공학석사학위논문

# 잔차 합성곱 신경망을 통한 산업용 로봇 기어박스의 동작 적응형 퓨샷 고장 감지 방법

# Motion-Adaptive Few-Shot Fault Detection Method of Industrial Robot Gearboxes via Residual Convolutional Neural Network

2020 년 8 월


서울대학교 대학원

기계공학부

오 영 탁

# Abstract

## Motion-Adaptive Few-Shot Fault Detection Method of Industrial Robot Gearboxes via Residual Convolutional Neural Network

Yeongtak Oh
Department of Mechanical Engineering
The Graduate School
Seoul National University

Nowadays, industrial robots are indispensable equipment for automated manufacturing processes because they can perform repetitive tasks with consistent precision and accuracy. However, when faults occur in the industrial robot, it can lead to the unexpected shutdown of the production line, which brings significant economic losses, so the fault detection is important. The gearbox, one of the main drivetrain components of an industrial robot, is often subjected to high torque loads, and faults occur frequently. When faults occur in the gearbox, the amplitude and frequency of the torque signal are modulated, which leads to changes in the characteristics of the torque signal. Although several previous studies have proposed fault detection methods for industrial robots using torque signals, it is still a challenge to extract fault-related features under various environmental and operating conditions and to detect faults in the complex motions used in industrial sites.

To overcome such difficulties, in this paper, we propose a novel motion-adaptive few-shot (MAFS) fault detection method of industrial robot gearboxes using torque

ripples via a one-dimensional (1D) residual-convolutional neural network (Res-CNN) and binary-supervised domain adaptation (BSDA). The overall procedure of the proposed method is as follows. First, applying the moving average filtering to the torque signal to extract the data trend, and the torque ripples of the high-frequency band are obtained as a residual value between the original signal and the filtered signal. Second, classifying the state of pre-processed torque ripples under various operating and environmental conditions. It is shown that Res-CNN network 1) distinguishes small differences between normal and fault torque ripples effectively, and 2) focuses on important regions of the input data by the attention effect. Third, after constructing the Siamese network with a pre-trained network in the source domain, which consisted of simple motions, detecting the faults on the target domain, which consisted of complex motions through BSDA. As a result, 1) the similarities of the jointly shared physical mechanisms of torque ripples between simple and complex motions are learned, and 2) faults of the gearbox are adaptively detected while the industrial robot executes complex motions. The proposed method showed the most superior accuracy over other deep learning-based methods in few-shot conditions where only one cycle of each normal and fault data of complex motions is available. In addition, the transferable regions on the torque ripples after domain adaptation was highlighted using 1D guided grad-CAM.

The effectiveness of the proposed method was validated with experimental data of multi-axial welding motions in constant and transient speed, which are commonly executed in real-industrial fields such as the automobile manufacturing line. Furthermore, it is expected that the proposed method is applicable to other types of motions, such as inspection, painting, assembly, and so on. The source code is available on my GitHub page of https://github.com/oyt9306/MAFS.

**Keywords:** Industrial robot
Gearbox fault detection
Torque ripples
Residual convolutional neural network
Binary supervised domain adaptation
Few-shot learning
1D guided grad-CAM

**Student Number:** 2018-27498

# Table of Contents

# List of Tables

# List of Figures

# Nomenclatures

| | |
|---|---|
| *MAFS* | Motion-Adaptive Few-Shot |
| *PHM* | Prognostics and Health Management |
| *TFR* | Time-Frequency Representations |
| *HPF* | High-Pass Filtering |
| *MAF* | Moving-Average Filtering |
| *GAP* | Global Average Pooling |
| *BN* | Batch Normalization |
| *Res-CNN* | Residual-Convolutional Neural Network |
| *DA* | Domain Adaptation |
| *UDA* | Unsupervised Domain Adaptation |
| *SDA* | Supervised Domain Adaptation |
| *BSDA* | Binary Supervised Domain Adaptation |
| *GAN* | Generative Adversarial Network |
| *t-SNE* | t-Stochastic Neighbor Embedding |
| *grad-CAM* | Gradient-weighted Class Activation Map |
| $D_S$ | Source Domain Datasets |
| $D_T$ | Target Domain Datasets |
| $f_e$ | Feature Extractor |
| $D_{Label}$ | Label Discriminator |
| $L_C$ | Classification Loss |
| $L_{SA}$ | Semantic Alignment Loss |
| $L_S$ | Separation Loss |
| $L_{Total}$ | Total Loss |

| | |
|---|---|
| $d_{Euc}$ | Metric for Euclidean Distance of Embedding Space |
| $d_{Sep}$ | Metric for Similarity Measure using Penalty Function |
| $N_P$ | Pre-trained Network |
| $N_P$ | Constructed Siamese Network |

# Chapter 1. Introduction

## 1.1 Research Motivation

Robotic arms are widely used in various fields such as automotive, semiconductor, manufacturing, medicine, and so on as they are capable of various complex motions specialized in their tasks. Among them, especially, industrial robots are receiving lots of attention because they can withstand high torque loads, and they are specialized in repetitive motions like welding, assembly, painting, and inspection on the automated production line. However, when faults occur on industrial robots, a decrease in production performance could cause an unexpected shutdown of the production line, which brings significant economic losses. Since operating robots in the industrial fields become aged and exposed to the potential danger to failure, fault detection of industrial robots is essential.

Conventionally, available data to detect the faults can be divided by vibration signals and torque signals. Firstly, the vibration signal is acquirable through the acceleration sensors and DAQ (Data Acquisition) system. It is known as a sensitive signal to detect the faults of the gearbox [1]. According to the research of Kim et al. [2], a proposed phase-based time-domain averaging (PTDA) method is possible to detect the faults of the industrial robot gearboxes on the constant-speed regions of the motion. However, it is limited to the constant speed ranges and also, attaching and managing the wires of the acceleration sensors to acquire the vibration data is not easy in real industrial fields. On the contrary, the torque signal can be acquired at the control stage, and it requires neither installing any supplementary DAQ system

nor intruding the components. So, it can be accessible in the actual industrial line [3] , and fault detection methods based on torque data are widely studied. First, Cheng et al. [4] proposed a Gaussian mixture model (GMM) based gearbox fault detection method with clustering the handcrafted features of the normal and fault torque data using an industrial robot. Second, Bittencourt et al. [5] proposed the method to monitor the degradation of industrial robots executing repetitive motions using kernel density estimation (KDE) method as an indicator of robot wear. However, the aforementioned studies have some limitations in that 1) the suggested feature extraction methods are based on the shallow learning approach, and 2) the aforementioned methods were not validated under various motions of the industrial robot.

The challenges to solve within this thesis are as follows; 1) Hard to extract the health features to detect the faults under various environmental and operating conditions on each motion that are optimized for operating on the real-production line. 2) Pre-trained models have low flexibility for different robot motions. Since robots have different data distributions according to their motions, it is necessary to develop defect detection methods generalized with robot motions. And 3) it is not known how much datasets are needed to train the deep network from scratch for each individual motions sufficiently.

