layers are not able to learn the general features in the target domain, SPF allows the general features at the to be learned while also reducing the number of trainable parameters.



(a) (b)

(c) (d)

Figure 6-6 Feature space of networks using t-SNE with respect to the natural fault data

6.4.3 Visual inspection of feature space

To better understand the effect of SPF, we looked at the feature space of the last fully connected layer in 2D space using t-SNE. In Figure 6-6 (a), features from the target data through the source network are plotted before training for target data. Interestingly, the features from the natural fault samples are weakly clustered even though the network is not yet trained for the natural fault domain. For example, normal and fault group are clearly classified, and in the fault group, the samples are observed in the order of severe spall - early spalling and flaking - and severe flaking (the arrow direction in the plot). This feature space shows that the source features learned for artificial faults can already classify normal and faulty classes and express some difference between fault types. Also, the region with mixed early spalling and flaking samples implies that they are represented by similar features. Figure 6-6 (b) is the feature space of the frozen all convolutional layer CNN_5 trained for 100 samples. Here, we can identify faulty sample groups that are similar to Figure 6-6 (a), but more distinctive. This indicates that the frozen network performs classification utilizing the transferred features. Figure 6-6 (c) represents the feature space of CNN_3. Unlike the case of Figure 6-6 (b), early faults and severe faults form separate groups to each other. This implies that more feature space transformations occurred than CNN_5.

Finally, the feature space of $\mathrm{CNN}_{\mathrm{SPF}}$ is plotted in Figure 6-6 (d). In the feature space of $\mathrm{CNN}_{\mathrm{SPF}}$, each class has a distinctive group, but it is located in the same order as the fault clusters observed in the feature space of source network. In this example, the SPF trained the four fault types in classifiable clusters, taking the feature space similar to the that of source network. In that $\mathrm{CNN}_{\mathrm{SPF}}$, α_{per} of the SPF was 80, which means only 20% of the low sensitive parameters were trained to the target domain. In CNN_3, on the other hand, the number of trainable parameters was about 60% of the total number of parameters, even though three shallow layers were frozen. SPF can provide both domain adaptability of fine tuning and overfitting mitigation of freezing method when it is possible to acquire sufficiently good features in the source domain such as from artificial to natural fault diagnosis problem.

6.5 Conclusion

In this paper, we propose a selective parameter freezing-based parameter transfer approach to perform vibration-based fault diagnosis in extreme situations, even when only a small number of training samples are available, for example, natural fault data from industrial fields. The SPF approach provides an intermediate option between freezing and fine-tuning to take advantage from both approaches. The performance of the proposed approach was validated using two datasets. The proposed method achieves a maximum diagnostic accuracy when the training samples are insufficient regardless of the network size. We also demonstrate that the parameter transfer from artificial fault diagnostic model to natural faults is effective and the proposed method is available to improve the performance more. We looked into the feature space to investigate how the approach improves the performance of parameter transfer. Through the inspection, we made two discoveries. First, the feature space the CNN trained with only artificial faults are able to extract features to classify normal and faulty state of the natural fault data. Second, the proposed method maintains the feature space of the source network and improves the accuracy by adapting to the domain of the natural fault. In the proposed method, the network is able to learn new features from target domain with reduced network capacity, and mitigates overfitting and enhance the domain adaptability. Through this study, we showed that the configuration of parameter repurposing is influential, and the proposed method provide a new option for the optimization of the parameter transfer approach.

Table 6-2 Details of source and target data for the experiments in different fault size

Transfer task	Domain	No. class	Type	Size (mm)	Load (hp)	No. sample
A	Source	4	Normal, BF, IRF, ORF	0.18	1,2,3	5000
	Target	7		0.36, 0.54		280
B	Source	4		0.36		5000
	Target	7		0.18, 0.54		280
C	Source	4		0.54		5000
	Target	7		0.18, 0.36		280

Table 6-3 Result of three parameter repurposing methods by size of dataset for transfer set A, B and C

