공학석사학위논문

# 건전성 예측 및 관리 기술을 위한 해석가능 자동 딥러닝 프레임워크 개발

**An Automated Interpretable Deep Learning (AutoIDL) Framework Development for Prognostics and Health Management (PHM)**

**2020 년 2 월**


서울대학교 대학원

기계항공공학부

공 현 배

# Abstract

## An Automated Interpretable Deep Learning (AutoIDL) Framework Development for Prognostics and Health Management (PHM)

Hyeon Bae Kong
Department of Mechanical and Aerospace Engineering
The Graduate School
Seoul National University

This paper proposes an automated interpretable deep learning framework. The proposed method consists of two steps. First step is to optimize the neural network automatically by defining the neural architecture hyper-parameters with pre-trained model. By using pre-trained model and Bayesian optimization based neural architecture search, we can take advantages of the two methodologies. Second step is to make the existing deep learning model interpretable. The second step is divided into explaining the reason for the prediction of the individual data and estimating how confident prediction is. First step is a method to give analytical power in the time-frequency domain, which is mainly user for fault diagnosis of mechanical system. Second step is that predictive uncertainty is estimated through deep ensemble methods. Proposed method is validated under noisy environment and different load cases using ball bearing data. In addition, proposed method can be easily applied to the various domain.

**Keywords:** Prognostics and Health Management
Fault Diagnosis
Automated Machine Learning
Interpretable Machine Learning

**Student Number:** 2018-20405

# Table of Contents

# List of Tables

# List of Figures

# Chapter 1. Introduction

## 1.1 Motivation

Prognostics and Health Management (PHM) is a study aimed at minimizing economic losses by securing the reliability and safety of many industrial components. In order to ensure the stability of industrial components, PHM uses rule-based, physics-based, data-driven, and hybrid approaches according to the number of data and domain knowledge.

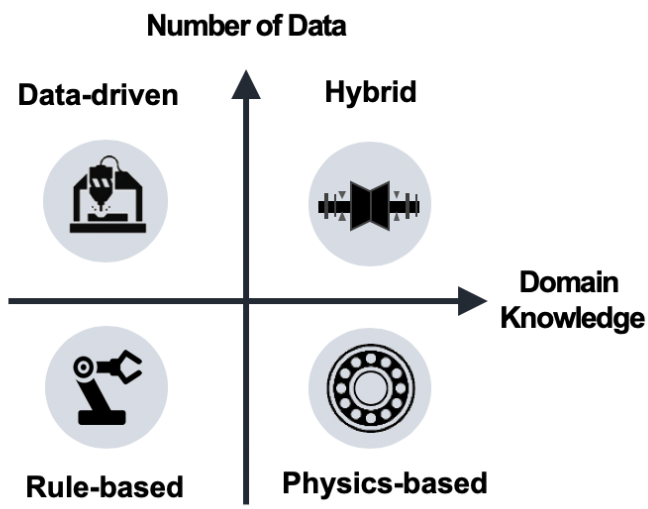



**Figure 1-1. Standard PHM Approaches**

PHM performs fault diagnosis and prognosis using various information such as sensors and physical model of industrial component. This method allows the user to know the status of small components of a large system and plays a secondary role in the administrator's health state decision. However, data-driven approaches are widely

used because complex industrial systems have a difficult understanding of physical behavior. In this paper, we focused on data-driven approaches. In addition, empirical data is needed to predict health conditions such as Remaining Useful Life (RUL) of industrial components. However, as there is no data available to perform generalizations over changes in lifespan, we likewise focused on fault diagnosis in this paper, PHM. However, if the data is sufficient, the proposed methodology is equally applicable.

Therefore, typical data-driven based PHM frameworks typically proceed in the order of data acquisition, signal processing, feature extraction, feature selection, and fault diagnosis. Until the emergence of deep learning, many studies were performed to increase the performance of PHM by performing various signal processing, feature extraction, and feature selection in stages. The disadvantage, however, is that this approach depends on the domain knowledge and intuition of the engineer working with the industrial component. For example, effective and optimal feature engineering to express kinetic energy, data statistics, and waveforms as time domain features has been studied for the journal bearing rotor system case. That is, the PHM framework has a system dependent problem because the feature engineering has to be newly performed every time the application changes. To solve this problem, we need a framework that can apply consistent algorithms regardless of industrial components. The methodology that solves this problem is deep learning. Unlike conventional approaches, deep learning framework is an end-to-end based methodology that performs fault diagnosis using only given data. Therefore, the system dependency problem, which is a problem of the conventional methodology, is solved, and there is an advantage that a consistent methodology is applicable to all industrial components. In addition, the existing methodology has the disadvantage

that each step of feature engineering does not directly affect the performance of troubleshooting. However, since deep learning autonomously performs feature learning to maximize the accuracy of troubleshooting, the performance of troubleshooting is very reliable and robust compared to the existing methodology. Through this methodology, we can see that we have achieved very high accuracy through Deep Learning in Large Scale Visual Recognition Challenge (ILSVRC) 2017 as shown below. Since then, as various Neural Architectures have been developed, it can be seen that the performance of human recognition is higher than 5%. However, deep learning, a new framework for layering, does not have its advantages.

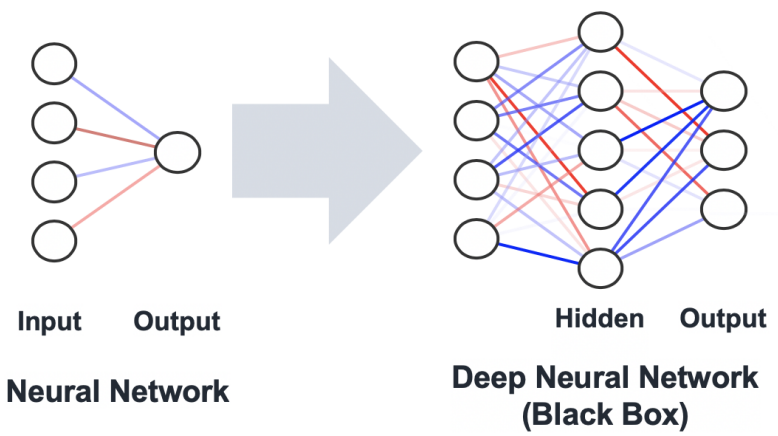



**Figure 1-2. Disadvantage of Deep Learning**

## 1.2 Scope of Research

This thesis proposes automated interpretable deep learning framework. The research scope of the proposed method is as below.

1) Since deep learning is a black box model that users cannot interpret, there is a disadvantage that you have to manually select or search the model composition and learning method. Paradoxically, the problem was that deep learning was used to eliminate existing domain knowledge and system dependencies, but a new hyper-parameter dependency was introduced to learn deep learning, resulting in a hyper-parameter dependency problem. In other words, the user can be independent of the algorithm by solving this hyper-parameter dependency problem. However, because hyperparameters are dependent on data and models, there is no golden rule in the selection of hyperparameters, and optimization must always be performed. The proposed method is robust against the parameter variations and unmodeled dynamics.
2) Training hyper-parameter, model hyper-parameter Optimization requires a lot of hyper-parameters, so the search space is very large. In addition, deep learning must be learned to find hyper-parameters that are optimized for the search space, and scratch learning must be performed.
3) There is a problem that deep learning is more difficult to interpret than the existing methodology. In the case of Logistic Regression, a neural network with the most representative layer among the existing methodologies, both the global interpretability, which is interpreted according to the size and the sign of weight, and the local interpretability, which is interpreted according to how activated the given data is obtained, are obtained. Can be. However, the disadvantage is that this interpretation power is lost when two layers become two. Therefore, Deep Learning has a problem called Black Box because it has good performance but cannot be interpreted. In particular, the PHM must identify the cause of the fault diagnosis as well as perform fault

diagnosis of the industrial component. This is because not only can there be huge economic losses in case of a diagnosis, but there can also be human losses. Therefore, interpretation power is essential for effective decision support for users.