Therefore, in this thesis, a motion-adaptive few-shot (MAFS) fault detection method via residual-convolutional neural network (Res-CNN) is proposed to handle such challenges based on the deep-learning approach. The main contributions of the proposed method are summarized as follows; 1) Minimizing the motion effects from torque signals with pre-processing to extract fault-related torque ripples of the simple and the complex motions. 2) Classifying the torque ripples for simple motions based

on the proposed one-dimensional residual-convolutional neural network (1D Res-CNN) model while robots are under varying conditions of rotating speeds, and temperatures. 3) Adapting a fault detection method through binary supervised domain adaptation (BSDA) to complex motions with a relatively small amount of the data, which needs just one cycle of normal and fault data. As a result, the deep network possible to detect the faults of the industrial robot gearbox flexibly to robot motions, and the proposed fault detection method can be adaptive to any complex motion of the industrial robot. Furthermore, to the best of author's knowledge, it is the first attempt to learn the joint-physical consistency of fault mechanisms on different motions of the robot under the same joint on the robot.

## 1.2 Scope of Research

The scope of this thesis lies in developing the following three research thrusts to address the above challenges. The overall thrusts of the MAFS method are as follows.

**Research thrust 1:** Pre-processing to minimize the motion effects

As the characteristics of the torque signal vary on the environmental conditions such as rotating speeds, temperatures, it is essential to minimize the motion effects from torque signals to detect the faults of the robot. So, by applying cross-correlation, the normal and fault signals are phase-matched and aligned with time. After then, by taking residuals of the raw signals and filtered signals using moving average filtering, torque ripples on the high-frequency components are extracted. So, pre-processed torque ripples on simple and complex motions are used as the input data, respectively. After then, by using a sliding window, one-dimensional data augmentation is performed.

**Research thrust 2:** Robust feature extraction via pre-trained deep network

In this thesis, some experiments are conducted to find the optimized architecture as an end-to-end feature extractor. It is shown that the proposed Res-CNN network performs the best feature extraction performance while training the network. Also, using the residual connection of the network, information flow was improved and small differences between the normal and fault torque ripples are effectively distinguished through the residual propagation. The pre-trained network on unit-axis motions of the industrial robot achieved a high accuracy under variations of operating and environmental conditions such as speeds, and temperatures.

**Research thrust 3:** BSDA based motion-adaptation to the complex motions

As distributions of the torque signals are different depend on robot motions, a pre-trained model on the unit-axis motion cannot be generalized on other motions specified on their own task. In addition, it is quite challenging to determine the number of datasets to train the network. To overcome such difficulties, a pre-trained model on the simple motion was used to train the network adaptively to other robot motions effectively. In addition, it is shown that only one-cycle data is sufficient to train the proposed network adaptively to robot motions. Furthermore, visual interpretations of extracted features are explained via t-Stochastic Neighbor Embedding (t-SNE). And a one-dimensional guided gradient-weighted class activation map (1D guided grad-CAM) was implemented to compare the highlighted regions of before and after MAFS method to verify the localization effects, which shows the improvements in the knowledge transfer.

## 1.3 Thesis Layout

The rest of this thesis is organized as follows. Chapter 2 handles research backgrounds. Chapter 3 describes detailed explanations of the proposed MAFS method. Next, Chapter 4 include experimental validations with generating a pre-trained network and motion-adaptive few-shot learning for complex motions. Finally, conclusions and future works are described in Chapter 5.

# Chapter 2. Research Backgrounds

In this chapter, first, the interpretations of torque ripples are explained in chapter 2.1 with the main causes and modulation effects due to gearbox faults. Second, the architectures of Res-CNN is explained in chapter 2.2 with the convolutional operation, pooling operation, activation, batch normalization, and residual learning. Finally, domain adaptation is explained in chapter 2.3.

## 2.1 Interpretations of Torque Ripples

### 2.1.1. Causes of torque ripples

Articulated robotic arms, which include industrial robots and cooperative robots, are a complicated control system composed of drivetrain mechanical components coupled on each robot joint such as motors and gearboxes. Configuration of the robot control system on each joint is composed of an adjustable speed drive, encoder, and motion control unit for feedback control [4]. An encoder attached to the rotating shaft records the rotating speed of the motor shaft. A current sensor measures the current signal between the adjustable-speed drive and robot system, and the signal from the encoder is fed into the motion control unit. After the reference motion input, a feedback control signal which is the input of the adjustable-speed drive is generated to control the rotating speed. A motion control unit compares the real motion with reference motion, which is measured by rotating speed from an encoder on the motor shaft. So, the feedback control adjusts the robot motions by minimizing the differences of reference and output motions. The detailed control diagram configuration is summarized in Figure 1. For the detail, a simple one-mass model

equation for one-joint of the robot system can be defined by as follows.

$$\mathrm{T(t)} = J_S \frac{dw_r(t)}{dt} + Bw_r(t) + T_m(t) \tag{1}$$

Where $J_S$ is the total inertia of the system. $\mathrm{w}_r$ is the rotating speed of the angular shaft.

*Torque ripples*, which are known as the noise oscillating along the pathway of the robot, lead to uncontrollable vibration and deteriorate the control performance of the robot. Torque ripples [6-8] are inevitable for the robot systems, which include drivetrain components such as a motor and a gearbox. Also, torque ripples have their own characteristics of the frequency band depending on the mechanical properties of the motor and gearbox [9]. Torque ripples are mainly caused by excitations due to vibrations, which mainly come from two causes [10]: external input torque and internal torque. The former will not be covered in this thesis, and the latter is mainly caused by drive train components.

First, in permanent magnet synchronous motor (PMSM), which is commonly used in the industrial robot, variable magnetic reluctance due to rotation of the shaft is the main cause of torque ripples. In the normal state, ripples are caused by the interaction between the stator and the rotor slots, which is called the commutation effect. The fundamental frequency of the commutation torque ripple is directly related to the rotational speed of the motor. The frequency is known as the pole passing frequency which would be determined as,

$$\mathrm{f_c} = wN_rN_p/2\pi \tag{2}$$

Here, ω = mechanical speed [rad/s], $N_r$ = the number of rotor poles, and $N_p$ = the number of rotor phases.

Second, due to mechanical misalignment of the motor shaft and the gearbox, sinusoidal behaviors on ripples show periodical shapes, and its characteristic frequency is induced by motor shaft speed. The main cause of the mechanical misalignment is from the imperfect mounting of a motor shaft and reducer shaft.

Third, as the gearbox generates a periodical angular transmission error during high-load torque transmission on the contact point, a characteristic frequency of high-frequency oscillations is determined by the rotating speed of the motor due to gear-meshing vibrations occur. So, path control of the end-effector motions is effected by such oscillations.