Transfer task	Method	Number of training samples				
		70	140	280	500	1000
A	Fine-tuning	79.79%	90.14%	91.42%	97.36%	99.22%
	SPF$_{max}$	**92.48%**	**93.95%**	**95.37%**	**97.86%**	**99.71%**
	Freezing	88.67%	88.94%	89.05%	92.50%	94.43%
B	Fine-tuning	84.04%	88.55%	92.54%	96.97%	98.07%
	SPF$_{max}$	**94.24%**	**94.73%**	**95.96%**	**98.05%**	**98.36%**
	Freezing	88.70%	89.60%	89.92%	93.90%	94.48%
C	Fine-tuning	81.15%	92.95%	94.41%	99.21%	99.51%
	SPF$_{max}$	**91.21%**	**96.34%**	**96.62%**	**99.80%**	**99.80%**
	Freezing	88.06%	88.55%	88.68%	93.07%	93.14%

Table 6-4 Performance improvement by size of dataset for SPF compared to the conventional methods

Transfer task	Number of training samples				
	70	140	280	500	1000
A	3.80%	3.80%	3.94%	0.50%	0.49%
B	5.53%	5.13%	3.42%	1.07%	0.29%
C	3.14%	3.39%	2.22%	0.60%	0.29%

Table 6-5 Result of three parameter repurposing methods by size of network for transfer set A, B and C

Transfer set	Method	Number of parameters							
		13k	41k	85k	146k	223k	315k	424k	550k
A	Fine-tuning	**84.1%**	91.4%	92.6%	92.4%	92.2%	88.4%	88.3%	87.3%
	SPF$_{max}$	81.5%	**95.4%**	**96.9%**	**96.8%**	**96.5%**	**96.1%**	**95.3%**	**94.6%**
	Freezing	77.0%	89.0%	89.7%	90.2%	90.5%	91.1%	92.6%	92.9%
B	Fine-tuning	**91.91%**	92.54%	93.19%	93.46%	95.02%	95.29%	94.90%	93.82%
	SPF$_{max}$	90.58%	**95.67%**	**95.90%**	**97.00%**	**97.11%**	**96.78%**	**96.29%**	**96.09%**
	Freezing	85.69%	89.92%	90.76%	92.19%	93.69%	93.72%	93.43%	93.25%
C	Fine-tuning	92.09%	94.41%	96.54%	97.12%	97.80%	97.41%	96.93%	96.72%
	SPF$_{max}$	**92.38%**	**96.62%**	**98.84%**	**99.83%**	**99.32%**	**99.12%**	**97.93%**	**98.41%**
	Freezing	87.79%	88.68%	88.60%	93.34%	94.85%	94.87%	95.48%	96.02%

Table 6-6 Performance improvement by size of network

Transfer set	Size of Network							
	13k	41k	85k	146k	223k	315k	424k	550k
A	-2.56%	3.94%	4.28%	4.48%	4.30%	5.00%	2.65%	1.71%
B	-1.33%	3.13%	2.71%	3.54%	2.09%	1.49%	1.39%	2.27%
C	0.29%	2.21%	2.30%	2.71%	1.51%	1.71%	1.00%	1.70%

Table 6-7 Details of data for parameter transfer from artificial fault to natural fault

Domain	No. class	Type	Load (hp)	No. sample
Source	16	(Normal, BF, IRF, ORF)×4 speeds	0.34	8000
Target	5	Normal, early spalling, early flaking, severe spalling, severe flaking	0.329 ~ 0.344	100 - 2000

Table 6-8 Results diagnostic performance for various diagnostic approaches

No. sample	No parameter transfer			Parameter transfer with freezing and fine-tuning						SPF
	SVM	MLP	CNN	CNN$_0$	CNN$_1$	CNN$_2$	CNN$_3$	CNN$_4$	CNN$_5$	CNN$_{SPF}$
100	50.2%	63.3%	61.9%	71.5%	70.3%	76.4%	86.6%	82.8%	79.1%	**92.7%**
200	52.6%	66.5%	67.2%	80.1%	82.3%	86.9%	88.6%	84.6%	80.3%	**93.3%**
400	62.0%	75.8%	85.0%	89.5%	90.4%	91.6%	92.1%	86.5%	81.1%	**94.9%**
1000	77.4%	86.5%	92.8%	95.6%	**97.7%**	96.0%	92.7%	90.3%	85.2%	97.4%
2000	80.7%	88.2%	97.0%	**99.8%**	99.6%	98.3%	97.5%	92.9%	86.9%	98.2%