## 1.3 Thesis Layout

The reminder of this paper is organized as follows: A brief introduction to Deep Learning, Automated Machine Learning (AutoML) and interpretability are described in Section 2. The proposed automated interpretable deep learning is introduced in Section 3. Different loading condition and noisy environment experiments are conducted to evaluate out method against some other common deep learning-based model. After this, discussion about the results and interpretability of this experiments is presented in Section 4. We conclude this research and present the future work in Section 5.

# Chapter 2. Literature Review

## 2.1 Overview of Deep Learning

Since deep learning has been developed, variant deep neural networks have been developed, and have been utilized in various domains such as various domains, for example, computer vision, natural language processing and speech recognition [1]. This is because deep neural networks can universally approximate all functions using only given data. Deep learning is widely used in the PHM domain, and utilizes many data types such as vibration signals and physical state images. In particular, the Convolutional Neural Network (CNN) is mainly used in the PHM field because it mainly uses signals and images. This section will briefly introduce the Deep Neural Network and Convolutional Neural Network.

### 2.1.1 Overview of Deep Neural Network

Deep neural networks (DNN), also called Multi Layer Perceptrons (MLP), feedforward neural networks, or deep feedforward networks, are deep learning models. The goal of a deep neural network is to approximate any function $f^*$. Deep neural network can approximate regression task and classification task. For example, regression task maps $y = f_\theta(x)$ to a continuous value $y$ and classification $y = f_\theta(x)$ to a discrete value also called labels. And, according to the Generalized Linear Model, the loss function and activation function are determined. Classification tasks typically use sigmoid activation and softmax activation functions, cross entropy loss, and regression tasks typically use linear activation function and mean square error loss.

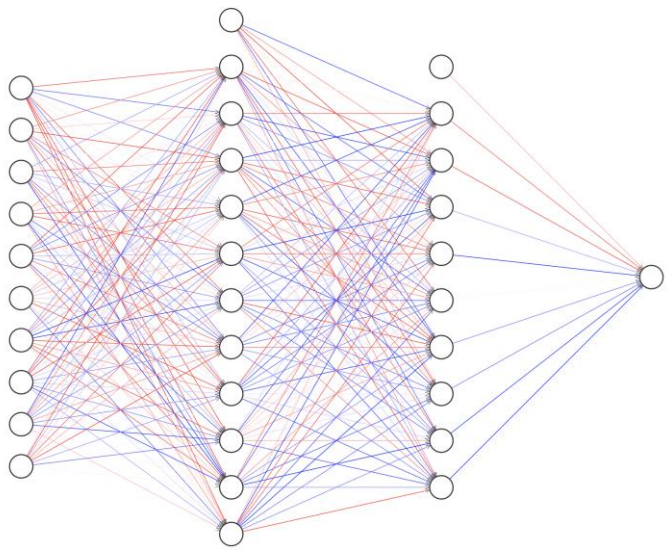


**Figure 2-1. Scheme of Deep Neural Network**

Deep neural networks are basic structures that form the basis of convolutional neural networks and recurrent neural networks, and are very powerful universal approximators. Deep neural networks are also graph structures, which are directed acyclic graphs of dense layers. For example, in the figure below, $f$ consists of first layer $f^{(1)}$, second layer $f^{(2)}$, and third layer $f^{(3)}$, and $f(x) = f^{(3)}(f^{(2)}(f^{(1)}(x)))$ where dense layer is directly connected. Thus, when an input is given, to output the output, the middle layer's output is automatically trained, so it is called a hidden layer because we do not get the desired output. The deeper the layer is, the more you can approximate any function, but the disadvantage is more parameters, and more room for overfitting, depending on the curse of the dimension. In addition to the depth of the layers, it is important to map the functions to the widths of the dimensions of each layer. Finally, each layer requires an activation function for non-linear mapping. Depth is determined by how many layers are determined,

width is determined by how many nodes are determined, and the activation function of the layer is then complicated deep neural network based function approximator is composed. Finally, the back-propagation algorithm makes it easy to compute gradients of analytically all weights and biases.

### 2.1.2 Overview of Convolutional Neural Network

Convolutional Neural Network (CNN), also known as Convolutional Networks, is a neural network optimized to handle grid-like topologies. For example, it is a structure that works well for both 2-D grid image of pixels and 3-D grid data of voxels like spectrogram as well as time-series based sensor data which is 1-D grid in PHM domain. Already, it works well for various fields and domains, and it is a structure that is widely used in PHM field.

#### 2.1.2.1. Convolutional Layer

Convolutional Neural Network is composed of neural network used in Convolutional Layer, Pooling Layer and Deep Neural Network. The convolutional layer is the same principle mathematically as the convolution operation used mainly in signal processing. However, since we are learning a convolution filter that maps the input data well, even if we use the cross correlation operation without flipping the kernel instead of the convolution operation, it performs the same function mathematically. To perform. In many, deep learning frameworks, the convolutional layer is based on cross correlation operations.

$$\boldsymbol{S(i,j) = (K * I)(i,j) = \sum\sum I(i+m, j+n)K(m,n)} \qquad (2\text{-}2)$$

In order to map a given input to an output through a convolution operation, we perform a convolution operation on all **(*i, j*)** position-independently using the parameter-sharing feature, and learn the filter to be highly activated pattern matching. You can set the size and number of convolution filters, as well as the size of the padding.

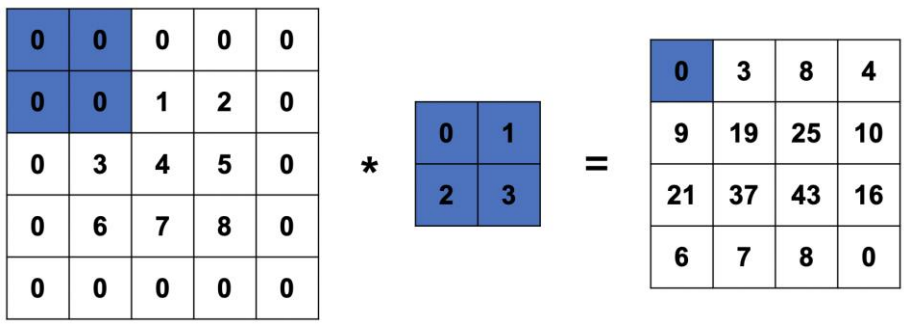



**Figure 2-3. Processing of Convolution Operation**

Input shape is $\boldsymbol{n_h \times n_w}$, padding size is $\boldsymbol{p_h \times p_w}$, convolution kernel size is $\boldsymbol{k_h \times k_w}$, then output shape is $\boldsymbol{(n_h - k_h + p_h + 1) \times (n_w - k_w + p_w + 1)}$.