### 2.1.1. Modulations on torque ripples due to gearbox faults

As torque ripple mechanisms are different in terms of the size and tasks of the robot, industrial robots can be categorized depending on the sizes and the payloads. Relatively small size and low payload, such as cooperative robots commonly use the harmonic drive as a gearbox, and contrary, relatively big size and large payload such as industrial robots use cycloidal gearboxes. A cycloidal gearbox is widely used in the industrial robots for transmitting high torque due to the high gear transmission ratio [11]. The general configuration of the cycloidal gearbox is shown in Figure 2, which is configurated by an input gear and input shaft, spur gears, and cycloidal disk and pin. In the drivetrain system, the electric motor is driven at high speed with low torque, and the gearbox reduces the high speed of the motor shaft to transmit the large torque [12]. The major causes of industrial robot faults can be mainly divided into an electrical and mechanical component fault, and about 45% of the faults are caused by the degradation of a mechanical system [13], especially for the gearbox

system. As industrial robot systems operate inside the factory, they are isolated from the external environment and rarely affected by external factors such as noise and vibrations. Factors that can affect the ripples in case of gearbox faults include operating conditions such as speeds, motions, and temperatures. When faults occur on the gearbox, amplitude and frequency modulations cause the changes in ripples of the feedback signal, which differ from normal ripples. So, when the faults occur on the gearbox of a robot, faults can be detected on the high-frequency band compared to the normal case due to modulation effects. The equation of modulated torque signals is as follows;

$$\mathrm{T(t)} = T_0(t) + \Sigma_{n=1}^{N} T_n(t)\cos\left[\int 2\pi f_n(t)dt + \phi_n(t)\right] \quad (3)$$

Where $\mathrm{T}_n$, $f_n$, and $\phi_n$ each denotes the amplitude, frequency, and phase of the torque ripple components induced by external vibrations.

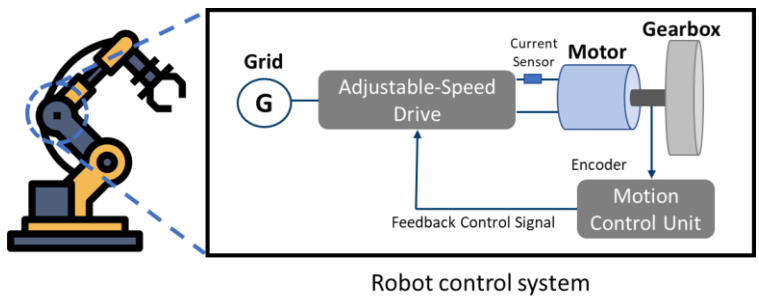


**Figure 1.** The control diagram of a robot system

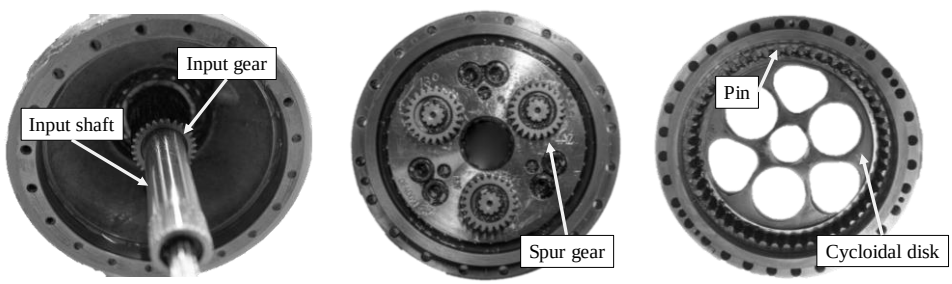


**Figure 2.** Configuration of the cycloidal gearbox in an industrial robot: (a) input gear and input shaft, (b) spur gears, (c) cycloidal disk and pin [2]

## 2.2 Architectures of Res-CNN

This chapter examines the fundamental principles of the proposed Res-CNN. First, we will discuss the convolutional operation, pooling operation, activation, batch normalization. Next, the residual learning part, information flow, and regularization effect are covered.

### 2.2.1 Convolutional Operation

Convolutional operation convolves the local regions of the input to extract non-linear output features with filter kernels. During convolution, each filter is used to extract the specific features of the input by applying the same kernel, whose operation is referred to as weight-sharing. In each layer of the network, the number of filters determines the weight parameters, and the more filter kernels learn the more abundant representations. The hierarchical filter structure enables extracting the meaningful features of the input evolve from a low level to a high level through a hierarchical deep-layers. The convolution process is explained as follows.

$$y_i^{l+1}(j) = K_i^l \cdot x^l(j) + b_i^l \tag{4}$$

Where $l$ means the layer, and in layer $l$, the $i^{th}$ filter kernel is composed of the weights $\mathrm{K}_i^l$, and the bias $b_i^l$. Also, $\cdot$ is the dot product, and $x_i^l$ denotes the $j^{th}$ a local region of the layer. $y_i^{l+1}(j)$ represents the input of the $j^{th}$ neuron corresponds to $i^{th}$ frame on layer $l$+1. As Bengio et al. [14] showed that the representation performance is proportional to the depth of the layer, so as the layer goes deeper, learning the non-linear feature representation becomes drastically improved. Such learning is called manifold or representation learning.

### 2.2.2 Pooling Operation

#### 2.2.2.1 Max Pooling

To guide the network to learn the informative features from the input, increasing the receptive field of each layer is the key solution. In this way, max pooling is effective to transfer the global information on top-down architecture after a small number of residual propagations [15]. After a convolutional layer in the CNN architecture, successive pooling layer down-samples the spacial size of the features and the network parameters. Especially, max-pooling takes the role of local max operation on the input, so the parameters are reduced, and extracted features become locally invariant. The max-pooling operation is explained as follows:

$$\mathrm{P}_i^{l+1}(j) = \max_{(j-1)W+1 \le t \le jW} [q_i^l(t)] \tag{5}$$

Where $q_j^i(t)$ means the value in the $i^{\mathrm{th}}$ frame of layer $l$ at $t^{th}$ neuron. Also, the pooling operation pools the input region with the width of W and $\mathrm{P_i^{l+1}}(j)$ means the value of $j^{th}$ neuron in layer $l$+1.

#### 2.2.2.2 Global Average Pooling (GAP)

The conventional FC layer is prone to overfitting that hampers the generalization ability of the overall network and heavily depends on dropout, which has effects of the ensemble-based regularizer. However, in contrast to FC layer, Global Average Pooling (GAP) [16], which takes average values of all activation information, is known for better suited to localizing the entire location of things compared to the fully-connected (FC) layer. GAP layer is a structural regularizer, and overfitting does not occur during the training procedures as there is no parameter to optimize in the GAP layer. Thus, in this thesis, GAP is used as a flatten layer instead of FC layer.

Furthermore, as GAP layer sums out the spatial information, it is more robust than FC layer for any spatial translations over the input and extracts more meaningful and interpretable features from the input.

