공학석사학위논문

딥러닝 기반 음향 이상 강도 추정

Deep Learning Based
Acoustic Fault Severity Estimation

2019 년 2 월

서울대학교 대학원

기계항공공학부

송 주 환

Abstract

Deep Learning Based Acoustic Fault Severity Estimation

Joowhan Song
Department of Mechanical and Aerospace Engineering
The Graduate School
Seoul National University

This research proposes a deep learning-based method to estimate an intermediate severity fault state of acoustic data using a model trained only with normal and severe fault labels. First, two types of synthesized acoustic faults with five parameters were designed to simulate a gradually increasing fault. Then, a pretrained CNN model was applied to spectrogram images built from the data. The results from this model prove that classification of both normal and severe faults is possible with high accuracy. However, distinguishing intermediate faults was not possible, even with a fine-tuned model of highest accuracy. To overcome this limitation, latent space features were extracted using the model. Based on this information, the feature values were shown to gradually change as the severity of the fault increased in the reduced-dimension space. This phenomenon suggests that it is possible to map data with intermediate-level faults in the space somewhere between normal and severe fault clusters. The method was tested on real data, including non-acoustic vibrational data. It is anticipated that the proposed method can be applied not only to acoustic signals but also to any signals with a fault characteristic that gradually changes in the time-frequency domain as the fault propagates.

Keywords: Acoustic fault
Deep learning
Spectrogram
Fault classification
Fault estimation

Student Number: 2017-22983

Table of Contents

List of Tables

List of Figures

Chapter 1. Introduction

1.1 Motivation

Quality inspection of a product is a confirmation process that determines whether a product from a manufacturing line is suitable to be delivered to a customer. If delivered to a customer, a product that does not satisfy the examination criteria may incur additional expenses that arise from after-sale customer service. Inspection criteria may include both functional and aesthetic standards; thus, a comprehensive approach is required to verify these standards.

Visual inspection is one of the most common inspection methods. Visual inspection focuses on the existence of visual flaws, like incorrect assembly, poor painting, and/or cracks. Since visual inspection is widely utilized in a variety of areas, there have been many attempts to automate this process using various principles, from rule-based methods to deep learning [1].

Another type of commonly used examination is an acoustic inspection. Targets of an acoustic inspection may include an operating product or running equipment. By setting sound from the desirable condition as a reference, one may distinguish a sound with different characteristics from the reference. Since differences in an acoustic signal may indicate condition changes in the source of the sound, this may imply defects, such as wear, cracks, and/or looseness in running equipment. This is why data engineers have tried to detect anomalies with acoustic signals.

Meanwhile, regardless of the condition of the source, some sounds are simply undesirable or irritating. A product with such acoustic characteristics is considered

inappropriate to be delivered to any user.

As compared to visual data, there have been relatively few attempts to automate inspection of acoustic signals. This is likely because the complicated nature of acoustic signals makes the automation process difficult. For example, while visual data is a time-invariant, two-dimensional array acoustic data is a temporal signal that varies across time.

Informal knowledge of the state of the industry suggests that, presently, the acoustic quality inspection process is poorly automated. Even in cases where automatic examination software may be implemented on an inspection line, the software can classify a sound only as “pass” or “fail” by following predefined criteria related to time and frequency features.

There are two primary limitations in the current approach. First, the fact that determination criteria must be set by a user implies that the user must define the unwanted acoustic characteristic in a quantitative way before running the software. This process requires a thorough knowledge of potential irregular signals in advance of testing. It also requires user knowledge of acoustics, signal processing, feature extraction, and statistics. Second, the software cannot tell the degree to which an identified sound is different from a reference “normal” sound, since the software can only decide if a sound passes or fails. To determine an appropriate action to fix the sound, or to eliminate the cause of the sound, the severity of the fault can be an important measure.

Because of the above-mentioned limitations, the inspection lines the author has witnessed in industry still require final confirmation by a human inspector; thus, this is still far from an automated process. Since analyzing specific acoustic faults requires a deep understanding of the physical and psychophysical aspects of a sound,

establishing rules for automated acoustic inspection is not an easy task for field engineers. Even though conventional acoustic features like Mel-Frequency Cepstral Coefficients (MFCCs) [2], loudness, sharpness, fluctuation strength, and roughness [3,4] have been invented to depict characteristics of an acoustic sound, applying these features to specific cases remains challenging.

This project was inspired by the belief that a task like this – which is not difficult for a human – should also be possible for a machine. Especially with recent advances in deep-learning algorithms, an inspection architecture should be developed that is much faster and easier than conventional ones. Furthermore, a deep-learning architecture can be developed that can capture physical characteristics as features that can express the intensity of the acoustic characteristics.

Based on these facts, the research outlined in this thesis focused on developing a deep-learning-based fault estimation method that can be used for general acoustic signals.

1.2 Scope of the Research

The research outlined in this thesis focuses on automating sound quality inspection via a deep-learning-based method. The proposed model can be trained with only normal and severe fault labels, and then applied to estimate the severity of untrained data with intermediate faults, overcoming the limitations mentioned in Section 1.1.

This research focuses on four issues. The first issue is the selection of an appropriate preprocessing method for acoustic signals. The second issue is the selection of the most accurate deep-learning model. The third consideration examines appropriate and realistic data feeding. The final issue addresses how to represent high

dimensional features in an intuitive way, thereby allowing estimation of intermediate faults.

Since acoustic signals are one or two channel consecutive arrays, mostly with a 44.1kHz sampling rate, a method must be devised to convert this data into a proper data format, while considering two important factors. First, this transformation must preserve the physical characteristics of the acoustic signal. A transformed shape should also be determined accordingly, based on the input format of the deep-learning architecture.

Among possible architectures (e.g., neural networks, convolutional networks, and recurrent networks) a model should be selected that is suitable for the given task. Selection of a specific version of the network must also be considered.

The third issue outlined above is the key idea of this research. To train a model with a deep-learning algorithm, labels are needed for the "right" answer; this allows the model to learn what is wrong and right. Unlike general studies on deep learning, this research focuses on a continuous label, not a categorical one. What this means is that there is some amount of degree in the severity of a fault that enables comparison of how severe one fault is in comparison with another.

If the labels are perfect, which means that continuous severity can be divided into 10, 100, or more classes, classification of the fault data can be attempted. However, this is not a realistic assumption from two perspectives. First, labeling is done by a person, and no matter how trained the person is, it is not possible for a human to judge subtle differences in severity without any mistakes. Furthermore, even if perfect labeling is assumed to be possible, there will be an insufficient amount of data in each class to be trained.

Therefore, this research attempts to verify something different from general deep-

learning research. By assuming that there is a sufficient amount of both normal and fault data labels that can be agreed-upon without being questioned, the proposed method suggests first training the model for classification. Then, using the trained model, new data with intermediate severity can be classified. The model has not experienced intermediate severity events; thus, this process can show how the model tends to classify the given data into one of the trained classes, which seems somewhat unreasonable.

The proposed method assumes that existing deep-learning algorithms already have an ability to capture the information of input data as features. However, when the features go through the sequence of fully connected layers, they lose their information and become overfitted and unable to solve the original task, i.e., classification of normal and fault in our case. This research seeks to prove that features extracted from the data contain information about how severe the fault is.

Visualization of deep-learning features is the last issue of interest for this research. To the best of our knowledge, it is not yet possible to understand the exact meaning of each feature in a deep-learning architecture. High dimensionality of a feature also makes it difficult to comprehend the results. By reducing the dimension of the features, it will be possible to achieve a more perceptible representation of the result.

The following issues are beyond the scope of this research. No novel method is suggested for preprocessing of sound data. This research also does not include optimization of a deep-learning architecture. Instead, an existing model suitable for the purposes of this research is chosen. The visualization methods considered are also limited to previously studied, well-known methods.

1.3 Thesis Layout

This thesis is organized as follows. Section 1 introduces the background of the proposed research, outlining existing research challenges and why they are of interest. Section 2 provides an introduction to the important concepts and methodologies used to develop the ideas proposed in the research. Section 3 describes the proposed method step by step. Experimental results that examine both synthesized and real acoustic data are presented in section 4. A conclusion and suggestions for future work are provided in the last section.

Chapter 2. Research Background

This section delivers an explanation of the terminologies and concepts needed to help understand the suggested idea. The contents of this section mostly summarize existing concepts and techniques available in the literature.

2.1 Types of Acoustic Faults

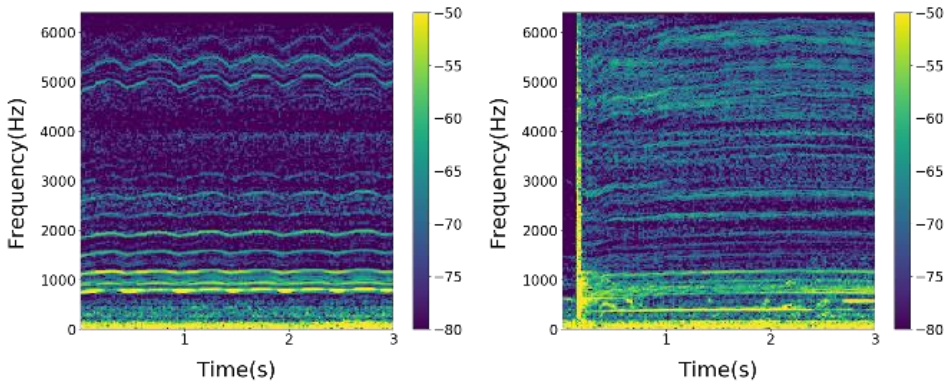
An acoustic fault can be described as a sound that is irritating to the average person. Unfortunately, this is a vague definition that is unsuitable for designing an objective experiment. As mentioned in section 1, an acoustic fault must be first quantified to be useful for the purposes of this research. To derive a proper quantification method based on real acoustic data, a closer look at real acoustic fault signals is required.