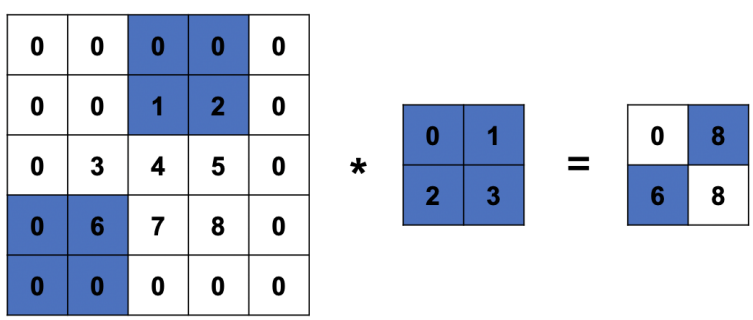



**Figure 2-4. Processing of Stride Operation**

Generally, default value of stride size is 1. However, sometimes, for down sampling

to save the computational efficiency and higher receptive field. This figures cross-correlation with strides of $(\mathbf{3} \times \mathbf{2})$ for height and width. Input shape is $\boldsymbol{n_h} \times \boldsymbol{n_w}$, padding size is $\boldsymbol{p_h} \times \boldsymbol{p_w}$, convolution kernel size is $\boldsymbol{k_h} \times \boldsymbol{k_w}$, stride size is $\boldsymbol{s_h} \times \boldsymbol{s_w}$, then output shape is $|(\boldsymbol{n_h} - \boldsymbol{k_h} + \boldsymbol{p_h} + \mathbf{1})/\boldsymbol{s_h}| \times |(\boldsymbol{n_w} - \boldsymbol{k_w} + \boldsymbol{p_w} + \mathbf{1})/\boldsymbol{s_w}|$ . Padding and stride can be used to adjust the shape of data dimension effectively. Since these hyper-parameters of convolutional layer directly affect performance, choosing a hyper-parameter is very important.

## 2.1.2.2. Pooling Layer

Pooling layer reduces spatial dimensions of hidden representation and aggregate the sensitive spatial information. Since invariant representation learning is possible for these Pooling layers translations, slightly shifted input images have the same hidden activation. The pooling layer includes a max pooling layer and an average pooling layer. Max pooling brings highly activated features by moving the pooling window. For example, when stride size is 1, as shown in the figure above, $\boldsymbol{max}(\mathbf{0}, \mathbf{1}, \mathbf{3}, \mathbf{4}) = \mathbf{4}$, and the same is done for the remaining pooling operations. Average pooling is avg $\boldsymbol{avg}(\mathbf{0}, \mathbf{1}, \mathbf{3}, \mathbf{4}) = \mathbf{2}$ similar to the Max pooling operation, and the rest is similar. In general, max pooling, which brings highly activated feature maps, is mainly used, and effectively reduces resolution, so that overfitting can be avoided, so the convolutional layer and pooling are used alternately. Unlike the convolutional layer, the pooling layer is a discrete operation because there is no learnable parameter, and the number of parameters does not increase, but the disadvantage is that the spatial information loss is large. This problem, especially in generative adversarial networks that need to generate a clear image, has solved this problem by trying to improve image clarity using stride convolution even if the number of parameters increases.

Max pooling is most commonly used for classification tasks, such as the fault diagnosis task of PHM.

## 2.2 Types of Convolutional Neural Network

Convolutional Neural Network is composed of Convolutional layer, Pooling layer, and dense layer. The figure and notation shown above are LeNet5 notation, and LeNet is a convolutional network originally developed by Yann Lecun and developed to recognize handwritten digits. This model is the methodology adopted for ATM machines, and still some ATM machines are driven by this methodology. A convolutional network consists of a feature extraction part and a part that performs a task. The feature extraction part repeatedly accumulates convolution blocks consisting of convolutional layers and pooling, and performs hierarchical feature learning from low level features such as edges and colors to high level features such as faces, depending on the depth of the layer. However, due to the deep neural network, there is a problem that the learning is difficult due to the vanishing gradient problem.

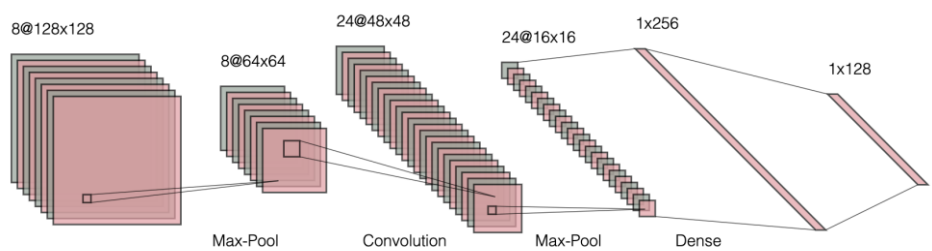



**Figure 2-5. LeNet Types Figures and Notation**

Convolutional networks cannot be used due to vanishing gradient problems, and methodologies such as classical feature-based support vector machines have been widely used. The 2010, 2011 winning models of the Large Scale Visual Recognition Challenge (ILSVRC) are conventional model-based methodologies rather than convolutional neural networks. However, since AlexNet, where the 2012 ILSVRC model solved the vanishing gradient problem using the Rectified Linear Unit (ReLU), deep learning has been widely used in image recognition, as well as speech recognition and PHM.

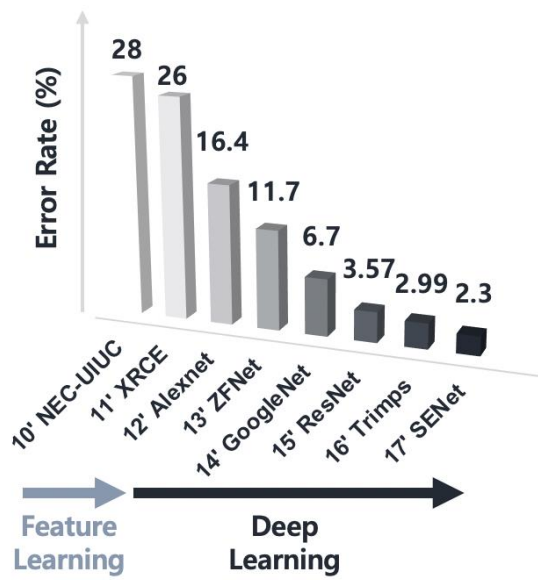



**Figure 2-6. ILSVRC performance graph**

## 2.3 Neural Architecture Search

Neural Architecture Search is an area of AutoML, which is about automating machine learning. When it comes to solving problems with machine learning, it aims

to automate the process, and hyperparameter search is a typical case. Since deep learning has been in place, the need for a way to automatically find network structures, rather than simply optimizing hyperparameters, has grown. This is the latest Neural Architecture Search (NAS) [2]. AutoML, including NAS, is an optimization problem [3], and generally has validation accuracy and loss as an objective function, and validation accuracy is non-differential, so the network is end-to-end and the results are checked repeatedly. As a result, huge computational costs are required.