### 2.2.3 Activation

After the convolution operation, an activation function is essential to apply. It enhances the network performance to train the non-linear expression of the input and makes the learned features more informative. In recent years, the Rectified Linear Unit (ReLU) [17] has been widely adopted as a non-linear activation unit of CNNs. The detailed formula is described as follows:

$$a_{\mathrm{i}}^{(l+1)}(j) = \mathrm{f}\left(\mathrm{y}_{\mathrm{i}}^{(\mathrm{l}+1)}(j)\right) = \max[0, \mathrm{y}_{\mathrm{i}}^{(\mathrm{l}+1)}(j)] \tag{6}$$

where $y_i^{(l+1)}(j)$ is the output value of convolution operation and $a_i^{(l+1)}(j)$ is the activation of $y_i^{(l+1)}(j)$. ReLU is one of the most popular activation functions for deep learning methods, including the sigmoid and hyperbolic tangent (tanh) functions, as it is effective in preventing the gradient vanishing problem.

### 2.2.4 Batch Normalization

The batch normalization (BN) layer [18] is known as regularization effects similar to dropout, and normalize the internal covariance and shift of the data to facilitate the deep neural network to converge into the global minimum during the training process. Conventionally, in the network architecture, the BN layer is added between the convolutional layer and the activation unit. Given the input to a BN layer with $\mathrm{x} = [\mathrm{x}^{(1)}, \mathrm{x}^{(2)}, \ \ldots, \mathrm{x}^{(m)}]$, the BN layer is described as follows:

$$\hat{x}^{(i)} = \frac{\mathrm{x}^{(i)} - \mathrm{E}\left[\mathrm{x}^{(i)}\right]}{\sqrt{Var[x^{(i)}]}} \tag{7}$$

$$\mathrm{y}^{(i)} = \gamma^{(i)}\hat{x}^{(i)} + \beta^{(i)} \tag{8}$$

Where $\mathrm{y}^{(i)}$ is the output corresponds to the input $\mathrm{x}^{(i)}$, trainable parameters of $\gamma^i$ and $\beta^i$ are the scale and shift parameters. The process of BN is as follows. After standardizing the features in each dimension independently, $\gamma^i$ and $\beta^i$ are used to restore each normalized feature by scaling and shifting During training procedure, $\gamma^i$ and $\beta^i$ are optimized after iterative back-propagations. So, data distributions of input are improved after such an identity transformation.

### 2.2.5 Residual Learning

Residual learning [19] was first proposed by He et al. in 2015 and represented remarkable performance improvements in the image recognition field compared to other networks such as VGGNet [20], GoogleNet [21], and so on. Residual networks take some advantages in 1) solving vanishing/exploding gradient issues, and 2) promoting the convergence speeds of the network, whether on the deep depth of the network.

#### 2.2.5.1 Network Architectures

In the network architectures, the residual network has some structural advantages in that while the existing methods update the weights of layer output $\mathrm{H}(\mathrm{x})$, which is shown in equation (9), residual network update the weights using a short connection by adding the input x on the output value, which is called identity mapping in equation (10):

$$\mathrm{F}(\mathrm{x}) = \mathrm{H}(\mathrm{x}) - \mathrm{x} \tag{9}$$

$$\mathrm{H}(\mathrm{x}) = \mathrm{F}(\mathrm{x}) + \mathrm{x} \tag{10}$$

Therefore, the output in the layer can be expressed as the summation of the input $x_0$ and residual operation F. Residual network has drastically simplified forward and backward paths compared to typical CNN which paths consist of plenty of multiplications. So, based on residual propagation, the residual network has some advantages in aspects of the small number of network parameters, the low computational complexities for stability, and the increased accuracy of the network. Detailed operations are explained in the below equation (11):

$$x_{L+1} = x_l + F(x_l) = \cdots = x_0 + \Sigma_{i=1}^{L} F(x_i) \quad (11)$$

Where input residual block operation is F, and output $x_i$ at $i^{th}$ layer which also means the input of $i + 1^{th}$ layer. As a result of the addition, which directly passed into the next layer, the input is preserved and also information would always flow unimpeded. Consequently, by considering the original input to compute the gradients, so that information flow can be transmitted into the deep layer as gradients at every single layer of the network do not vanish.

#### 2.2.5.2 Information transfer

Through a bunch of the architecture study, the milestone to the success of learning the deep networks was the use of batch normalization known as regularization effects similar to dropout. Also, to guide or force the network to learn the informative features from the input, increasing the receptive field of each layer is the key solution. In this way, max pooling is effective to transfer the global information on top-down architecture after a small number of residual propagations [15]. In addition, a residual network explained in the above chapters is also effective for information transfer. Wang et al. [22] showed that the residual network focuses on the local regions learning the transferable features across domains that attend to the objects of interest. By the attention effect of residual network, rather than focusing on wrong local regions, which decreases the performance, the regions of more transferable representations become weighted. Thus, the model focuses on those important regions, even when it is used for the domain adaptation task. After the residual propagation, transferable regions that share the cross-domain features become highlighted.

## 2.3 Domain Adaptation (DA)

In order to mitigate the generalization performance induced by the covariance shift, transfer learning is commonly used in broad pre-trained models to perform a similar task on the other domain [23]. Transfer learning enables the deep network to adjust the detailed parameters in deep layers trained on the large datasets, by parameter freezing and fine-tuning the network that prevents overfitting and enables the generalization performance. Moreover, domain adaptation (DA) is a branch of transfer learning, which efficiently transfers the information when the number of data in the target domain is relatively small. Therefore, DA has been received lots of attention since deep networks are enabled to learn the domain-invariant representations from the cross-domains.

However, when the domain discrepancy is severe, the performance of the DA is significantly reduced. So, to minimize the domain discrepancy, lots of studies have been proposed to match the source and target data distributions on the feature space. Unsupervised domain adaptation (UDA) [24, 25] has been receiving lots of attention because it does not require any labeled data of the target domain, but when the number of data is small, estimating the distributions from the given data is challenging. For the solution of such a situation, supervised domain adaptation (SDA) [26] is proposed as an effective method to sustain the same supervision on the target domain from the source domain to find a shared subspace across the domain. The main idea of SDA is to construct the Siamese network to learn similarities of cross-domain feature distributions by point-wise comparisons on the embedding subspace. Siamese network is well known for one-shot or few-shot learning as this architecture compares the high dimensional feature representations using distance metrics with shared network parameters. So, as a result, the target domain feature distributions

become semantically aligned yet maximally separated, even the number of labeled target datasets per category is low. In this way, supervised learning could be a countermeasure of UDA in the domain adaptation tasks.

Prognostics and health management (PHM) is a key technology for diagnosing the health conditions of the mechanical systems that detect abnormalities from the data collected from sensors attached to the equipment. In PHM, deep learning-based fault detection methods have low generalization performances in the real industrial fields because of the reasons as follows. 1) Weight parameters of the network are not fitted on different conditions caused by time-varying operating conditions (e.g., loads, rpm) and environmental conditions (e.g., noise, temperature). 2) Acquiring the labeled target data is difficult because labeling the data requires much labor forces. So, to solve the above challenges, several attempts using DA have emerged toward accomplishing the flexible fault detection method of bearing [27], gearbox [28], [29], and so on.