Figure 2-1 shows a log-scaled spectrogram of real acoustic data recorded from a quality examination line. From the figure, some general characteristics of sound and a few fault characteristics can be determined. These characteristics include harmonics and frequency amplitude ratio.

Figures (a) and (b) each express a different type of fault. Periodic fluctuations in the frequency domain are observed as time passes in (a). This fluctuation represents a siren-like frequency modulation in the original time domain. The vertical yellow line observed in (b) implies an impulse generated and dissipating within a short time interval.

These two types of acoustic faults are the focus of this thesis research. In this paper, the inconsistent behavior in the frequency domain is called a *modulation fault* and

the short-time impulse is called an *impulsive fault*.



(a) (b)

Figure 2-1 Log-scaled spectrogram of acoustic data with a fault:

(a) Modulation fault (b) Impulsive fault

To simulate intermediate severity, important acoustic characteristics of both the modulation and impulse faults were determined. Three characteristics of a modulation fault, and two characteristics of an impulse fault, were determined as the main parameters. Further information on how data are synthesized is provided in section 3.1.

2.2 Spectrogram

From previous research examining both deep-learning and non-deep-learning methods, MFCCs (Mel Frequency Cepstral Coefficients) have emerged as the most common representative form for acoustic signals [5-12]. Figure 2-2 shows the process used to extract MFCCs from an acoustic signal. In simple terms, MFCCs are log-scaled time-frequency features windowed by filter banks called mel filter banks.

It is known that humans perceive sound intensity in a logarithmic scale, and that 23 mel filter banks are considered enough in previous research [5].

Since MFCCs have a two-dimensional shape with time and frequency axes, a CNN (Convolutional Neural Network) architecture is commonly used for deep-learning research on acoustic signals (e.g., sound classification [7, 9, 10, 12], sound detection [6, 8, 9, 11], speech recognition, and speech synthesis). Based on this convention from prior research, the research outlined in this thesis assumes that spectrogram features preserve important characteristics of acoustic data without losing too much information after transformation to a two-dimensional array.

In this research, however, a log-scaled spectrogram is used instead of MFCC. Even though mel filter banks may emphasize a certain frequency, resolution in the frequency domain is reduced to the number of filter banks. This research is based on the belief that deep-learning convolutional layers will learn to emphasize the frequency band if the band is truly crucial for determining the class. Likewise, the research builds on the belief that it is better to minimize the preprocessing steps before applying deep learning. Still, it is assumed that the weight multiplying process in convolutional kernels will have difficulty imitating log-like computation; thus, a log-scaled spectrogram is chosen as an input.

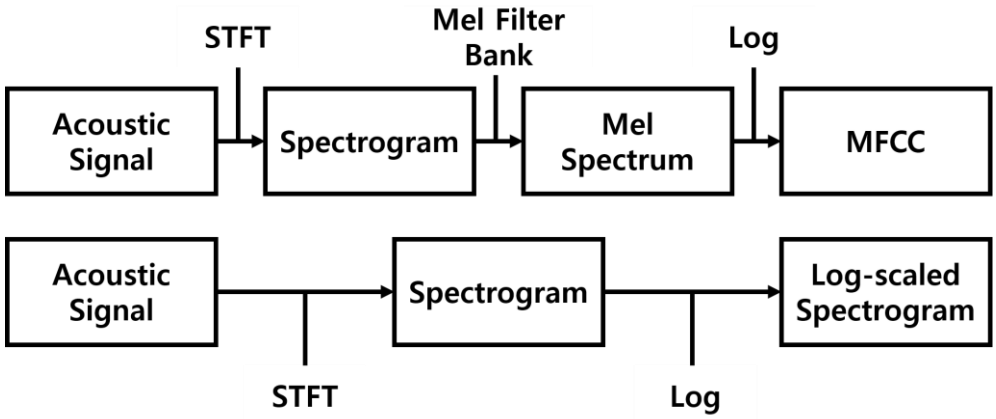


Figure 2-2 Comparison between Mel-frequency Cepstral Coefficients and a Log-scaled spectrogram

2.3 CNN Models

The convolutional neural network (CNN) is the most common form of architecture used in image deep learning. Figure 2-3 shows the basic form of the CNN architecture. CNN is comprised of two parts: feature extraction layers and classification layers. The purpose of the feature extraction layers is to extract features that represent the input image. The term feature has almost the same meaning here as when it is used in other fields of data science, including PHM. The biggest difference is that while conventional features have extraction steps that can be physically explained, these features lack physical meanings. To the best of our knowledge, there is no clear explanation of how to interpret the deep learning feature extraction process. It is known that only the classification accuracy puts a value on the importance of the feature and the repeated learning process updates the feature to

maximize the prediction accuracy.

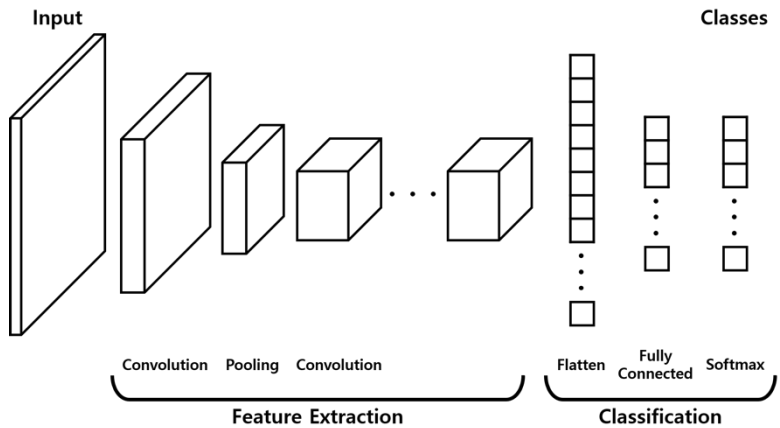


Figure 2-3 Simple CNN architecture

Many deep-learning acoustic experiments are based on CNN architectures [7, 8, 10, 11, 12]. This is because most of the progress in deep learning has been made with visual data. Since a spectrogram is also an image, it is appropriate to utilize a CNN architecture that has been proven to be effective in image classification [13]. The specific structure of CNN models is provided by the Keras opensource deep-learning library in [14].

2.3.1 VGG-16 and VGG-19

VGGNet [15] is a structure that was suggested to solve the ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) 2014. VGGNet is an improved model, as compared to AlexNet. Compared to AlexNet, VGGNet replaced former kernels with smaller sizes, 3×3. With the reduced number of parameters in kernels, the VGGNet

approach was able to stack more layers than the AlexNet. The numbers 16 and 19 imply the numbers of weighted layers in each model.

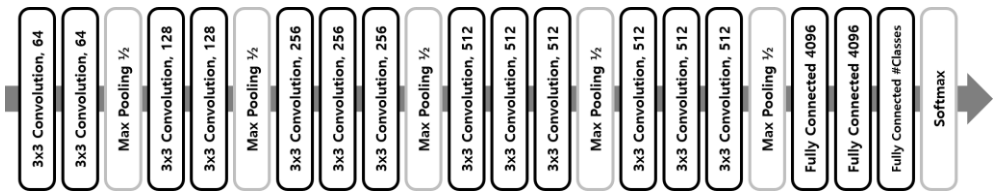


Figure 2-4 Structure of VGG-16

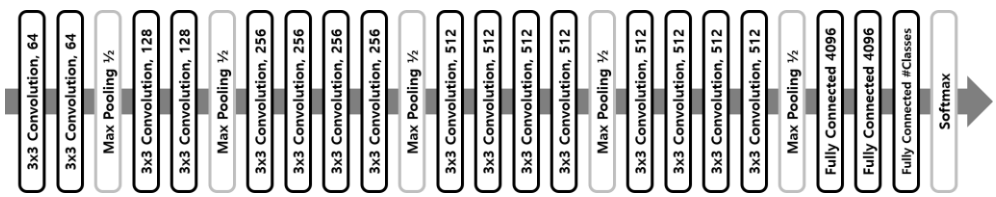


Figure 2-5 Structure of VGG-19

2.3.2 ResNet-50

ResNet [16] is a structure suggested by Microsoft in 2015. One of the most distinguishable characteristics of ResNet is the depth of the model. A depth of up to 152 layers was suggested in the paper as a way to solve the ImageNet classification problem. To prevent a gradient descent problem, short-cut connections were established in the model. A short cut is an identity layer that connects an input directly to a deeper layer, skipping two or three layers ahead.

A residual connection is an element-wise addition of two tensors, represented as a

circle with a plus sign in Figure 2-6. To add up the two tensors, the size of these tensors must be the same. Since the model was proven to be effective, the residual connection approach was also used in later models.

This research uses a ResNet architecture with 50 layers.