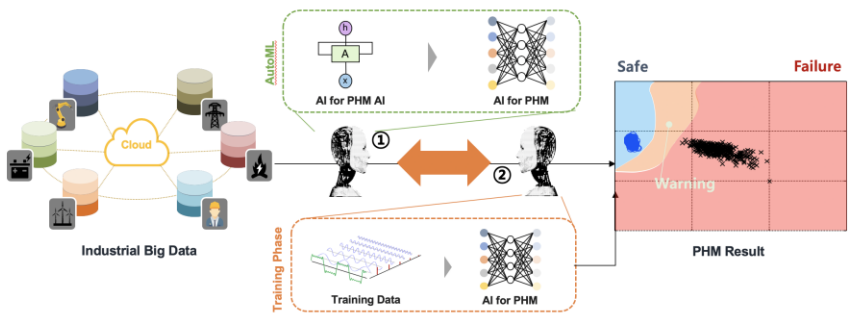



**Figure 2-7. Scheme of Automated Machine Learning**

When performing the existing optimization, many algorithms are used in Neural Architecture Search method, and various methodologies such as reinforcement learning, evolutionary algorithm, Bayesian optimization, and gradient descent method are used [3,4,5].Most methodologies define network generators, and network generators, also called generators, sample child networks to predict performance [6,7]. In addition, the sampled network is trained to have higher performance by learning the sampled child network. Using the neural architecture search methodology, manual search can be used to eliminate unnecessary user intervention. In addition, research continues to show that higher performance is achieved than

conventional methods. However, the problem with this methodology is that there is not much room for improvement if the given data is not diverse, by optimizing the neural network. It is also considered very inefficient because it starts with scratches and optimizes the neural network. This methodology is not well used in schemes such as online learning where continuous data is obtained, but it is mainly used when ILSVRC requires some high performance. Therefore, effective Neural Architecture Search is essential [8,9,10].

## 2.4 Transfer Learning

Transfer Learning is information transfer from the source domain to the target domain in order to perform effective learning. Due to the development of internet technology, it is easy to collect big data such as over 1 million natural images like ImageNet. Therefore, even if the target domain we want to learn is different from the source domain, it is very important to utilize the knowledge of the source domain. There are various methods of transfer learning, but recently, as the deep learning technology that autonomously learns the characteristics of big data using big data has become popular, parameter transfer learning technology that can easily use source domain knowledge has been widely used. Since the convolutional neural network performs general feature learning in the source domain, this technique extracts high level features from the raw input image with the fixed parameters. And the model connects the fully connected layer to perform the task. If the target domain is similar to the source domain, performance is maximized, so other tasks such as image detection and segmentation can be used efficiently. When the source domain is different from the target domain, domain adaptation techniques are often used. Prediction of fault classes using our target domain, the spectral domain image, is very

different from the source domain, but it is like starting with a relatively good initial point. In addition, since there is no need to learn convolutional neural networks for feature learning, the trained parameters are dramatically reduced. Therefore, the framework of the target task can be performed relatively quickly with good performance without falling into the curse of dimension. Various networks can be used as a pretrained network.

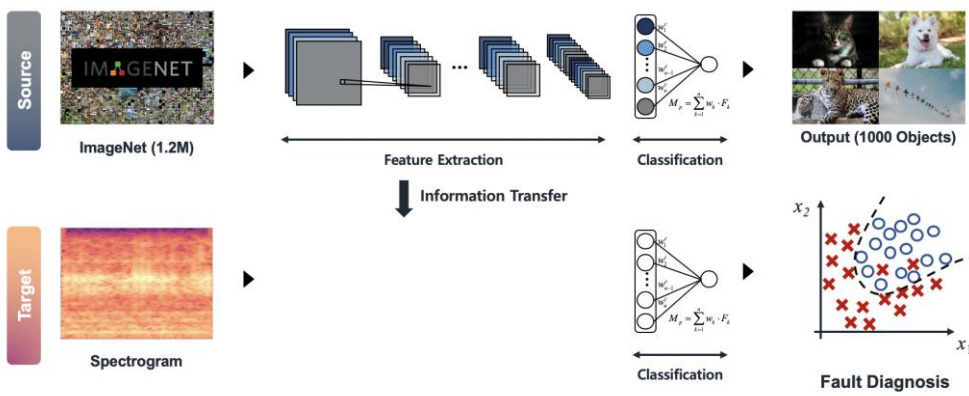



**Figure 2-8. Scheme of Transfer Learning**

## 2.5 Interpretability

Deep learning is generally considered a black-box model. However, many fields use deep learning, and a lot of interpretability research is being done to find the physical meaning. In general, interpretability includes (1) global interpretability, (2) local interpretability, and (3) uncertainty estimation [11]. (1) The global interpretability method describes how the model predicts overall. In general, the weight of a feature indicates its feature importance. A method like LASSO performs feature selection based on feature importance. However, due to problems such as

multiple collinearity or correlation, weight values can be obtained differently from physical understanding. To solve this problem, it has to go through additional processing such as removing highly correlated features. It is also easy to enforce monotonicity on the relationship between input features and outputs, to prevent common sense and deviations, and to explain the overall operation of the model. Recently, research has been continued to add monotonic conditions to nonlinear deep neural networks. (2) Local interpretability is an interpretation method in which the model explains the reason for the prediction for individual data points. Among local interpretability methods, there are largely local surrogate based method, counterfactual explanation, and gradient-based explanation methodology. The local surrogate methodology creates a locally approximated linear model for an interpretable model-agnostic explanation. Counterfactual explanation is a way of interpreting how results vary by hypothesis, presenting virtual or real data points. Finally, the Gradient-based explanation methodology. Class Activation Map (CAM) [15] was analyzed using linearity between global average pooling and output. However, to solve the limitation that only models using global average pooling are applicable, Grad-CAM (Gradient-weighted Class Activation Map) [16] has been proposed. This visualization technique is a class-discriminative technique and can only be applied to image classification tasks. This methodology is a methodology that can analyze various tasks without any modification to the network and without any performance degradation.

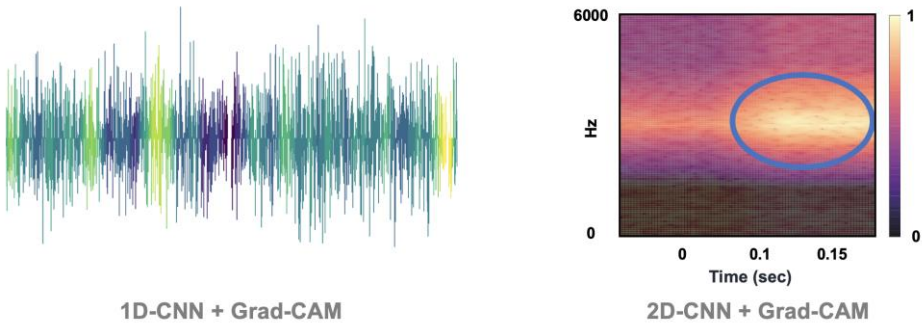



**Figure 2-9. Disadvantage of Gradient-based Local Interpretability Method**

However, as the figure below shows, the feature map is reduced in dimension, resulting in lower resolution. Therefore, a new local interpretability method is needed. Finally, we use the confidence score of the model to perform uncertainty estimation. For example and support vector machines, the farther they are from the decision boundary, the more confident they are. In the case of ensemble methods, the predicted results of each model are high confident and low confident if not. In the case of deep neural networks, the probability is output, so it can be determined based on entropy. Recently, it is common to estimate predictive uncertainty using model uncertainty such as dropout and batch normalization of deep learning model.