Chapter 7

Conclusions

7.1 Contributions and Significance

This thesis aims to achieve higher and robust performance for fault diagnosis of rotating machinery with insufficient data. This thesis contains three research thrusts. (1) learning developing a robust and efficient architecture of neural network that is specialized to vibration signal based fault diagnosis. (2) Augment dataset of vibration signals for deep learning training using cepstrum editing, and (3) Improving performance for small dataset using novel parameter transfer method in transfer. The three thrusts of the research are expected to offer the following contributions.

Contribution 1: Robust Feature Learning from Inherent Preprocessing in Neural Network

The filter-envelope network which is proposed in this thesis stabilizes the diagnostic results and improve the diagnostic performance by learning generalized features from limited samples. The proposed two types of blocks can also function individually, and can be implemented as a feature extractor by being implemented at the front of the network regardless of MLP, CNN or RNN. The proposed method not only showed that high generalization performance can be achieved through strong structural regulation, but also suggests that it is important to design the network using knowledge of mechanical engineering and signal processing domain

Contribution 2: Development of Specialized Data Augmentation Option to Vibration Signal

The newly proposed data augmentation technique in this thesis, CEDA is applicable option for vibration signal when the training sample is insufficient. The CEDA is not harm the physical properties of the vibration signal but enhance the features of the fault of rotating machinery. Augmented samples increase the size of the training sample and prevents the network to overfit the training data. This leads to higher and robust performance of the diagnosis result from the diagnosis model.

Contribution 3: Optimization for Performance of Transfer Learning by Providing Intermediate Approach for Parameter Transfer

The SPF approach provides an intermediate option between freezing and fine-tuning to take advantage from both approaches. The performance of the proposed approach was validated using two datasets. The proposed method achieves a maximum diagnostic accuracy when the training samples are insufficient regardless of the network size. In the proposed method, the network is able to learn new features from target domain with reduced network capacity, and mitigates overfitting and enhance the domain adaptability. Through this study, we showed that the configuration of parameter repurposing is influential, and the proposed method provide a new option for the optimization of the parameter transfer approach

References

1. Perkins, D.N. and G. Salomon, *Transfer of learning.* International encyclopedia of education, 1992. **2**: p. 6452-6457.
2. Kirkpatrick, J., et al., *Overcoming catastrophic forgetting in neural networks.* Proceedings of the National Academy of Sciences, 2017. **114**(13): p. 3521-3526.
3. Pan, S.J. and Q. Yang, *A survey on transfer learning.* IEEE Transactions on knowledge and data engineering, 2010. **22**(10): p. 1345-1359.
4. Weiss, K., T.M. Khoshgoftaar, and D. Wang, *A survey of transfer learning.* Journal of Big Data, 2016. **3**(1): p. 9.
5. Wang, C. and S. Mahadevan. *Heterogeneous domain adaptation using manifold alignment*. in *IJCAI proceedings-international joint conference on artificial intelligence*. 2011.
6. Kulis, B., K. Saenko, and T. Darrell. *What you saw is not what you get: Domain adaptation using asymmetric kernel transforms*. in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. 2011. IEEE.
7. Zhu, Y., et al. *Heterogeneous Transfer Learning for Image Classification*. in *AAAI*. 2011.
8. Li, W., et al., *Learning with augmented features for supervised and semi-supervised heterogeneous domain adaptation.* IEEE transactions on pattern analysis and machine intelligence, 2014. **36**(6): p. 1134-1148.
9. Szegedy, C., et al. *Inception-v4, inception-resnet and the impact of residual connections on learning*. in *AAAI*. 2017.

10. Russakovsky, O., et al., *Imagenet large scale visual recognition challenge.* International Journal of Computer Vision, 2015. **115**(3): p. 211-252.

11. Shin, H.-C., et al., *Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning.* IEEE transactions on medical imaging, 2016. **35**(5): p. 1285-1298.