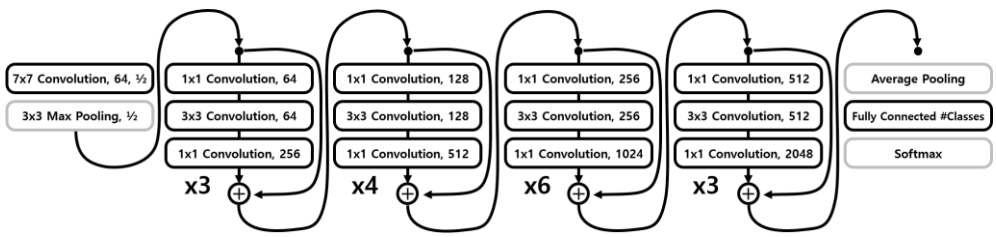


Figure 2-6 Structure of ResNet-50

2.3.3 InceptionV3

The First Inception [17] model was suggested in 2014 by Google. One of the most popular versions of this Inception model is GoogLeNet, which achieved the top score at ILSVRC 2014. Advantages of the Inception model include reduced computational cost and smaller memory space requirements, as compared to other approaches. These advantages are made possible by the Inception module. An Inception module can be described as a channel-separated convolution kernel. As some of the operations between channels are suppressed, computation costs are reduced, and computation time is shortened.

Figure 2-7 shows how an Inception module works. The role of the parallel structure is to compress the depth of an input layer to reduce the computation costs.

In the concatenation step, the output of each convolution is stacked in the depth direction. Therefore, the size of each convolution must be the same.

In this research, we use the third model of Inception [18], as shown in Figure 2-8.

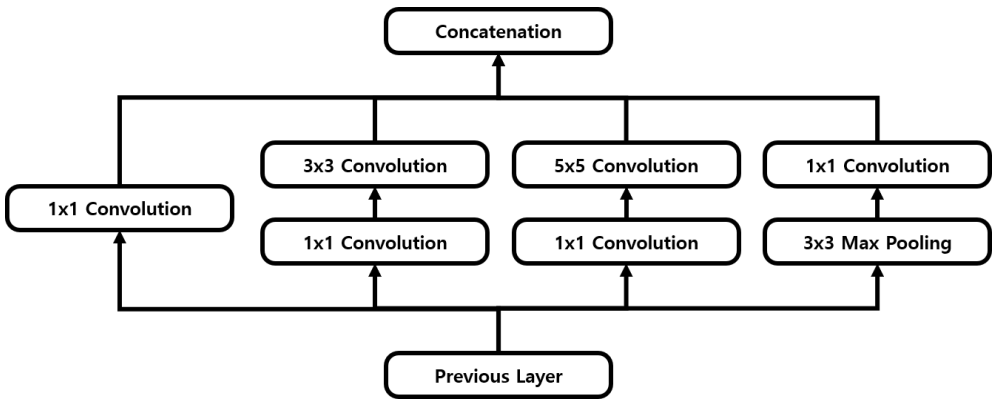


Figure 2-7 Structure of the Inception module

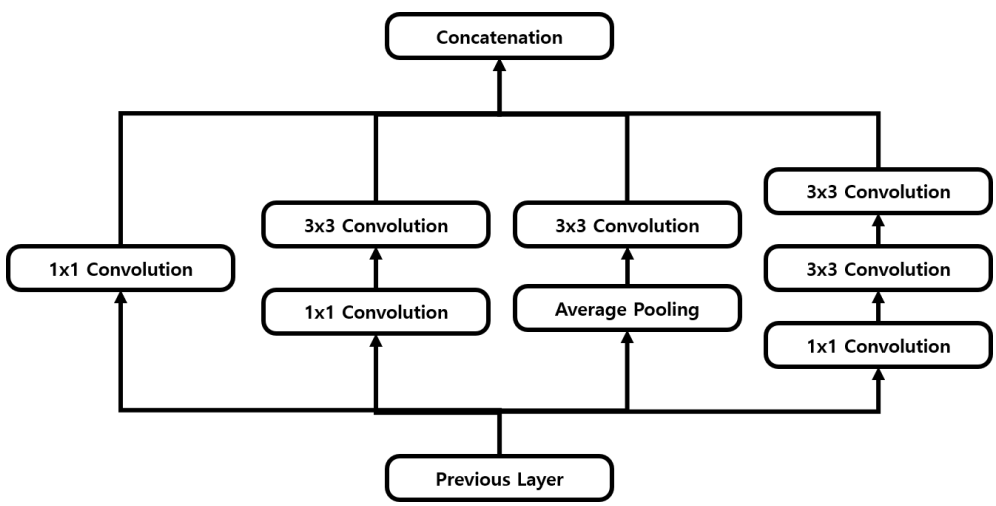


Figure 2-8 Structure of the Inception V3 module

2.3.4 Xception

The Xception [19] model is a successor of Inception. Instead of an Inception module, here, depth-wise separable convolution layers are used. Xception stands for "Extreme Inception," which implies that there is no operation among channels. Figure 2-9 represents the Xception module. The 1x1 convolution layer reduces the depth of the input to one. Since there is no depth, depth-wise operation is suppressed; instead, the output of each convolution layer in the next step is stacked along the depth, resulting in a similar shaped output.

Xception has a similar number of parameters, as compared to InceptionV3. However, it shows better results, implying that Xception is a more efficient architecture. Xception showed better performance on ImageNet, as compared to VGG-16, ResNet-152, and Inception V3.

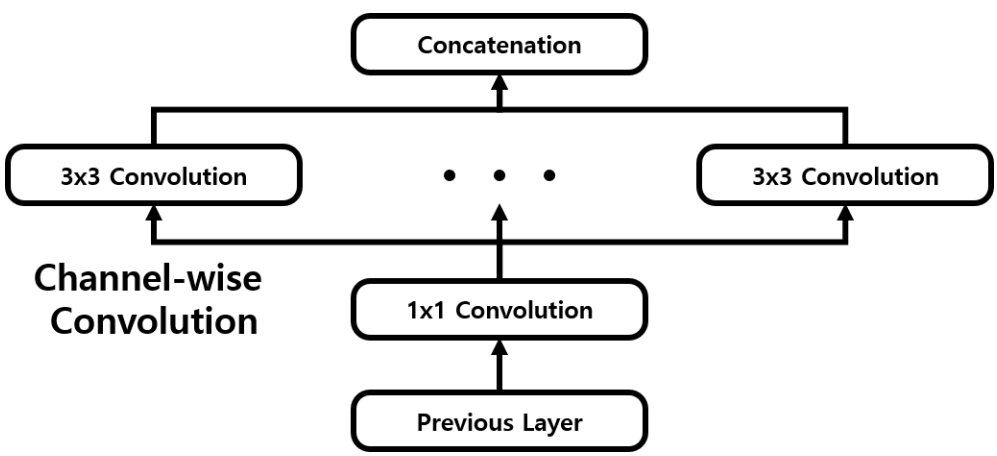


Figure 2-9 Structure of the Xception module

2.4 Transfer Learning

In studies of machine learning, transfer learning indicates techniques that use pretrained weights of a model as the initial weights for training of a different set of data. This concept assumes that the feature extraction structure of a model can be used for general feature extraction tasks. Many different tasks do focus on classification or detection of certain visual characteristics.

Figure 2-10 visualizes the concept of transfer learning. Before training a randomly initialized network from the beginning, the network is trained with a different target dataset. Then, the network is trained again to solve a task with the dataset of interest.

Transfer learning has some advantages. Pretrained weights are trained much faster than random weights. Since there are lots of models already trained for image classification tasks (e.g., ImageNet) it is efficient to start from such models. If the size of the target dataset is small enough that random training is difficult, transfer learning can be a solution.

It is both possible to fix the transferred weight to not be updated or to let it be freely tuned. Previous research has reported that fixed weights are less successful; meanwhile, fine-tuned models showed better results [20].

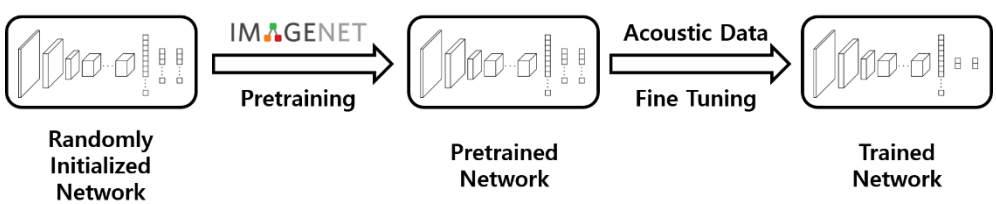


Figure 2-10 Concept of transfer learning

2.5 Latent Space

As mentioned in the Section 2.4, convolutional layers extract features that are used in the classification step. Spatial features extracted by convolutional layers have high dimensionality. The space where features are mapped is called a latent space. As the name implies, latent space features are less distinct in a visual sense than pixel-based input. The classification ability of fully-connected layers implies that latent space features do have information in their labels. DCGAN [21] has shown that arithmetic operation in latent space also makes sense in pixel space, by generating interpolated images with interpolated latent features.

2.5.1 Latent Space Visualization

Visualization of high-dimensional features requires a dimension reduction process. Over 2000 features can be extracted from one acoustic data, depending on the deep learning structure. PCA and t-SNE [22] are commonly used dimension reduction and visualization methods for latent space [20].

Chapter 3. Proposed Estimation Method

In this chapter, a sequence of algorithms is proposed to estimate an intermediate fault.