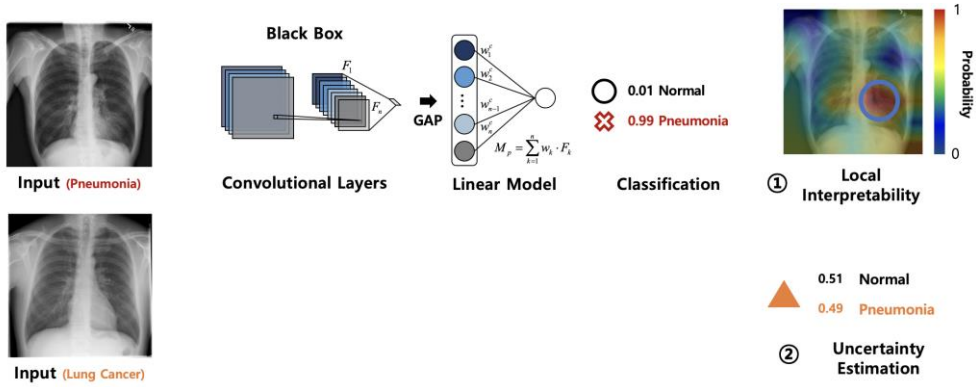



**Figure 2-10. Scheme of Interpretability**

# Chapter 3. Proposed Automated Interpretable Deep Learning (AutoIDL) Framework

The proposed Automated Interpretable Deep Learning (AutoIDL) Framework has the following process. First, after changing to spectrogram, a domain that users can easily interpret, in the second step, we design the search space in consideration of transfer learning, and in the third step, perform neural architecture search based on Bayesian optimization. Finally, the process of giving analysis power to the optimized model.

## 3.1 Preprocessing

Convolutional neural networks are performing fault diagnosis using 1-D vibration. However, since the length of the data is longer compared to the image, a wider receptive field is needed. In addition, noise is distributed over a wide frequency range, requiring deeper, larger kernels and larger stride sizes to find key features for fault diagnosis [12,13].

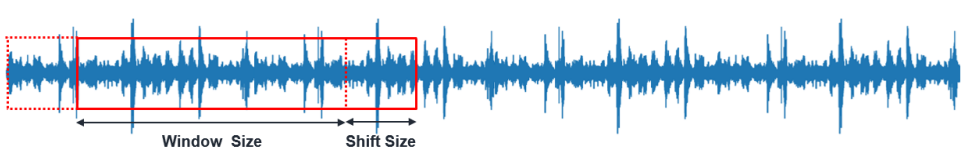



**Figure 3-1. Augmentation of Time series data**

In addition, deep learning models require augmentation techniques because they require a lot of data. In the computer vision domain, the data is augmented by various image transformation techniques such as flipping, cropping, and scaling under

human-recognized conditions even if the deformation is applied. In general, by performing augmentation, general feature representation learning can be performed to avoid overfitting and increase generalization performance. In the PHM domain, since a long 1-D vibration signal is given in the fault diagnosis, generalization of the phase is required. Augmentation is performed using random cropping with a length of 2048 and a shift size of 1 for a given long time sequence. The test step was performed so that there was no overlap in the test step. Although this method is a simple augmentation method, it is known as a method to effectively avoid overfitting.

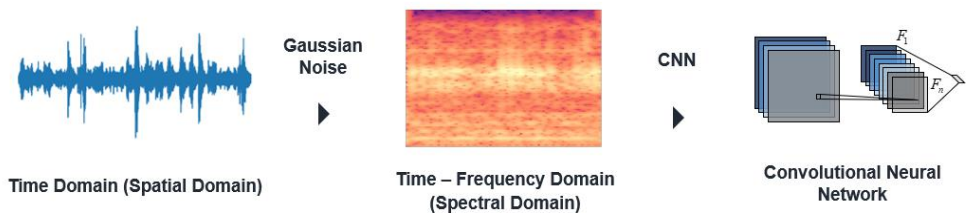



**Figure 3-2. Scheme of Overall Augmentation Process**

In the next step, white gaussian noise is added. By adding noise, a robust algorithm can be built. In Section 4, various levels of noise are added depending on the experimental environment. The next step is to perform a spectrogram 2-D image transformation that is easy for the user to interpret and is already used in troubleshooting. By utilizing image transformation, not only can you effectively use the deep learning techniques used in traditional computer vision, but you can also use models that have already been trained in feature representation, such as transfer learning in other domains.

## 3.2 Automated Neural Architecture Construction

Neural Architecture Search, which finds the optimal neural network among automated machine learning, is continuously researched. However, because it is an inefficient way to learn from scratch, we propose a methodology to use it as a pre-trained model of transfer learning [18]. However, there are many problems with the proposed method. First, because there are a variety of models in the pretrained model, you need to choose which model to choose. In addition, the size of the model must be chosen depending on the complexity of the problem. Second, there is a drawback that the statistics of the data used in the pretrained model differ from the statistics of the PHM domain that we want to solve. In addition, there is a problem of increasing the channel of the input data redundantly. Finally, the signal data is a very important hyperparameter as the resize coefficient of the dimension of the input data is also recently studied in Efficient-Net [19].

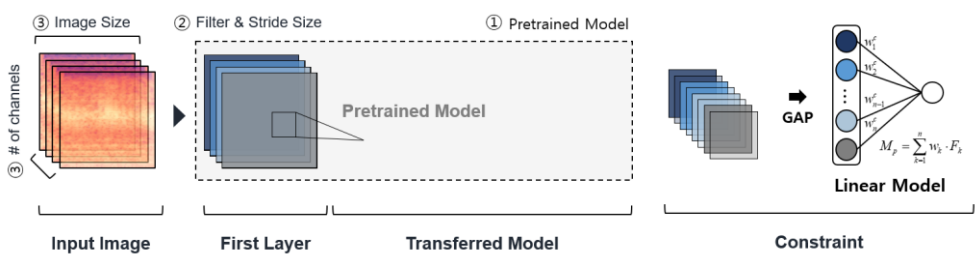



**Figure 3-3. Search Space of Proposed Method**

Therefore, to solve the above problem, we propose the First Layer Optimization with Transfer Learning (FLOTL). First, we define a search space to find which pretrained model is optimal. Here, we chose a model with Global Average Pooling (GAP) like

ResNet. The reason is that because GAP can be approximated with a linear model, it can not only utilize the interpretability technology like the existing Class Activation Map [16], but also can be used with the proposed methodology. Second, optimize the hyperparameters of the first layer, filter size, stride size and number of channels. Finally, we optimize the hyperparameter image resize coefficient and number of channels associated with the input data. Through the proposed method, the neural network can be optimized by using the feature representation learning part of the pretrained model, and the GAP and dense layer are given as constraints, which avoids overfitting and is an interpretable methodology.