12. Krizhevsky, A., I. Sutskever, and G.E. Hinton. *Imagenet classification with deep convolutional neural networks*. in *Advances in neural information processing systems*. 2012.

13. Yosinski, J., et al. *How transferable are features in deep neural networks?* in *Advances in neural information processing systems*. 2014.

14. Marmanis, D., et al., *Deep learning earth observation classification using ImageNet pretrained networks.* IEEE Geoscience and Remote Sensing Letters, 2016. **13**(1): p. 105-109.

15. Ng, H.-W., et al. *Deep learning for emotion recognition on small datasets using transfer learning*. in *Proceedings of the 2015 ACM on international conference on multimodal interaction*. 2015. ACM.

16. Oquab, M., et al. *Learning and transferring mid-level image representations using convolutional neural networks*. in *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*. 2014. IEEE.

17. Shao, S., et al., *Highly-accurate machine fault diagnosis using deep transfer learning.* IEEE Transactions on Industrial Informatics, 2018.

18. Kim, S., et al. *Transfer learning for automated optical inspection*. in *Neural Networks (IJCNN), 2017 International Joint Conference on*. 2017. IEEE.

19. Deng, J., et al., *Imagenet: A large-scale hierarchical image database.* 2009.

20. Zhang, R., et al., *Transfer learning with neural networks for bearing fault*

diagnosis in changing working conditions. IEEE Access, 2017. **5**: p. 14347-14357.

21. Selvaraju, R.R., et al. *Grad-cam: Visual explanations from deep networks via gradient-based localization*. in *Proceedings of the IEEE International Conference on Computer Vision*. 2017.

22. Jing, L., et al., *A convolutional neural network based feature learning and fault diagnosis method for the condition monitoring of gearbox.* Measurement, 2017. **111**: p. 1-10.

23. Wen, L., et al., *A new convolutional neural network-based data-driven fault diagnosis method.* IEEE Transactions on Industrial Electronics, 2017. **65**(7): p. 5990-5998.

24. Chen, Z., C. Li, and R.-V. Sanchez, *Gearbox fault identification and classification with convolutional neural networks.* Shock and Vibration, 2015. **2015**.

25. Lu, C., Z. Wang, and B. Zhou, *Intelligent fault diagnosis of rolling bearing using hierarchical convolutional network based health state classification.* Advanced Engineering Informatics, 2017. **32**: p. 139-151.

26. He, M. and D. He, *Deep learning based approach for bearing fault diagnosis*. IEEE Transactions on Industry Applications, 2017. **53**(3): p. 3057-3065.

27. Xia, M., et al., *Fault diagnosis for rotating machinery using multiple sensors and convolutional neural networks.* IEEE/ASME Transactions on Mechatronics, 2017. **23**(1): p. 101-110.

28. Zhang, W., et al., *A deep convolutional neural network with new training methods for bearing fault diagnosis under noisy environment and different*

working load. Mechanical Systems and Signal Processing, 2018. **100**: p. 439-453.

29. Guo, X., L. Chen, and C. Shen, *Hierarchical adaptive deep convolution neural network and its application to bearing fault diagnosis.* Measurement, 2016. **93**: p. 490-502.

30. Sun, W., et al., *Convolutional discriminative feature learning for induction motor fault diagnosis.* IEEE Transactions on Industrial Informatics, 2017. **13**(3): p. 1350-1359.

31. Liu, H., L. Li, and J. Ma, *Rolling bearing fault diagnosis based on STFT-deep learning and sound signals.* Shock and Vibration, 2016. **2016**.

32. Mehala, N. and R. Dahiya. *A comparative study of FFT, STFT and wavelet techniques for induction machine fault diagnostic analysis*. in *Proceedings of the 7th WSEAS international conference on computational intelligence, man-machine systems and cybernetics, Cairo, Egypt*. 2008.

33. Cocconcelli, M., et al., *STFT based approach for ball bearing fault detection in a varying speed motor,* in *Condition Monitoring of Machinery in Non-Stationary Operations*. 2012, Springer. p. 41-50.