3.1 Simulating an Acoustic Fault

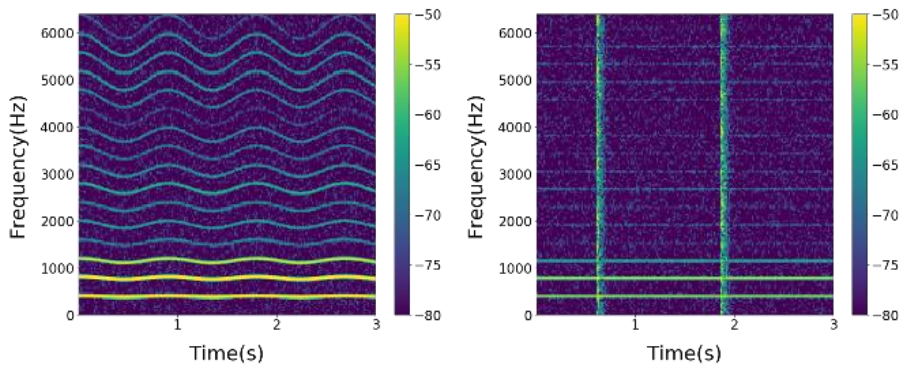
From the observations outlined in section 2.1, it was found that deriving an objective value for the fault severity of real data is difficult for two reasons. First, there are various types of faults, each fault type having different characteristics that require different quantification approaches. Next, even for a specific type of a fault, there are several uncorrelated variables involved; thus, determining the appropriate way of deriving a single representative value is difficult.

Instead of labeling in a subjective and case-specific way that might result in a doubtful conclusion, this research proposes to synthesize data with representative parametric fault indices, which serve as objective fault values. Figure 3-1 shows sample spectrogram images derived from the data. Representative indices are selected to imitate the real data as closely as possible. In this research, the number of fault types is limited to only two for simplicity. Compare this figure (3-1) with Figure 2-1 and note the similarities.

Three primary issues were considered in developing the artificial data. First, randomness in phase, fundamental frequency, and noise was added. Without enough randomness, there could be overfitting during training steps. Second, the number of harmonics was preserved as sixteen, and their amplitude relation was fixed to allow a clear comparison between different levels of amplitude, as shown in Table 3-2.

Finally, five acoustic characteristics were divided into five segments according to their severity factor. Table 3-1 and 3-2 provide more information.

A more detailed explanation is provided in the following sections.



(a) (b)

Figure 3-1 Log-scaled spectrogram of synthesized acoustic data with a fault: (a) Modulation fault (b) Impulsive fault

Table 3-1 Description of the symbols used for synthesized acoustic data

Description	Symbol	Distribution
Phase	$\boldsymbol{\phi}$	$\phi \sim \mathcal{U}(0,\ 2\pi)$
Modulation Phase	$\boldsymbol{\varphi}$	$\varphi \sim \mathcal{U}(0,\ 2\pi)$
Harmonic Frequency	$\boldsymbol{f_H}$	$f_H \sim \mathcal{N}(375,\ 50/3)$
Modulation Amplitude Factor	$\boldsymbol{A_M}$	$A_M \in \{1,\ 2,\ 3,\ 4,\ 5\}$
Modulation Frequency Factor	$\boldsymbol{F}$	$F \in \{1,\ 2,\ 3,\ 4,\ 5\}$
Modulation Pitch Factor	$\boldsymbol{P}$	$P \in \{1,\ 2,\ 3,\ 4,\ 5\}$
Impulse Amplitude Factor	$\boldsymbol{A_I}$	$A_I \in \{1,\ 2,\ 3,\ 4,\ 5\}$
Impulse Occurrence Factor	$\boldsymbol{N}$	$N \in \{1,\ 2,\ 3,\ 4,\ 5\}$
Amplitude Scale	$\boldsymbol{A_s}$	$A_s \sim \mathcal{U}(A_M - 0.5,\ A_M + 0.5)$
Modulation Frequency	$\boldsymbol{f_m}$	$f_m \sim \mathcal{U}(0.4F - 0.4,\ 0.4F)$
Modulation Pitch	$\boldsymbol{P_m}$	$P_m \sim \mathcal{U}(0.8P - 0.8,\ 0.8P)$
Impulse Amplitude Scale	$\boldsymbol{A_{Is}}$	$A_{Is} \sim \mathcal{U}(0.2A_I - 0.2,\ 0.2A_I)$
ith Harmonic Amplitude	$\boldsymbol{A_{hi}}$	$-$
Random Noise	$\boldsymbol{RN}$	$RN \sim \mathcal{U}(-0.0275,\ 0.0275)$

Table 3-2 Amplitude values for the synthesized harmonic data

Symbol	Value	Symbol	Value
$\boldsymbol{A_{h1}}$	0.0205	$\boldsymbol{A_{h9}}$	0.0017
$\boldsymbol{A_{h2}}$	0.0249	$\boldsymbol{A_{h10}}$	0.0020
$\boldsymbol{A_{h3}}$	0.0096	$\boldsymbol{A_{h11}}$	0.0009
$\boldsymbol{A_{h4}}$	0.0020	$\boldsymbol{A_{h12}}$	0.0019
$\boldsymbol{A_{h5}}$	0.0025	$\boldsymbol{A_{h13}}$	0.0026
$\boldsymbol{A_{h6}}$	0.0022	$\boldsymbol{A_{h14}}$	0.0022
$\boldsymbol{A_{h7}}$	0.0032	$\boldsymbol{A_{h15}}$	0.0025
$\boldsymbol{A_{h8}}$	0.0023	$\boldsymbol{A_{h16}}$	0.0012

3.1.1 Modulation Faults

The modulation characteristic of the frequency was formulated with three parameters. The modulation amplitude factor ($\boldsymbol{A_M}$), modulation frequency factor ($\boldsymbol{F}$), and modulation Pitch Factor ($\boldsymbol{P}$), each affect amplitude, modulation period, and frequency range of modulation. The exact equation is given as follows.

$$\sum_{i=1}^{16} A_{hi} \times A_s \times \sin[2\pi[f_H + P_m sin(2\pi f_m t + \varphi)]t \times i + \phi] + RN$$

Using factor values, the specific class in our datasets is denoted in a format like A1F2P3. By setting other factors to 3, except for a factor of interest, three kinds of datasets were developed, as shown in Figure 3-2. The two classes located at each end

serve as the normal and severe fault data, and the three classes in the middle represent the intermediate faults.

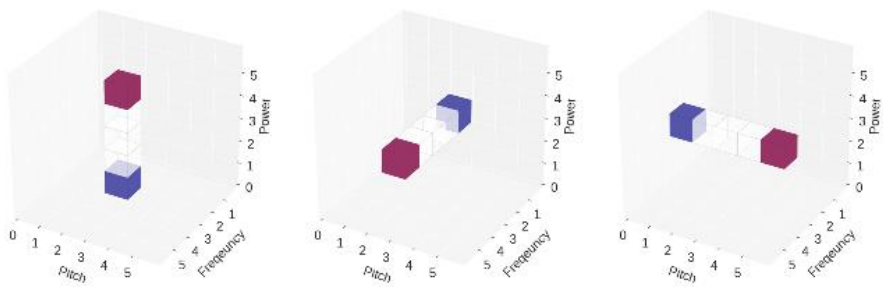


Figure 3-2 Configuration of three datasets for modulation faults

From a visual perspective, the task for a modulation fault can be summarized as follows. Can the curvature of an image pattern be extracted through a deep-learning model that has been trained to classify only a straight pattern and an extremely curvy pattern?

3.1.2 Impulsive Fault

For impulsive faults all factors mentioned in section 3.1.1 were set as 1 and then an impulsive sound was added. The impulse amplitude factor $(\boldsymbol{A_I})$ stands for the amplitude and impulse occurrence factor $(\boldsymbol{N})$ for the number of impulses in the given acoustic data. An impulse is formulated as outlined in the following equation.

$$A_{Is} \times \mathcal{N}(0,\ 2/3) \times exp(-\frac{1}{30}t)$$

The equation above expresses an exponentially decaying randomized impulse.

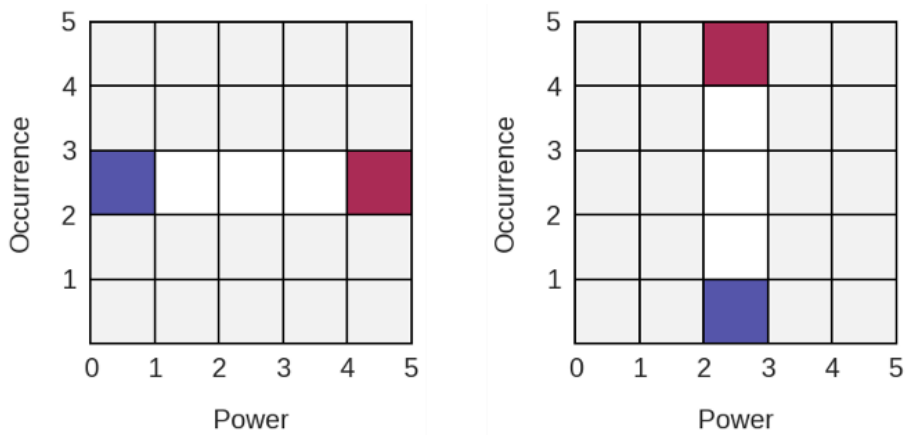


Figure 3-3 Configuration of two datasets of the impulsive fault

As in section 3.1.1, datasets are formulated as shown Figure 3-3.

From a visual perspective, the task for an impulsive fault can be summarized as follows. Can the number of vertical straight lines in an image pattern be extracted through a deep learning model that has been trained only to classify one and five lines?

3.2 Spectrogram Parameters

Conventional convolution-based deep learning models use 224×244 or 299×299-

pixel sized images as an input. Even though it is not mandatory to match the input size, minimizing differences between the original training condition and the transfer learning condition is preferred in this research. In this research, the input size was matched at 224×244 and the spectrogram parameters were set as shown in Table 3-3.