## 3.3 Interpretability

In order to give high resolution to the black box deep learning model, we propose a new methodology. This framework focuses on local interpretability and uncertainty estimation techniques. First, the local interpretability method is widely used as a gradient-based explanation method such as Grad-CAM. Due to the nature of the convolutional neural network, resolution has to be reduced. Although there is a way to make the resolution of the feature map constant, the overfitting problem cannot be avoided, and the proposed framework uses a pre-trained model of the transfer learning technique. In addition, the Grad-CAM [17] methodology is a localization-based methodology, which is an algorithm that determines where it is at first, and thus is difficult to interpret precisely. In order to solve this problem, we propose the Frequency Domain Occlusion based Interpretability (FDO) methodology which calculates the importance of the frequency band by using the frequency masking as shown in the figure below. $p(x;\theta) - p(x_{occlusion};\theta)$ means frequency feature importance, occlusion moves with 1 pixel iteratively, and a new image with

resolution like spectrogram can be obtained. The methodology for obtaining the prediction difference of the spectral domain is called Time Domain Occlusion based Interpretability (TDO). The proposed methodology indicates that the band is important if the difference between the probability values is large and that the band is not important if the difference between the probability values is small.

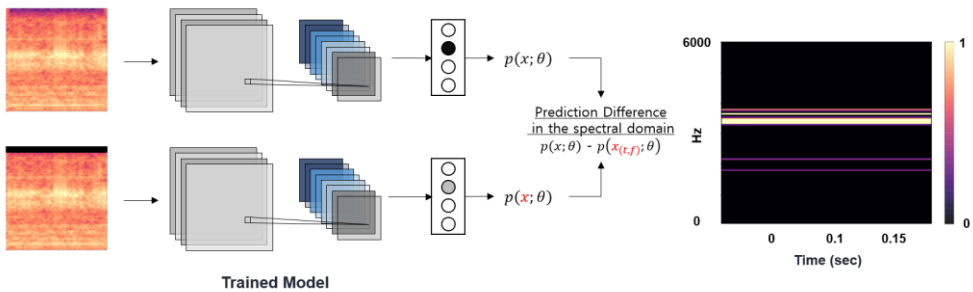



**Figure 3-4. Scheme of Time Domain Occlusion based Interpretability (TDO)**

Second, we estimate the predicted uncertainty by using the uncertainty of the model. The most widely used Bayesian deep learning model is more computation is required because it is accompanied by other learning processes such as variational inference. Also, since the performance is not higher than that of the general neural network, the methodology using the general neural network has been continuously studied. In particular, deep ensemble-based predictive uncertainty estimation is very simple and the uncertainty estimation performance is excellent. However, there is a disadvantage that it is very time consuming because the model must be trained independently. To solve this problem, using the model obtained while performing Bayesian optimization, predictive uncertainty is estimated using the ensemble technique $p(y|x) = \frac{1}{M}\sum_{m=1}^{M} f(y|x;\theta)$. This method can reduce the computation

cost by M times compared to the time methodology, and the performance is similar to the existing model. Also, when certain data is input, it meets low bias and low variance, and when uncertain data is input, it meets high variance.

# Chapter 4. Experiment Results

The chapter 4 deals with experiment results of the proposed method to estimate the efficiency, performance and the robustness.

## 4.1 Data Description

To verify the proposed algorithm, we used Case Western Reserve University (CWRU) bearing dataset with sampling rate of 12000Hz. As shown in the table below, there are normal, ball fault, inner race fault and outer race fault. Since there are 10 classes, the softmax activation function is used, and the length of the raw signal of each data is 2048. For each class, 900 training samples points uses augmentation technique with random cropping and test set has 75 non-overlapping test sample points among the data that are not used in training set. Epoch was trained using 20 and learning rate was 3e-4, based on the model with the highest validation accuracy among epochs.

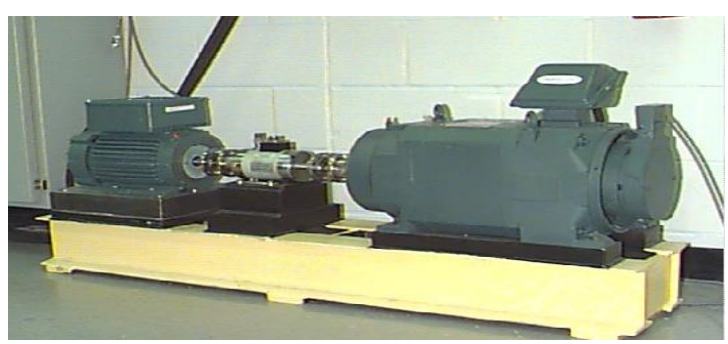


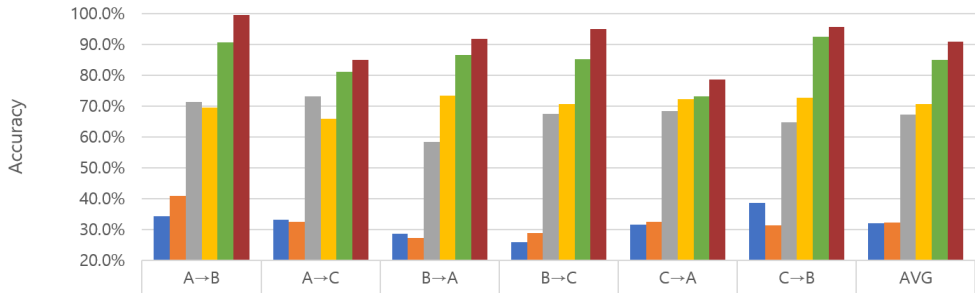
**Figure 4-1. Case Western Reserve University (CWRU) Bearing Dataset**

**Table 1. Description of Bearing Dataset**

| Health state | | None | Ball | | | Inner Race | | | Outer Race | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Fault Diameter (inch) | | 0 | 0.007 | 0.014 | 0.021 | 0.007 | 0.014 | 0.021 | 0.007 | 0.014 | 0.021 |
| Load 0 | Valid | 300 | 300 | 300 | 300 | 300 | 300 | 300 | 300 | 300 | 300 |
| Load 1 | Train | 300 | 300 | 300 | 300 | 300 | 300 | 300 | 300 | 300 | 300 |
| | Test | 25 | 25 | 25 | 25 | 25 | 25 | 25 | 25 | 25 | 25 |
| Load 2 | Train | 300 | 300 | 300 | 300 | 300 | 300 | 300 | 300 | 300 | 300 |
| | Test | 25 | 25 | 25 | 25 | 25 | 25 | 25 | 25 | 25 | 25 |
| Load 3 | Train | 300 | 300 | 300 | 300 | 300 | 300 | 300 | 300 | 300 | 300 |
| | Test | 25 | 25 | 25 | 25 | 25 | 25 | 25 | 25 | 25 | 25 |
| Total | Train | 900 | 900 | 900 | 900 | 900 | 900 | 900 | 900 | 900 | 900 |
| | Test | 75 | 75 | 75 | 75 | 75 | 75 | 75 | 75 | 75 | 75 |