34. Wang, D., W.T. Peter, and K.L. Tsui, *An enhanced Kurtogram method for fault diagnosis of rolling element bearings.* Mechanical Systems and Signal Processing, 2013. **35**(1-2): p. 176-199.

35. Lei, Y., et al., *Application of an improved kurtogram method for fault diagnosis of rolling element bearings.* Mechanical Systems and Signal Processing, 2011. **25**(5): p. 1738-1749.

36. Yu, D., J. Cheng, and Y. Yang, *Application of EMD method and Hilbert spectrum to the fault diagnosis of roller bearings.* Mechanical systems and

signal processing, 2005. **19**(2): p. 259-270.

37. Peter, W.T. and D. Wang, *The design of a new sparsogram for fast bearing fault diagnosis: Part 1 of the two related manuscripts that have a joint title as "Two automatic vibration-based fault diagnostic methods using the novel sparsity measurement–Parts 1 and 2".* Mechanical Systems and Signal Processing, 2013. **40**(2): p. 499-519.

38. Zhang, Y. and R. Randall, *Rolling element bearing fault diagnosis based on the combination of genetic algorithms and fast kurtogram.* Mechanical Systems and Signal Processing, 2009. **23**(5): p. 1509-1517.

39. Rubini, R. and U. Meneghetti, *Application of the envelope and wavelet transform analyses for the diagnosis of incipient faults in ball bearings.* Mechanical systems and signal processing, 2001. **15**(2): p. 287-302.

40. Ho, D. and R. Randall, *Optimisation of bearing diagnostic techniques using simulated and actual bearing fault signals.* Mechanical systems and signal processing, 2000. **14**(5): p. 763-788.

41. Lim, Y. and S. Parker, *FIR filter design over a discrete powers-of-two coefficient space.* IEEE Transactions on Acoustics, Speech, and Signal Processing, 1983. **31**(3): p. 583-591.

42. Vetterli, M. and D. Le Gall, *Perfect reconstruction FIR filter banks: Some properties and factorizations.* IEEE Transactions on Acoustics, Speech, and Signal Processing, 1989. **37**(7): p. 1057-1071.

43. Bhuiyan, S.M., R.R. Adhami, and J.F. Khan, *Fast and adaptive bidimensional empirical mode decomposition using order-statistics filter based envelope estimation.* EURASIP Journal on Advances in Signal Processing, 2008. **2008**(1): p. 728356.

44. Meng, Q., et al. *An empirical envelope estimation algorithm*. in *2013 6th International Congress on Image and Signal Processing (CISP)*. 2013. IEEE.

45. Flandrin, P., G. Rilling, and P. Goncalves, *Empirical mode decomposition as a filter bank.* IEEE signal processing letters, 2004. **11**(2): p. 112-114.

46. Rilling, G., P. Flandrin, and P. Goncalves. *On empirical mode decomposition and its algorithms*. in *IEEE-EURASIP workshop on nonlinear signal and image processing*. 2003. NSIP-03, Grado (I).

47. Wu, Z. and N.E. Huang, *Ensemble empirical mode decomposition: a noise-assisted data analysis method.* Advances in adaptive data analysis, 2009. **1**(01): p. 1-41.

48. Szegedy, C., et al. *Rethinking the inception architecture for computer vision*. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.

49. Chollet, F. *Xception: Deep learning with depthwise separable convolutions*. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017.

50. Perez, L. and J. Wang, *The effectiveness of data augmentation in image classification using deep learning.* arXiv preprint arXiv:1712.04621, 2017.

51. Salamon, J. and J.P. Bello, *Deep convolutional neural networks and data augmentation for environmental sound classification.* IEEE Signal Processing Letters, 2017. **24**(3): p. 279-283.

52. Eitel, A., et al. *Multimodal deep learning for robust RGB-D object recognition*. in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2015. IEEE.

53. Wang, J. and L. Perez, *The effectiveness of data augmentation in image*

classification using deep learning. Convolutional Neural Networks Vis. Recognit, 2017.

54. Ronneberger, O., P. Fischer, and T. Brox. *U-net: Convolutional networks for biomedical image segmentation*. in *International Conference on Medical image computing and computer-assisted intervention*. 2015. Springer.