The frequency range was cut off to match the image length of 224 pixels, which corresponds to up to 6.72kHz. Figure 3-1 is a sample that was built according to these parameters. One segment represents 33.3ms, and the time difference between the segments is 16.7ms.

Table 3-3 Spectrogram parameter specifications for synthesized acoustic data

Samples per time segment	1470
Overlapping samples	735
Fourier transform samples per time segment	1470
Sampling frequency	44.1kHz
Acoustic data length	3.75s
Spectrogram size	224×735
Clipped spectrogram size	224×224

Spectrograms in this thesis research were processed with a matplotlib library written in Python. A matplotlib [23] specgram function was used to obtain a log-scaled spectrogram. The other plots in the paper were drawn with matplotlib.

3.3 Transfer Learning and Fine-tuning

The CNN models mentioned in Section 2.3, pretrained with ImageNet, were used for transfer learning and were fine-tuned. Compared to 1000 labels in the ImageNet classification, the dataset in this research has only two labels. Because of this difference, the output end of the deep-learning architecture needed to be modified. This is illustrated in Figure 3-4. For all models, the weights of the convolutional layers, which serve as feature extractors only, were transferred. For classification, only the number of the last layer was changed to two: normal and fault. Because of the change in structure, weights in the classification layers were all randomly initialized.

Weights in both the convolution and fully connected layers were set to be freely modified in the transfer learning stages.

For simplicity, the classification performance of each model was tested with only synthesized data. The results of the models are provided in Section 4.1.1.

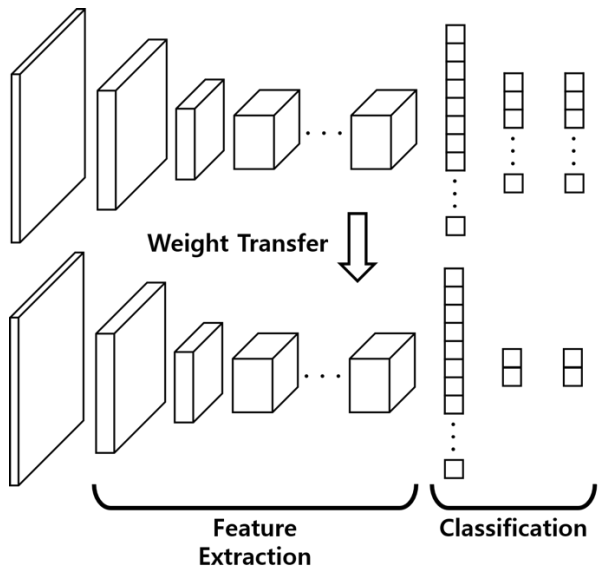


Figure 3-4 Concept of transfer learning in the experiment

3.4 Latent Space Visualization

In the last process of the experiment, latent space features were visualized with PCA and t-SNE and show that latent features do have a gradual varying tendency, from normal to severe fault, indicating that it is possible to estimate intermediate levels of faults in acoustic data.

Since PCA is parametric, it is possible to fit parameters of PCA transform only with normal and severe fault features and then apply trained PCA weights to untrained features. On the other hand, t-SNE is non-parametric; thus, it is not possible to simply apply t-SNE to untrained feature sets. t-SNE is applied to every feature at once.

Chapter 4. Experimental Results

In this chapter, the method introduced in Chapter 3 is applied to both synthesized data and experimental data, specifically, bearing vibration and fan noise.

4.1 Synthesized Data

First, the method was applied to synthesized data. As mentioned in Section 3.3, several model architectures were tested. The best model was selected, and the model extracted the features from all fault severity levels. It is shown that the conventional classification method cannot distinguish the intermediate fault properly. Yet, latent features showed a better distribution in the reduced dimension space.

4.1.1 Transfer Learning Results

The CCN models mentioned in Section 2.3 were used for transfer learning and fine-tuning. The classification result is given in Table 4-1. The classification accuracy was saturated within 5 epochs. That it took only this short amount of time before saturation suggests that two-class classification is an easy task for the deep-learning model. Yet, some cases could not reach 100% accuracy, even after a large enough number of epochs were trained. The reason for this phenomenon requires further investigation. Also, the test accuracy was higher than that of the training data in some cases, which is quite unnatural.

The Xception model that has shown better results among other models with

ImageNet also showed better results with the spectrogram images for the acoustic faults. Therefore, Xception was used as the CNN model for the research outlined in the rest of this thesis. In addition to the better accuracy, the Xception model was also faster and showed a stable prediction rate while training.

Every model was pretrained with ImageNet data supplied by Keras [16], with a TensorFlow background [24].

Table 4-1 Best classification result from the pre-trained models

Model	Modulation			Impulse	
	Amplitude	**Frequency**	**Pitch**	**Amplitude**	**Occurrence**
	Accuracy (%) Train/Test				
VGG16	50.00/50.00	50.00/50.00	50.06/50.00	50.00/50.00	50.00/50.00
VGG19	50.00/50.00	50.00/50.00	50.00/50.00	50.00/50.00	50.00/50.00
ResNet50	99.94/100.0	100.0/100.0	100.0/100.0	98.00/100.0	96.63/100.0
InceptionV3	97.69/100.0	100.0/100.0	100.0/100.0	100.0/100.0	100.0/100.0
Xception	100.0/100.0	100.0/100.0	100.0/100.0	100.0/100.0	100.0/100.0

4.1.2 Prediction Results

The limitations of plain deep-learning models are clarified in this section. Figures 4-1 to 4-5 show prediction results for each acoustic characteristic. Data points in each figure are ordered in increasing levels of fault severity. As shown in all figures, each model succeeds in predicting normal and fault states, corresponding to acoustic data numbers 0 to 1000 and 4000 to 5000. There is also a tendency to classify a weak fault as normal and a strong fault as a fault. For cases of intermediate faults, Figures 4-1

and Figure 4-4 show a gradual transition from normal to fault as the fault level increases, while Figures 4-2, Figure 4-3 and Figure 4-5 show rather wild predictions.

What can be assumed from these figures is that a deep learning classification architecture, i.e., fully connected softmax, tends to predict unknown data as data that the model already knows. For a weak fault, the model judges it as normal, since a weak fault is closer to a normal result than it is to a fault. Exactly the opposite happens in the case of a strong fault.

Even though this is a reasonable result, this is undesirable for this research. As the x axis represents fault severity and the y axis represents the prediction of severity, the goal is to match the x and the y values. From this point of view, the model cannot distinguish data number 3000 to 5000, since prediction results are almost the same.