## 4.2 Different Load Cases

We verified that the proposed method works well for other Load Cases. In order to perform validation on the load, for example, training is performed using the data of Load 1, validation is performed using the data of Load 0, the model having the highest validation accuracy is selected, and another load is performed. Perform a test to make sure it works for you. Fast Fourier Transform (FFT), the most widely used method using existing features, obtains 1024 Fourier coefficients, MLP consists of 1024, 1000, 10 nodes, and DNN consists of 1024, 1000, 1000, 10 Use the model to perform a diagnosis. In addition, in order to compare accurate performance, it repeated five times and averaged the result. In the case of 1D-CNN, the training was performed similarly to the existing widely used models, and 2D-CNN used the ResNet-50 model after performing spectrogram image transformation on the raw signal [14]. Much less data was used than previously used data. Therefore, the FFT-based methodology does not work well for different loads because of the high probability of overfitting over a narrow frequency range. Recently, deep learning models using deep learning have different results depending on the case of 1D-CNN and 2D-CNN, and the average of accuracy is very similar [15]. When we use the pretrained model, we can see that the performance increases drastically. To analyze this, we performed the feature visualization as shown below.




| | A→B | A→C | B→A | B→C | C→A | C→B | AVG |
|---|---|---|---|---|---|---|---|
| FFT-MLP | 34.24% | 33.12% | 28.72% | 25.92% | 31.68% | 38.72% | 32.07% |
| FFT-DNN | 40.88% | 32.56% | 27.20% | 28.88% | 32.40% | 31.28% | 32.20% |
| 1D-CNN | 71.28% | 73.20% | 58.40% | 67.60% | 68.40% | 64.87% | 67.29% |
| 2D-ResNet | 69.52% | 65.84% | 73.44% | 70.64% | 72.24% | 72.64% | 70.72% |
| 2D-ResNet-Pre | 90.80% | 81.24% | 86.68% | 85.23% | 73.12% | 92.48% | 84.93% |
| Proposed | 99.52% | 85.12% | 91.76% | 95.03% | 78.60% | 95.60% | 90.94% |

**Figure 4-2. Result of Different Load Cases**

When visualizing raw signals using PCA, it can be seen that they are clustered in the same area that is difficult to classify. However, when spectrogram is used, it can be seen that the relationship according to load can be found to some extent. In other words, the method using the spectrogram is an appropriate method, and when the feature is extracted and visualized using the pretrained model, the pretrained model is classified so that it can be classified without learning the dense layer performing the classification task. It can be very helpful.

Finally, when the pretrained model and the fist layer optimization were performed at the same time, it was confirmed that the performance of fault diagnosis increased a little more, and the performance was estimated to be improved because the problems caused by the existing pretrained model were solved.

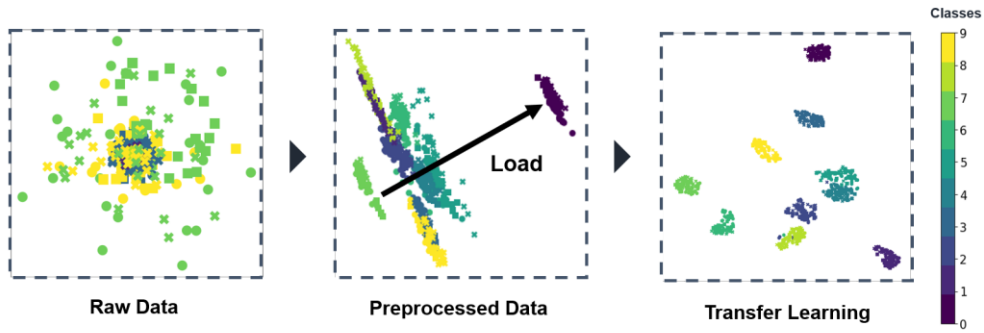
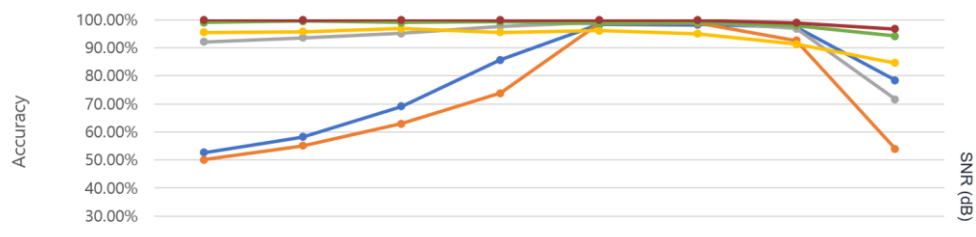



**Figure 4-3. Feature Visualization using PCA and t-SNE**

## 4.3 Noisy Environment Cases

In a real environment, it can be exposed to various noise environments, so we verified that it works well for noisy environment. To perform the validation, we added 0dB of white gaussian noise to the training set. In addition, as a validation set, a model was obtained using a model having high validation accuracy by utilizing data of a time step not used in the training set. Finally, the test set confirmed that it works well with varying levels of white gaussian noise from 10dB to -4dB. As shown in the figure below, the FFT-based method adds 0dB of white gaussian noise to the training set, so it works well at 0dB but does not work well for other noise levels away from 0dB. However, we found that the methodology using the pretrained model works well for various noise levels.




| | 10 | 8 | 6 | 4 | 2 | 0 | -2 | -4 |
|---|---|---|---|---|---|---|---|---|
| FFT-MLP | 52.61% | 58.16% | 69.14% | 85.73% | 98.32% | 98.23% | 97.35% | 78.40% |
| FFT-DNN | 50.08% | 55.06% | 62.96% | 73.78% | 98.58% | 98.77% | 92.58% | 53.98% |
| 1D-CNN | 92.10% | 93.60% | 95.17% | 97.68% | 98.98% | 99.28% | 96.80% | 71.70% |
| 2D-ResNet | 95.46% | 95.73% | 96.80% | 95.46% | 96.13% | 95.06% | 91.20% | 84.66% |
| 2D-ResNet-Pre | 99.06% | 99.46% | 99.06% | 99.20% | 98.66% | 99.06% | 97.86% | 94.13% |
| Proposed | 100.00% | 99.86% | 100.00% | 99.86% | 99.86% | 99.86% | 98.83% | 96.67% |

**Figure 4-4. Result of Noisy Environment Cases**

## 4.4 Interpretability

To verify the proposed FDO methodology, we compared the two models obtained

from the case study. The first model is a 2D-ResNet model with approximately 84% accuracy for -4dB white gaussian noise. Using the FDO methodology, we can see that the probability difference is very large for some frequency ranges. In other words, it can be seen that the model is overfitting over a narrow frequency range. However, even when the proposed model is analyzed using FDO, not only the probability difference is large due to the frequency masking, but also the fault energy of the bearing is in the wide frequency range. In addition, since there is no big probability difference for TDOs, we can see that the proposed occlusion-based analysis methods, FDO and TDO are suitable.