55. Cubuk, E.D., et al., *Autoaugment: Learning augmentation policies from data.* arXiv preprint arXiv:1805.09501, 2018.

56. Kim, S., et al. *Transfer learning for automated optical inspection*. in *2017 International Joint Conference on Neural Networks (IJCNN)*. 2017. IEEE.

57. Zhang, W., et al., *A new deep learning model for fault diagnosis with good anti-noise and domain adaptation ability on raw vibration signals.* Sensors, 2017. **17**(2): p. 425.

국문 초록

딥러닝 기반 회전기계 진단을 위한 진동신호 전처리 및 변환 연구

딥러닝은 기계 응용 분야의 결함 진단을 위한 유망한 접근 방식이다. 딥러닝 기술은 많은 양의 데이터를 학습하여 진단 모델의 개발을 용이하게 한다. 그러나 산업 분야에서는 많은 양의 데이터를 얻을 수 없거나 얻을 수 있더라도 고장 데이터는 일반적으로 획득하기 매우 어렵기 때문에 딥러닝 방법의 사용은 쉽지 않다. 회전 기계의 진단을 위하여 딥러닝을 학습시킬 때 발생하는 고장 데이터 부족 문제에 대처하기 위해 이 논문은 3 가지 연구를 제안한다. 1) 향상된 진동 특징 학습을 위한 필터-엔벨롭 네트워크 구조 2) 진동데이터 생성을 위한 Cepstrum 기반 데이터 증량법 3) 전이 학습에서 효율적인 파라미터 전이를 위한 선택적 파라미터 동결법. 첫 번째 연구는 진동 데이터에 대한 강건한 특징을 배우기 위해 신경망에 대한 새로운 형태의 네트워크 블록들을 제안한다. 합성곱 신경망을 포함하는 종래의 신경망은 학습 데이터가 작은 경우에 데이터로부터 편향된 특징을 배우는 경향이 있으며, 이는 다른 조건에서 작동하는 경우나 다른 시스템에 대해 적용되었을 때 낮은 진단 성능을 보인다. 따라서 본 연구는 기존의 신경망에 함께 사용될 수 있는 필터 블록 및 엔벨롭 블록을 제안한다.

각 블록은 주파수 필터와 엔벨롭 추출 기능을 네트워크 내에서 스스로 학습하여 신경망이 제한된 학습 진동데이터로부터 보다 강건하고 일반화된 특징을 학습하도록 한다. 두 번째 연구는 진동 신호의 진단 데이터에 특화된 새로운 데이터 증량법을 제안한다. 뒤집기, 회전 또는 전단과 같은 데이터 확대를 위한 이미지 데이터를 위한 기존의 기술이 1 차원 진동 데이터에 적합하지 않으며, 진동 신호의 물리적 특성에 맞지 않는 신호를 생성할 수 있다. 물리적 특성을 잃지 않고 진동 데이터를 증량하기 위해 제안된 방법은 cepstrum 의 주요성분을 추출하고 조정하여 역 cepstrum 을 수행하는 방식으로 새로운 샘플을 생성한다. 제안된 방법을 통해 데이터를 생성하여 증량된 데이터세트는 진단 모델 학습에 대해 성능향상을 가져온다. 세 번째 연구는 전이 학습에 사용되는 파라미터 전이를 위한 새로운 파라미터 재학습법을 제안한다. 제안된 선택적 파라미터 동결법은 소스 네트워크에서 전이된 파라미터를 선택적으로 동결하고 대상 도메인에 대해 불필요한 파라미터만 재학습하여 대상 데이터가 진단 모델에 재학습될 때의 과적합을 줄이고 소스 네트워크의 성능을 보존한다. 제안된 세 방법은 독립적으로 또는 동시에 진단모델에 사용되어 부족한 고장데이터로 인한 진단성능의 감소를 경감하거나 더 높은 성능을 이끌어낼 수 있다.

주요어: 고장진단
회전체
구름요소 베어링
딥러닝
합성곱 신경망

전이학습

데이터 증량

서로 다른 컨디션에서의 진단

학번: 2012-23163