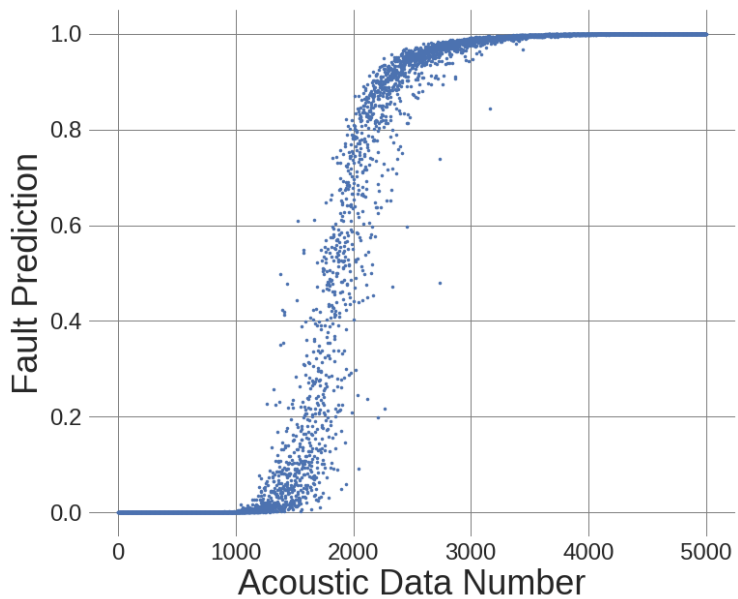


Figure 4-1 Prediction results for an increasing modulation amplitude

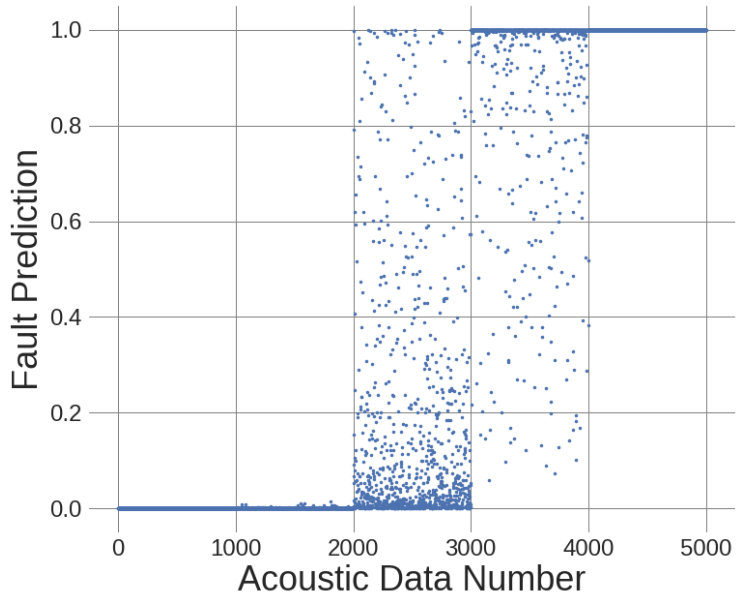


Figure 4-2 Prediction results for an increasing modulation frequency

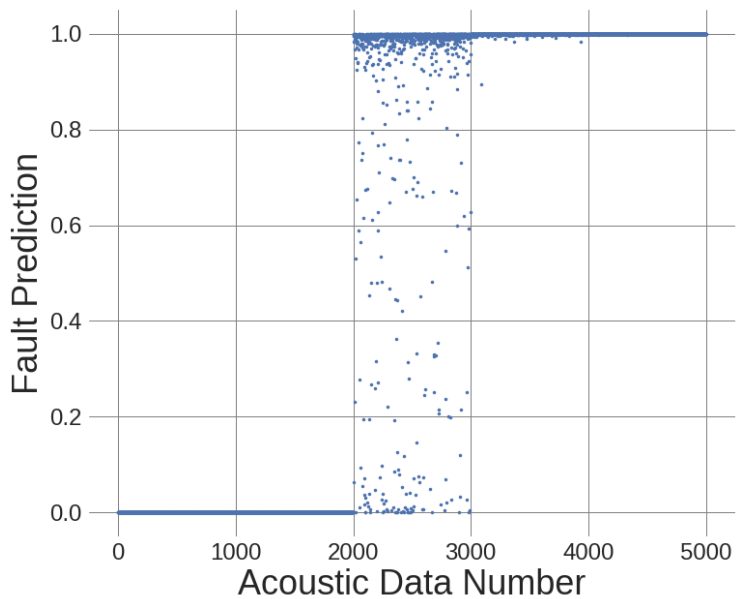


Figure 4-3 Prediction results for an increasing modulation pitch

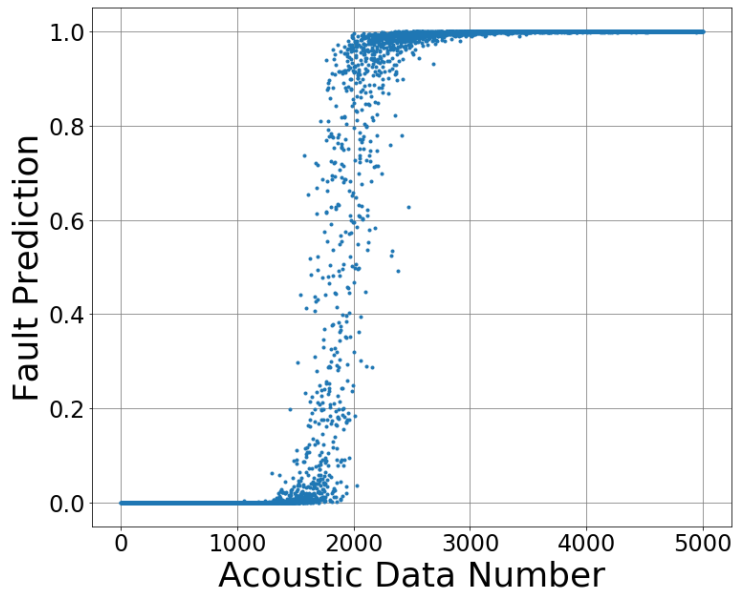


Figure 4-4 Prediction results for an increasing impulse amplitude

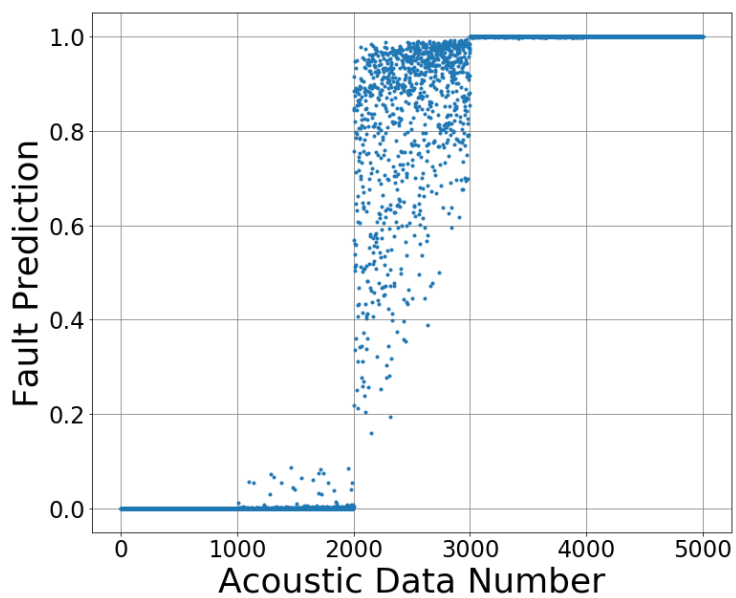


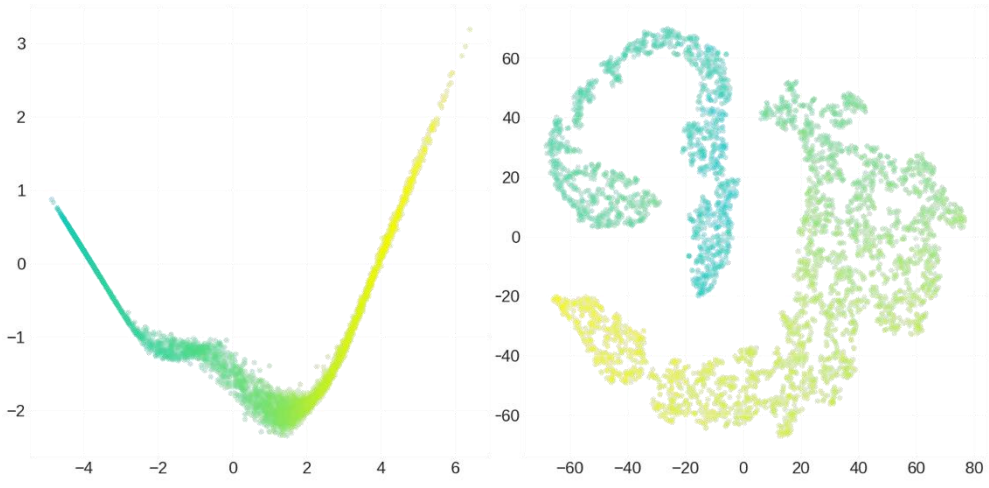
Figure 4-5 Prediction results for an increasing number of impulses

4.1.3 Latent Space Visualization Results

Latent features of the same model that produced the prediction results in Figures 4-1 to Figure 4-5 were used in this section. Figures 4-6 to 4-10 are PCA and t-SNE dimension reduced results. Each axis represents the first and second principle axes. Green represents normal and yellow represents fault. The severity of an intermediate fault is expressed with an interpolated color between green and yellow.

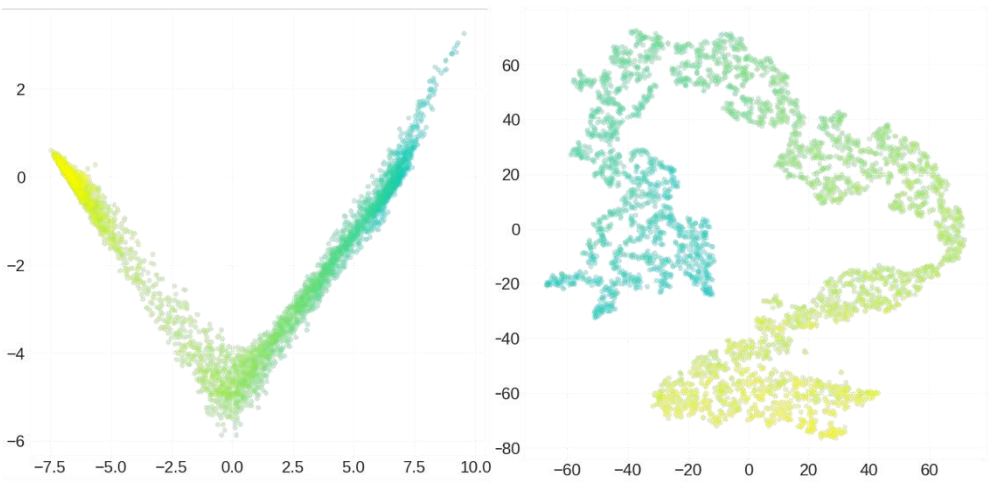
Figure 4-6 (a), Figure 4-7 (a), Figure 4-7 (b), Figure 4-9 (b), and Figure 4-10 (a) show desirable results. It can be seen that latent features are moving in a certain direction along the line as fault severity increases in Figure 4-6 (a) and Figure 4-7 (a). If this distribution can be linearized, a fault index can be derived. Figure 4-7 (b), Figure 4-9 (b) are also appropriate for fault estimation; although the distribution does not form a line, that transition from green to yellow takes place gradually. Figure 4-10 (a) shows an interesting result. Since the number of impulsive faults is discontinuous, discontinuous clusters can be seen to exist in the figure.

There are also some figures showing the limitations of this approach. Figure 4-6 (b), Figure 4-8 (b), and Figure 4-10 (b) show disconnected distributions. Figure 4-8 (a) and Figure 4-9 (a) show uneven distributions.