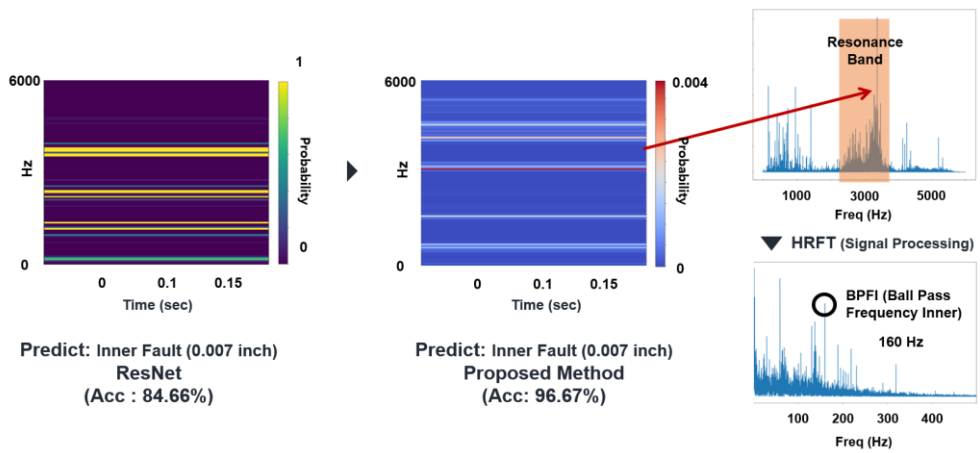



**Figure 4-5. Analysis of Interpretability between Two Models using FDO**

# Chapter 5. Conclusion and Future Work

## 5.1 Conclusion

This paper proposes a new Automated Interpretable Deep Learning Framework (AutoIDL) to solve the fault diagnosis problem and manual search to find the neural architecture and hyper parameter. AutoIDL has three main features, to find optimal first convolutional layer kernel and stride for domain adaptation with input data augmentation and propose a new time and frequency domain interpretability technique and easily achieve the state-of-the-art performance with transfer learning based pretrained model in public CWRU bearing dataset. Results in Section 4 show that proposed AutoIDL is noisy robust without denoising technique when other method suffers from degradation under noisy environment conditions. And, only pretrained model achieved better performance than previous 1-D CNN and 2-D CNN. So, pretrained model is helpful when fault diagnosis is conducted in PHM domain from different domain. And, first layer optimization with transfer learning is helpful to minimize distance between ImageNet source domain and our target domain. In addition, Time Frequency Domain based Interpretability (TDO) is used to investigate the mechanism of black box deep learning model. And, proposed method shows that highest performance without domain adaptation technique compared with previous method for domain adaptation. So, proposed method shows the very simple and robust model.

## 5.2 Future Work

In future work, we apply the domain adaptation technique with Automated Deep Learning. In this research, we don't apply domain adaptation technique such as batch normalization statistics and domain adversarial neural network. This technique is helpful such as different loading case and noisy environment case.

.

# Bibliography


[1] LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton. "Deep learning." nature 521.7553 (2015): 436.

[2] Feurer, Matthias, et al. "Efficient and robust automated machine learning." Advances in neural information processing systems. 2015

[3] Snoek, Jasper, Hugo Larochelle, and Ryan P. Adams. "Practical bayesian optimization of machine learning algorithms." Advances in neural information processing systems. 2012

[4] Baker, Bowen, et al. "Designing neural network architectures using reinforcement learning." arXiv preprint arXiv:1611.02167 (2016).

[5] Pham, Hieu, et al. "Efficient neural architecture search via parameter sharing." arXiv preprint arXiv:1802.03268 (2018).

[6] Desell, Travis. "Large scale evolution of convolutional neural networks using volunteer computing." Proceedings of the Genetic and Evolutionary Computation Conference Companion. ACM, 2017.

[7] Guo, Zichao, et al. "Single path one-shot neural architecture search with uniform sampling." arXiv preprint arXiv:1904.00420 (2019).

[8] Jin, Haifeng, Qingquan Song, and Xia Hu. "Auto-keras: Efficient neural architecture search with network morphism." arXiv preprint arXiv:1806.10282 (2018).

[9] Jin, Haifeng, Qingquan Song, and Xia Hu. "Auto-keras: An efficient neural architecture search system." Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. ACM, 2019.

[10] Liu, Hanxiao, Karen Simonyan, and Yiming Yang. "Darts: Differentiable architecture search." arXiv preprint arXiv:1806.09055 (2018)

[11] Lakshminarayanan, Balaji, Alexander Pritzel, and Charles Blundell. "Simple and scalable predictive uncertainty estimation using deep ensembles." Advances in Neural Information Processing Systems. 2017.

[12] Li, Bo, et al. "Neural-network-based motor rolling bearing fault diagnosis." IEEE transactions on industrial electronics 47.5 (2000): 1060-1069.

[13] Jia, Feng, et al. "Deep neural networks: A promising tool for fault characteristic mining and intelligent diagnosis of rotating machinery with massive data." Mechanical Systems and Signal Processing 72 (2016): 303-315.

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

[15] Zhang, Wei, et al. "A new deep learning model for fault diagnosis with good anti-noise and domain adaptation ability on raw vibration signals." Sensors 17.2 (2017): 425.

[16] Zhou, Bolei, et al. "Learning deep features for discriminative localization." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

[17] Selvaraju, Ramprasaath R., et al. "Grad-cam: Visual explanations from deep networks via gradient-based localization." Proceedings of the IEEE International Conference on Computer Vision. 2017.

[18] Mahajan, Dhruv, et al. "Exploring the limits of weakly supervised pretraining." Proceedings of the European Conference on Computer Vision (ECCV). 2018.

[19] Tan, Mingxing, and Quoc V. Le. "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks." arXiv preprint arXiv:1905.11946 (2019).

# 국문 초록

본 논문에서는 해석 가능한 자동 딥러닝 프레임워크를 제안한다. 제안하는 방법은 총 두 단계로 이루어져 있다. 첫 번째 단계는 사전 학습된 신경망을 고려하여, 신경망 구조의 핵심 하이퍼 파라미터를 탐색 공간으로 정의하여 자동으로 신경망을 최적화하는 단계이다. 사전 학습된 모델과 베이지안 최적화 기법을 이용하여, 정의된 탐색 공간을 최적화함으로써 두 방법론의 장점만을 취하여 강건한 딥러닝 모델을 얻을 수 있다. 두 번째 단계는 해석 불가능한 기존 딥러닝 모델을 해석 가능 하도록 하는 단계이다. 이 단계는 다시, 개별 데이터의 예측의 이유를 설명하는 단계와 예측에 대해 얼마나 확신할 수 있는지에 대해서 추정하는 단계로 나뉜다. 첫 번째로, 기계 시스템의 고장 진단에 주로 활용되는 시간–주파수 영역에서 해석력을 부여하는 방법을 제안하였는데, 특정한 시간 혹은 주파수 마스킹 기법에 따라 엔트로피 변화량을 통해서 중요도를 추정할 수 있다. 두 번째는, 베이지안 최적화 기법을 통해 샘플링된 신경망을 이용하여 앙상블 기반 예측 불확실성을 추정할 수 있다. 제안하는 방법은 볼 베어링 데이터를 활용하여, 다양한 노이즈와 다양한 하중에 대해서 검증되었다. 뿐만 아니라, 고장 진단 이외의 분야에도 다양한 도메인에 적용이 용이할 것으로 보인다.

**주요어:** 건전성 및 예측 관리
고장 진단
자동 머신 러닝
해석 가능 머신 러닝

**학 번:** 2018-20405