### 2.3.1 Few-shot domain adaptation

Few-shot learning is the research field of training the deep network to recognize the objects with only a few data. The recently introduced studies are mainly summarized as the following three parts. First, using a pre-trained network on similar tasks [30], the initial parameters taken from the existing model are fine-tuned with the target data, by aligning the feature distributions via adversarial learning, even if few samples are on the target domain. This method is effective in the target domain for doing similar tasks from the source data. Second, using a generative adversarial network (GAN) [31], few-shot classifiers of the network are generalized by learning the sharper decision boundary for the unlabeled data. Third, using meta-learning and

graph neural network (GNN) [32], prototype-based reduced network complexity allows the model to derive its own learning rules to the few-shot task in an episodic training. Also, GNN was introduced to improve the performance of few-shot learning to utilize complex relational information between data.

Recently, few-shot learning-based fault detection methods in the PHM field is have been emerging [33]. As collecting and labeling enough data need much labor forces and time-consuming on real production processes, such a few-shot learning-based domain adaptation is required to achieve the low-cost and life-long sustainable network. Thus, in this thesis, to cope with low-performance of the network trained using sparse labeled samples on the target domain, supervised domain adaptation based fault detection on the few-shot condition using a pre-trained model is used. Detailed explanations will be driven in the next proposed method chapters.

# Chapter 3. Motion-Adaptive Few-Shot (MAFS) Fault Detection Method

In this chapter, we propose a novel fault detection method named MAFS. The proposed MAFS method consists of three steps of pre-processing in chapter 3.1, pre-training in chapter 3.2, and binary-supervised domain adaptation (BSDA) in chapter 3.3. The definition of the major notations used in this chapter are described as follows. *Domain* means sample space of the motion on the specific joint of an industrial robot, where the motion difference induces the domain discrepancy. Second, *operating condition* means the occasion such as speed and motions when the robot performs the specific tasks. On the contrary, *environmental condition* means the ambient conditions such as noise and temperatures when the robot is working.

The overall framework of the proposed MAFS method is shown below in Figure 3. At step 1, after acquiring the normal and fault torque signals on each simple and the complex motion, pre-processed torque ripples are used as input data. At step 2, the Res-CNN model is trained using torque ripples on the source domain. At step 3, using a pre-trained model, fault detection of the simple motions under various operating and environmental conditions is performed. At step 4, after training the network, randomly sampled data from the source domain data is used to transfer the extracted features onto the target domain. At step 5, the Siamese network is constructed for knowledge transfer by initializing the weights using a pre-trained network. At step 6, the constructed network is trained with few samples of the target domain via BSDA by comparing the embedded feature distances from different domains. Finally, at step 7, fault detection on the complex motions using a motion-

adapted network is done. So, labels of the samples on the complex motions are predicted as normal and the fault states. Furthermore, using t-SNE and guided grad-CAM, visualizations of embedded feature distributions and interpretations on localization effect before and after domain adaptation are presented.

The schematic diagram of the proposed method is described in Figure 4. In the real-industrial sites, lots of robots perform complex motions optimized on their own tasks. When the robot which executes the various complex motions in the industrial field is suspected to the gearbox faults, torque signal is sent to a MAFS based deep-learning solution. Then, pre-processed torque ripples of normal and suspected to faults are pairly constructed as input data. Finally, a Siamese network constructed using a pre-trained network is trained via BSDA. While training, if the detection accuracy is low, gearbox fault would not be severe, or it is not related to the gearbox faults. On the contrary, if the detection accuracy is high, a fault is suspected to the similar case of pre-acquired gearbox fault occasions, and it could be diagnosed as a gearbox fault on the specific joint. As a result, the fault detection of the robot for each complex motion is possible.

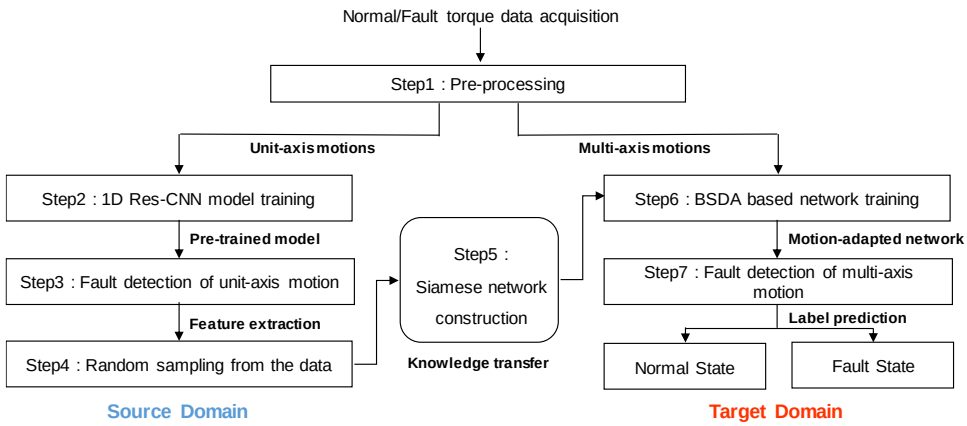


**Figure 3.** The overall framework of the proposed MAFS method

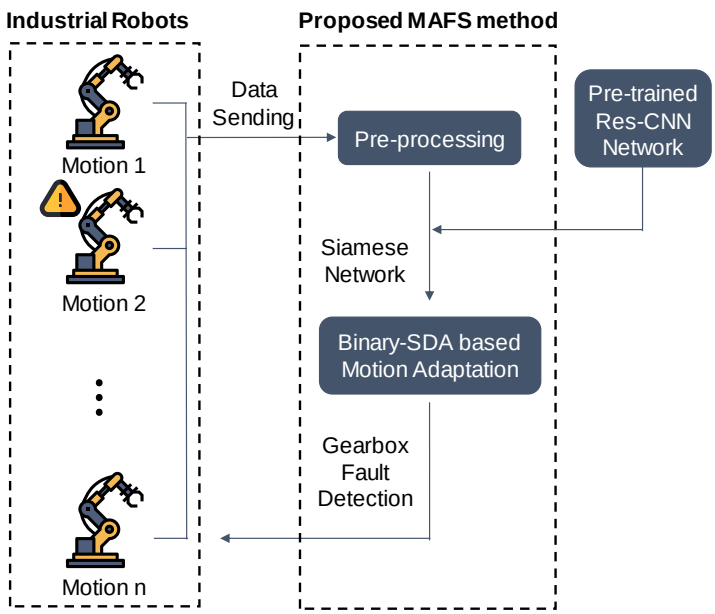


**Figure 4.** The schematic diagram for applications of the proposed method in the real-industrial field.

## 3.1 Pre-processing

In this chapter, pre-processing procedures are mainly composed of data acquisition, extracting torque ripples, and data augmentation. First, in the data acquisition procedure, cross-correlation is applied to match and synchronize the phases of normal and fault torque signals. The characteristic features of the torque ripples are as follows. The torque signals are obtained by converting the current signals in the control phase. Also, as the sampling rate of the torque signal is 1,000 Hz, the resolution in the frequency band is low because the Nyquist frequency of the signal is just about 500Hz. Therefore, conventional time-frequency representations (TFR) like a spectrogram, scalogram are hard to be applied properly to extract health features from the signal. To cope with such limitations, moving average filtering, which is almost identical to a low-pass filter (LPF), is applied to highlight the health features. As a result, by taking residuals of raw and filtered signals, the trend of the signal due to the effects of robot operation is minimized, and torque ripples on the high-frequency component related to the gearbox faults are obtained. Finally, sliding window-based data segmentation is performed for data augmentation.