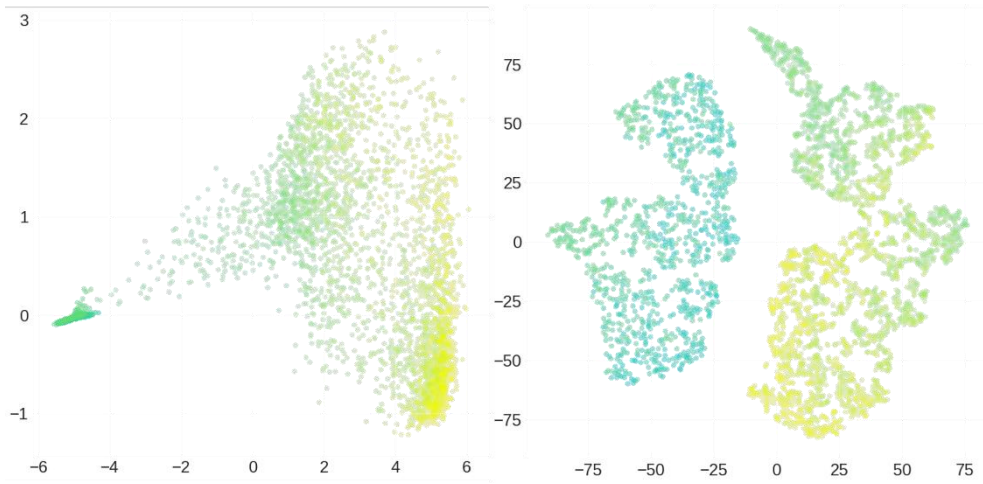
(a) PCA (b) t-SNE

Figure 4-6 Latent space visualization results for an increasing modulation amplitude



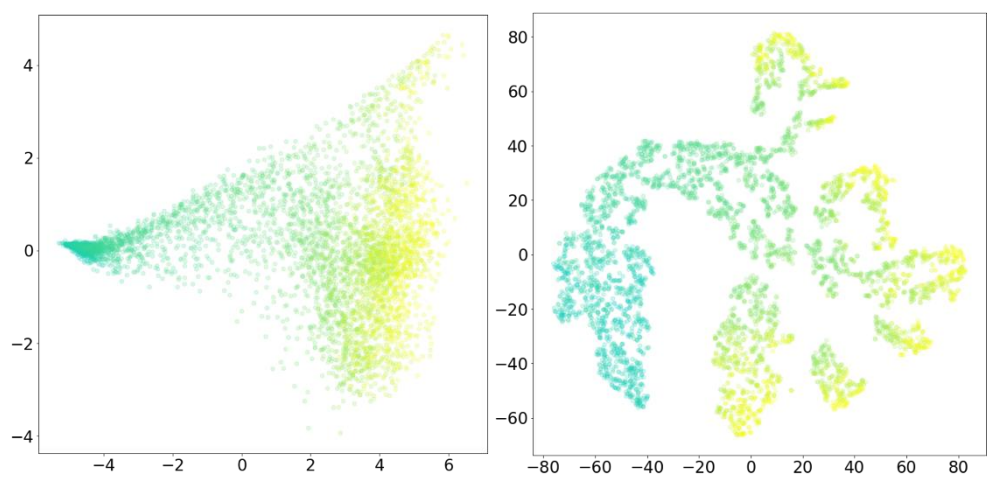
(a) PCA (b) t-SNE

Figure 4-7 Latent space visualization results for an increasing modulation frequency



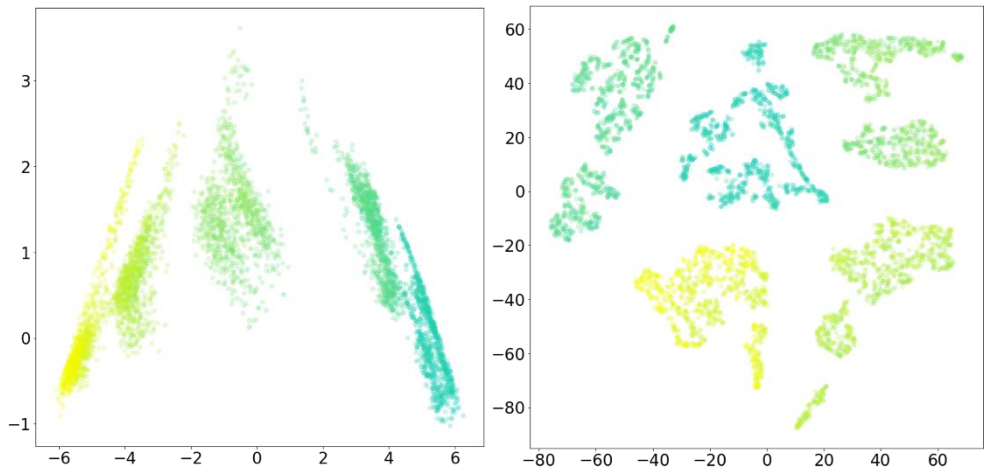
(a) PCA (b) t-SNE

Figure 4-8 Latent space visualization results for an increasing modulation pitch



(a) PCA (b) t-SNE

Figure 4-9 Latent space visualization results for an increasing impulse amplitude



(a) PCA (b) t-SNE

Figure 4-10 Latent space visualization results for an increasing number of impulses

4.2 Case Western Reserve University Bearing Dataset

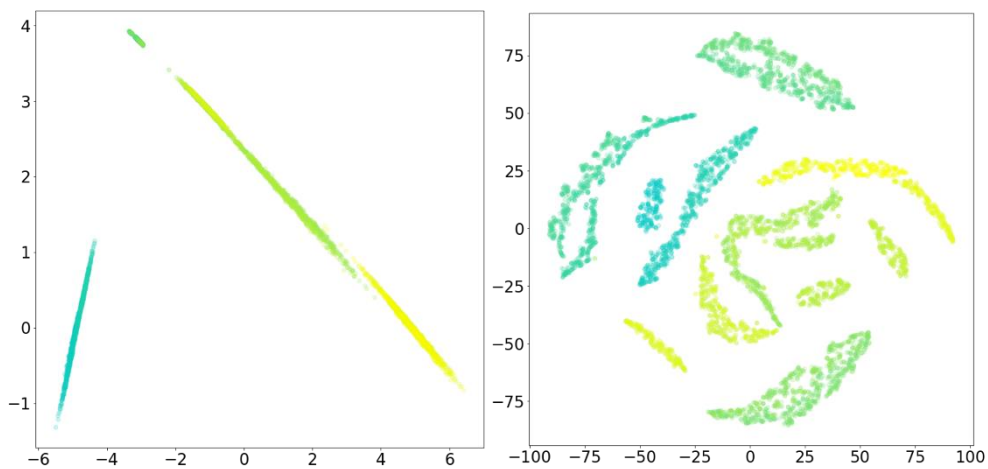
The proposed method was also tested on real data. Case Western Reserve University's (CWRU) bearing dataset [25] is provided by the CWRU Bearing Data Center. Since the original source is acceleration, the data was converted to an acoustic signal and acoustic faults were labeled subjectively.

4.2.1 Latent Space Visualization Results

Datasets were ordered green to yellow based on the following order: normal baseline data measured from the drive-end, 12k fan end bearing fault data (inner race fault size 7, 14, 21 mils) measured from the fan-end, and12k drive end bearing fault data (outer race fault size 21 mils at 6, 3, 12 o' clock direction) measured from the drive-end.

In Figure 4-11 (a), latent features are mapped into two lines. Normal features are mapped along the left line, and features from the fault are lined up on the right line. It seems that fault severity increases as features move to the right, yet there are some exceptions.

Figure 4-11 (b) shows t-SNE formed disconnected clusters.



(a) PCA (b) t-SNE

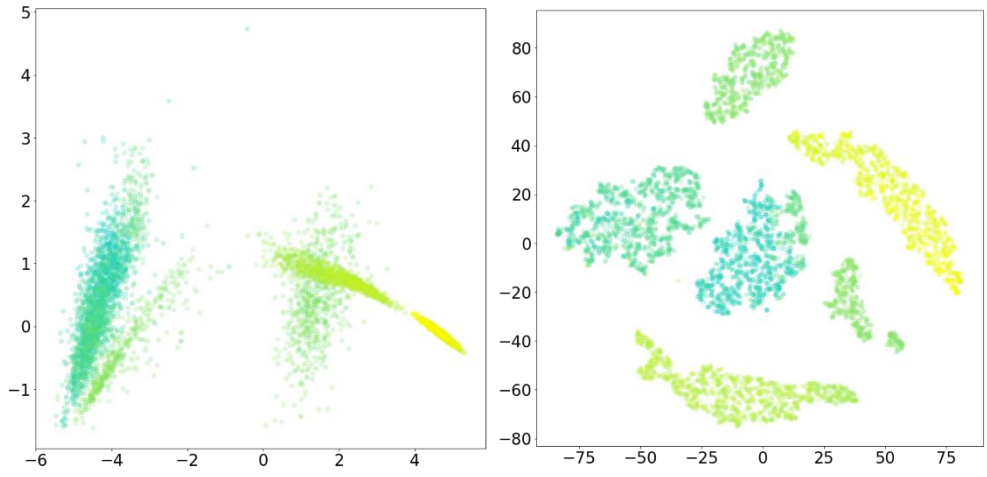
Figure 4-11 Latent space visualization results for the CWRU bearing dataset

4.3 Unbalanced Fan Data

To test with more objective datasets, the sound of an ordinary fan rotating at the same speed with different a mass attached was recorded. Sound was recorded from five different amounts of mass attached to the fan. Since the rotating speed is fixed, and the level of vibration amplitude increases as the mass increases, this phenomenon can be viewed as a realistic case of an impulsive fault with increasing amplitude.