### 3.1.1 Cross-correlation

The formula of the cross-correlation is shown below equation (12). As function $g$ slides along the x-axis, the summation of the product of function $f$ and $g$ is calculated at each position. When each function match with each other, the value of the product $(f * g)$ is maximized due to a large contribution of aligned positive areas, and vice versa. So, it measures the degree of similarities between two waveforms based on a time-lag as a series of a function of the distance relative to each other, and so-called as sliding dot product or sliding inner-product. So, through

the cross-correlation, the phase of normal and fault ripples is aligned in the direction of maximizing ($f * g$), by minimizing the time-lag.

$$(f * g) = \sum_{m=-\infty}^{\infty} f^{*}[m]g[m+n] \tag{12}$$

### 3.1.2 Moving-Average Filtering (MAF)

Although LPF is sensitive to the parameters such as cut-off frequency and the order, MAF is less sensitive than LPF because it only needs the number of samples to calculate the average value. The description of the MAF is as follows. In equation (11), $y[n]$ is the current output, $x[n]$ is the current input, $x[n-1]$ is the previous input, and N is the length of the average. We set the parameters of the filter as 50, which means one window takes 50 data points to smooth the signal. Also, we added the compensation term to restore the phase delay. By applying MAF to the raw signals, it takes some advantages of 1) minimizing the trend of the signal due to the robot movement, and 2) extracting the ripples from the original signal on the high-frequency band.

$$y[n] = \frac{1}{\mathrm{N}} \sum_{i=0}^{N-1} x[n-i] \tag{11}$$

In Figure 5, the plots of aligned normal and fault torque ripples on the simple motions (a) during clockwise rotation and (b) during counter-clockwise rotation are shown. Where the blue plot means the normal state, and the red plot means the fault state. In Figure 5, when a fault occurs on the gearbox, the peak of the ripples is larger, and also, the value of root-mean-square (RMS) tends to be higher than the normal case. In Figure 6, as the torque ripples are mainly dependent on the speed conditions, sidebands occur on the ambient frequency of each characteristic frequencies. Also,

any interferences from other axes do not exist during the rotation.

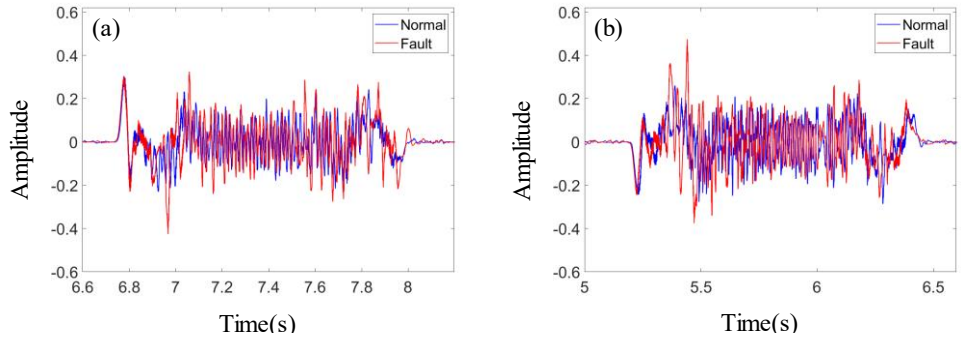


**Figure 5.** Time-domain plots of torque ripples on (a) CW, and (b) CCW rotation during simple motion

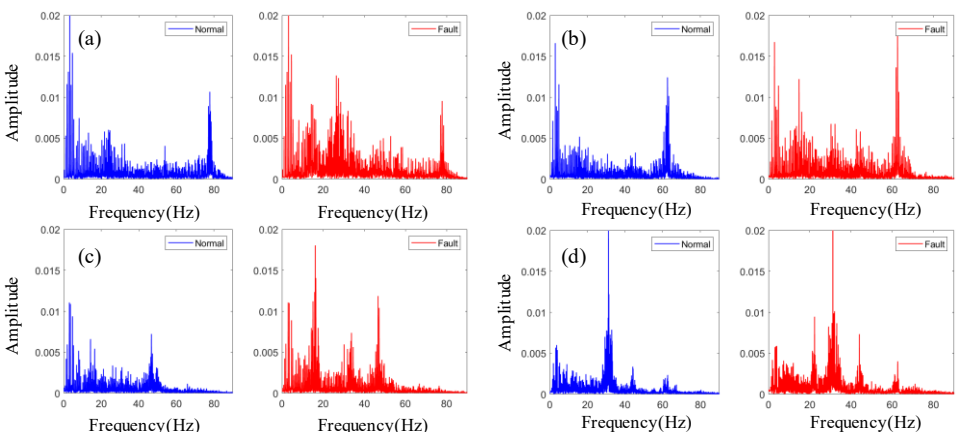


**Figure 6.** FFT plots of torque ripples on (a) 100%, (b) 80%, (c) 60%, and (d) 40% of the full speed of rotating speed during the simple motion.

In Figure 7 and Figure 8, for the case of complex motions 1 and 2, it can be seen that the states of complex motions are not clearly distinguished both in time and frequency domains, unlike the above simple motion case. As the robot moves complexly during the motion, many axes rotate simultaneously, and modulations on

the signal due to faults are not apparent due to non-periodical and non-stationary motions. A detailed description of the motions will be covered in the experimental validation chapter.

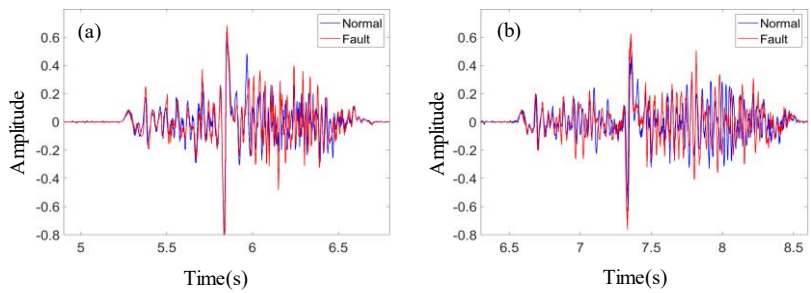


**Figure 7.** Time-domain plots of ripples on (a) complex 1 motion, and (b) complex 2 motion

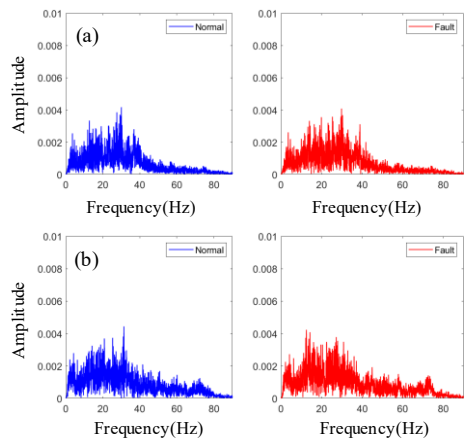


**Figure 8.** FFT plots of ripples on (a) complex 1 motion, and (b) complex 2 motion.

### 3.1.3 Data Augmentation

Since torque ripples contain fault-related characteristics on the variable speed conditions over the input regions, augmented data based on window sliding contains fault-related information. After aligning the normal and fault state torque ripples, each sliding window with overlap is used as an input of the network. The proper length of the window, which does not include a zero-speed region, was set as 3.6 (s) for window length and 0.1 (s) for the window shifting length. In Figure 9, the process of augmentation of the input data is shown, and the same augmentation procedure is performed both on the data of simple and complex motions.

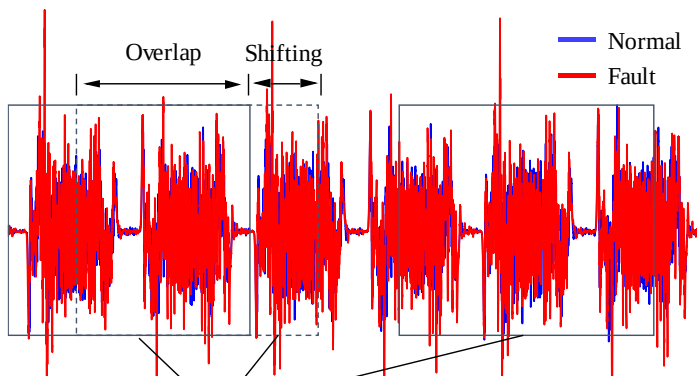


**Figure 9.** Data augmentation of torque ripples based on a sliding window.

## 3.2 Network Pre-training

In this chapter, the baseline network to train on the simple motion is a residual network, as residual-based network architecture takes some advantages on deep-layers with no gradient vanishing problem and fast convergence speed. In the view of fault detection, Ma et al. [34] showed that residual learning has better generalization performance than other CNNs, even under non-stationary working conditions, to extract features from the input due to the structural advantages of the network. Furthermore, in the network architecture, He et al. [35] showed the use of the pre-activation layer before merging with the shortcut contributes to optimization results in better accuracy on the test dataset through a bunch of architecture studies.

So, a skip connection with the pre-activation layer is finally adopted as a basic block of the proposed Res-CNN, as described in Figure 10, among several building block candidates. Since an exception of the addition path causes the output to become un-normalized, the input to the next layer is not normalized properly. By using the pre-activation layer, the input is always normalized at the very beginning of the identity unit. In one residual block, the BN layer adopts the full pre-activation structure in front of the ReLU, which shows the regularization effects. In Figure 11, (a) and (b) are no skip connection architecture, (c) and (d) is with skip connection network where only (d) uses the pre-activation layer. The notation of non-linear activation is $\sigma$, a convolutional block which consisted of batch norm-activation-convolutional operation is F, a max-pooling operation is $Pool$, and the layer input is $x_l$ at the $L^{th}$ layer. So after the proposed residual block, layer output is as follows;

$$x_{l+1} = Pool(F(\sigma(x_l))+\sigma(x_l)) \tag{22}$$

Where input non-linear function F of the $L + 1^{th}$ layer means skip connection

based residual block operation. We used a deep feature extraction with one plain convolutional block and three residual blocks before the GAP layer, and detailed descriptions of network parameters are as follows in Table 1. In one plain convolutional block, the concept of wide-kernel [36] was used. As a result, low-level features are extracted in the first layer, and while passing the shallow kernels on the sequential residual blocks, high-level features of the input can be efficiently extracted due to the attention effects. The number of total trainable parameters of the network is 75,138.

**Table 1.** Detailed parameters of the proposed Res-CNN

| Layer | Type | Kernel Size | Stride | Channels | Parameter |
|---|---|---|---|---|---|
| Input | Window | 3600 | - | - | - |
| Conv block1 | Convolution | 128 | 1 | 64 | 8256 |
| | Max pooling | 2 | 2 | 64 | |
| Res block1 | Batch norm | - | - | - | 256 |
| | Convolution | 6 | 1 | 64 | 24640 |
| | Max pooling | 2 | 2 | 64 | |
| Res block2 | Batch norm | - | - | - | 256 |
| | Convolution | 4 | 1 | 64 | 16448 |
| | Max pooling | 2 | 2 | 64 | |
| Res block3 | Batch norm | - | - | - | 256 |
| | Convolution | 4 | 1 | 64 | 16448 |
| | Max pooling | 2 | 2 | 64 | |
| GAP | Flatten | 1 | - | # of class | - |
| FC | Fully-connected | 128 | - | 1 | 8320 |
| | Classification | # of class | - | 1 | 258 |

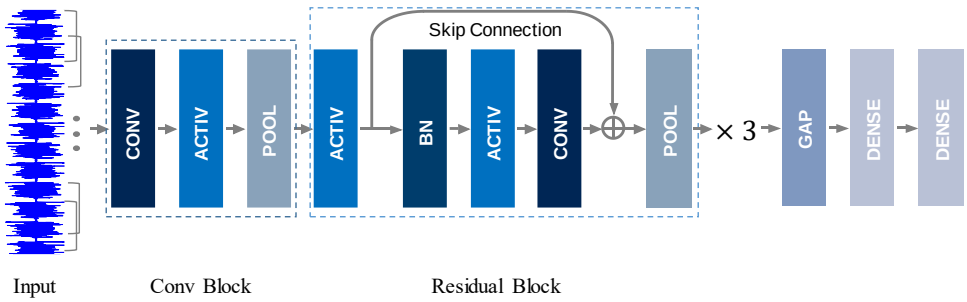


**Figure 10.** The proposed architecture of Res-CNN

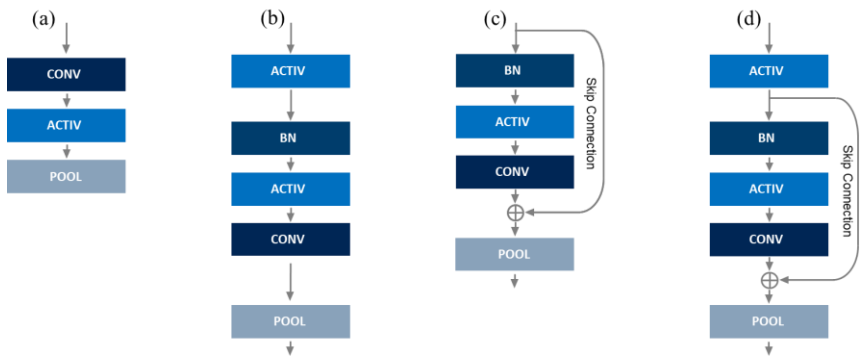


**Figure 11.** Architecture studies for the proposed network. Each (a) Plain block, (b) pre-activation without skip connection, (c) skip connection without pre-activation, (d) skip connection and pre-activation