4.3.1 Latent Space Visualization Results

Figure 4-12 (a) shows that it is possible to visualize the tendency that a faulty acoustic signal moves toward the right in the proposed scheme. Figure 4-12 (b) shows an undesirable result, which is expected from Figure 4-10 (b). Since the mass added to the fan was discrete, t-SNE could not form a continuous cluster.



(a) PCA (b) t-SNE

Figure 4-12 Latent space visualization results for the fan dataset

Chapter 5. Conclusions and Future Work

5.1 Conclusions

This research proposed a procedure to establish an estimation method for intermediate faults in acoustic data with limited label conditions. It was shown that latent space features extracted from suggested models can be mapped smoothly between normal clusters and severe fault clusters. Intermediate faults can then be estimated using these features in a reduced feature space, roughly based on the distances of these features from the normal cluster and the severe fault cluster. Through a case study, it was shown that the proposed method can also be applied to real acoustic data.

It is expected that the proposed model can be applied to various industrial sectors. The proposed method can be used to visualize the acoustic quality of manufactured products, making trend monitoring of acoustic quality possible. This method can be applied not only to inspection lines, but also to condition monitoring of equipment, such as a bearing. There should be further consideration of how this method can be applied in situations where there is more than one type of fault.

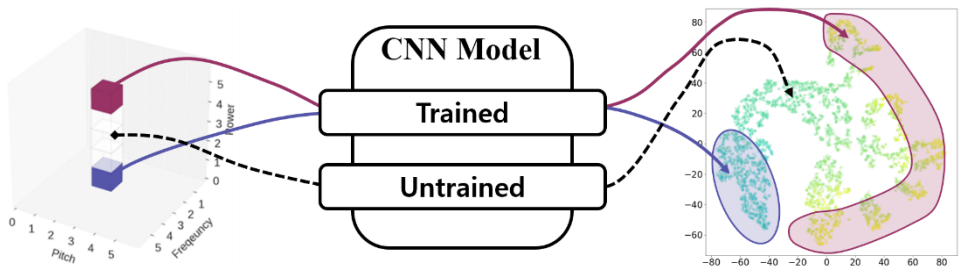


Figure 5-1 Concept of the proposed method

5.2 Contributions

There are two primary contributions of this study.

First, this research verified that the feature extractor of a CNN network can be trained to capture information of the input patterns. The feature extractor was also able to capture intermediate characteristics that have never previously been trained, as shown in Section 4.1.3. CNN models are not expected classify unknown data correctly. Still, our result shows that the model seems to know what the most similar class is and how close it is. In the conventional approach, this knowledge is forgotten during the classification step.

Second, this research shows that this phenomenon is not only limited to artificial data but also to real data, as outlined in Section 4.2 and Section 4.3. Even though features are not continuous for real data cases, mapping of each feature seems to express the relationship between the inputs.

5.3 Future Work

The results of this research leave several questions that can be considered in future work. Most importantly, more precisely labeled data from an industrial environment should be tested to prove that the suggested method is truly useful. For automated quality examination to be realized, high accuracy (e.g., above 99.9%) is required.

Second, the meaning of the latent features must be clearly understood. Why do features form a curvy line instead of a straight one? How many features are involved in explanation of a target parameter? What other characteristics are captured? After a thorough understanding of these issues is reached, a health index that scores the status of the input data should be derived.

Third, verification of what happens if two fault phenomena coexist is required. In realistic cases, amplitude, frequency, and pitch can all change together. Can the proposed model successfully capture all three characteristics at once? Maybe the model needs to consider all three cases separately for accurate learning.

Finally, additional progress in deep-learning methods is expected. The proposed method is somewhat different from deep learning. The model is trained the same, but fully connected layers are abandoned in the visualization step. This is because the purpose of the prosed model is closer to regression than classification. If a deep-learning structure for regression can be devised, the data can be trained and tested with the same structure.

Bibliography

[1] Ravikumar, S., K. I. Ramachandran, and V. Sugumaran. "Machine learning approach for automated visual inspection of machine components." Expert systems with applications, 38.4, 2011.

[2] Cotton, Courtenay V., and Daniel PW Ellis. "Spectral vs. spectro-temporal features for acoustic event detection." Applications of Signal Processing to Audio and Acoustics (WASPAA), IEEE, 2011.

[3] Willemsen, Andrew M., and Mohan D. Rao. "Characterization of sound quality of impulsive sounds using loudness based metric" Proceedings of 20th International Congress on Acoustics, Sydney. Vol. 5, 2010.

[4] Volandri, Gaia, et al. "A psychoacoustic approach for sound quality assessment of automotive power windows." Proceedings of ISMA2012, Katholieke Universiteit Leuven, 2012.

[5] Chu, Selina, Shrikanth Narayanan, and C-C. Jay Kuo. "Environmental sound recognition with time–frequency audio features." IEEE Transactions on Audio, Speech, and Language Processing 17.6, 2009.

[6] Dennis, Jonathan William. "Sound event recognition in unstructured environments using spectrogram image processing." Nanyang Technological University, Singapore, 2014.

[7] Lee, Honglak, et al. "Unsupervised feature learning for audio classification using convolutional deep belief networks." Advances in neural information processing systems, 2009.

[8] Espi, Miquel, et al. "Exploiting spectro-temporal locality in deep learning based acoustic event detection." EURASIP Journal on Audio, Speech, and Music Processing, 26.1, 2015.

[9] Stowell, Dan, et al. "Detection and classification of acoustic scenes and events." IEEE Transactions on Multimedia, 17.10, 2015.

[10] Piczak, Karol J. "Environmental sound classification with convolutional neural networks." Machine Learning for Signal Processing (MLSP), 2015 IEEE 25th International Workshop on. IEEE, 2015.

[11] Takahashi, Naoya, et al. "Deep convolutional neural networks and data augmentation for acoustic event detection." arXiv preprint arXiv:1604.07160, 2016.

[12] Salamon, Justin, and Juan Pablo Bello. "Deep convolutional neural networks and data augmentation for environmental sound classification." IEEE Signal Processing Letters, 24.3, 2017.

[13] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." Advances in neural information processing systems, 2012.

[14] Chollet, F. keras, GitHub. https://github.com/fchollet/keras, 2015.

[15] Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." arXiv preprint arXiv:1409.1556, 2014.

[16] He, Kaiming, et al. "Deep residual learning for image recognition." Proceedings of the IEEE conference on computer vision and pattern recognition, 2016.

[17] Szegedy, Christian, et al. "Going deeper with convolutions." Proceedings of the IEEE conference on computer vision and pattern recognition, 2015.

[18] Szegedy, Christian, et al. "Rethinking the inception architecture for computer vision." Proceedings of the IEEE conference on computer vision and pattern recognition, 2016.

[19] Chollet, François. "Xception: Deep learning with depthwise separable convolutions." arXiv preprint: 1610-02357, 2017.

[20] Kim, Seunghyeon, et al. "Transfer learning for automated optical inspection." Neural Networks (IJCNN), 2017 International Joint Conference on. IEEE, 2017.

[21] Radford, Alec, Luke Metz, and Soumith Chintala. "Unsupervised representation learning with deep convolutional generative adversarial networks." arXiv preprint arXiv:1511.06434, 2015.

[22] Maaten, Laurens van der, and Geoffrey Hinton. "Visualizing data using t-SNE." Journal of machine learning research 9, 2008.

[23] Hunter, John D. "Matplotlib: A 2D graphics environment." Computing in science & engineering 9.3, 2007.

[24] Abadi, M., et al. "TensorFlow: large-scale machine learning on heterogeneous distributed systems." arXiv preprint arXiv:1603.04467, 2016.

[25] Case Western Reserve University. “Seeded Fault Test Data”. Retrieved form http://csegroups.case.edu/bearingdatacenter/

국문 초록

이 연구는 극단적인 정상과 이상 음향 신호만을 학습하여, 임의의 음향 신호의 이상 정도를 추정할 수 있는 딥러닝 알고리즘 기반 방법론에 대한 것이다. 우선 연속적으로 강도가 변화하는 이상 음향을 구현하기 위해 두 종류의 이상 신호를 실험적으로 합성하였다. 정상과 심한 이상 음향을 스펙트로그램으로 변환하여 다른 데이터로 이미 가중치가 학습된 CNN 모델로 분류를 시도한 결과, 아주 높은 수준의 정확도로 분류가 가능한 것이 확인되었다. 그러나 이 과정에서 학습된 모델로도 중간 정도의 이상 음향을 구분해낼 수 없었다. 이 한계점을 극복하기 위해서 우리는 잠재 공간의 특징 인자를 추출하였다. 우리는 특징 인자의 차원을 축소한 결과, 이상 정도의 증가에 따라 차원 축소된 인자 값이 서서히 변하는 현상을 관찰하였다. 이 현상은 정상 상태와 이상 상태의 특징 인자 군집 사이에 중간 정도의 이상을 가진 음향의 특징 인자를 위치시킬 수 있음을 시사한다. 마지막으로 이 방법론은 비음향 진동 데이터를 포함한 실제 환경에서 계측된 데이터들에 적용되었다. 제시된 방법론은 실제 데이터에 대해서도 유의미한 결과를 보였으며, 시주파수 영역 상에서 상태가 변화하는 이상에 공통적으로 적용될 수 있음을 제시하였다.

주요어: 음향 이상
딥 러닝
스펙트로그램
이상 분류

이상 추정

학 번: 2017-22983