## 3.3 Binary-Supervised Domain Adaptation (BSDA)

In this thesis, DA is used to guide the network to learn the non-parametric data distributions of the shared physical mechanism encompassing the complex governing equations of the cycloidal gearboxes. As robots are complexly coupled, and delicate systems, the governing mechanisms of the occurring faults are hard to be identified. To overcome such limitation, minimizing the metric-based mismatches on the embedded feature distributions using DA was applied to disentangle the manifold of such feature representations of the complex motions.

The several challenges to solve in this chapter are as follows. 1) Data acquisition: it is challenging to construct a sufficient dataset to train the network depending on the motion. So, the use of DA allows few-shot learning with a decent performance that lowers the laboring cost to acquire the labeled data of complex motions. Also, as the fault-severity is most clearly displayed and labeled in the unit-axis motion, it is suitable to transfer the fault-related knowledge. 2) Motion adaptation: as an industrial robot performs various tasks, the network trained on a specific motion would have poor generalization performance for other motions due to the domain discrepancy. Therefore, motion-adaptive fault detection is required to generalize the fault detection method.

Since the robot performs various tasks repetitively, motion on the specified task is less affected by external environmental factors such as noise, so variations of the torque signal are not large. Thus, the problem formulated in this thesis is concerned with how much-labeled samples are required to sufficiently train the network. In this chapter, it was confirmed that the proposed BSDA method is effective to learn the shared features of motion-invariant physical mechanisms for different motions with few-samples. In summary, the proposed method sufficiently learns with only-one

cycle data of normal and fault state of complex motions with a pre-trained network.

The detailed notations used in this thesis are as follows. *Source domain* is described as $D_S = \{(x^S_i, y^S_i)\}_{i=1}^{n_S}$, where $n_S$ is the number of data on the source domain, the feature $x^S_i$ is from a random variable $X^S$ which is the source domain for training the pre-trained network, and the $y^S_i$ is a corresponding label from a random variable $Y^S$. In addition, *target domain* is described as $D_T = \{(x^T_j, y^T_j)\}_{j=1}^{n_T}$, where $n_T$ is the number of data on the source domain, feature $x^T_j$ is from random variables of $X^t$ which is the target domain for training the network, and the $y^T_i$ is a corresponding label from a random variable $Y^T$.

*Covariance shift*, which means the differences between probabilistic distributions of feature vectors from different domains after the FC layer, represents the discrepancy between the source and the target domain. The goal of the learning is to find out the mapping function $f : X \to Y$ on the target domain, that is equal to minimizing the covariance shift, where $f$ is composed of $f_e$ and $D_{Label}$, and $f_e$ means the notation of feature extractor of $f_e : X \to Z$. $D_{Label}$ is the label discriminator of $D_{Label}: Z \to Y$, and the notation Z is the embedding space after GAP layer. *Siamese network* [37] is an effective network architecture finding a shared subspace between source and target distributions. It is beneficial for transferring the knowledge from source to target domain, especially using small datasets. Since $f_{e,S} = f_{e,T} = f_e$, the parameters of feature extraction layer are shared in the architecture. In addition, the learning of sustaining the source stream would continue with GAP layer to train the $D_{Label}$. The classifier f is trained by minimizing the *classification loss* as follows.

$$L_C(f) = E[L_{BC}(f(X^S), Y)] = -Ylog\big(f(X^S)\big) - (1 - Y)\log(1 - f(X^S)) \qquad (13)$$

Where E denotes statistical expectation and $L_{BC}$ is the loss function of binary-cross-entropy with equation (13). Also, *semantic alignment loss*, which guarantees that the samples from different domains of the same label are aligned nearby after being mapped on the embedding space is as follows.

$$L_{SA}(g) = \sum_{a=1}^{N_C} d_{Euc}\left(p\left(f_e(X_a^s)\right), p\left(f_e(X_a^t)\right)\right) = \sum_{a=1}^{N_C}\sum_{i,j} d_{Euc}\left(f_e(x_i^s), f_e\left(x_j^t\right)\right) \quad (14)$$

Where $N_c$ means the number of class of the network, and we adopted the binary case as $N_C = 2$. This encourages the samples from different domains but the same label, to be mapped nearby in the embedding space, which means it minimizes the discrepancy on the in-class data distributions of the different domains. And $d_{Euc}$ means the metric of Euclidean distance between distributions in the embedding space as follows.

$$d_{Euc}\left(f_e(x_i^s), f_e\left(x_j^t\right)\right) = \frac{1}{2}\left|\left|f_e(\mathrm{x}_i^s) - f_e\left(\mathrm{x}_j^t\right)\right|\right|^2 \quad (35)$$

Also, to maximize the out of class data distributions of the different domains, a class separation that takes the role of mapping apart as possible in the embedding space is required. So, the *separation loss* is described by the following term as,

$$L_S(g) = \sum_{a,b|a\neq b} d_{Sep}\left(p\left(f_e(X_a^s)\right), p\left(f_e(X_b^t)\right)\right) = \sum_{a,b|a\neq b}\sum_{i,j} d_{Sep}\left(f_e(x_i^s), f_e\left(x_j^t\right)\right) \quad (46)$$

By combining it with the *semantic alignment* loss, it is less prone to generate errors and achieves much better performance. Where $d_{Sep}$ is the metric of calculating the similarities between the distributions by adding the penalty function when the class is different, the $||\cdot||$ denotes the Frobenius norm, and the Boolean calculation that the minimum $d_{Euc}$ should be larger than set values, which used 1e-3 in this thesis.

$$d_{Sep}\left(f_e(\mathrm{x}_i^s), f_e(\mathrm{x}_j^t)\right) = \frac{1}{2}\max\left(0, 1 - \left|\left|f_e(\mathrm{x}_i^s) - f_e(\mathrm{x}_j^t)\right|\right|\right)^2 \quad (17)$$

In summary, by combining with $L_{SA}$ and $L_C$, learning is minimizing the dissimilarities for different domains with the same label and maximizing the inconsistency for different domains with different labels. Based on the above *classification loss*, *semantic alignment loss*, and *separation loss*, the *total loss* term is described as follows.

$$L_{Total}(f) = \alpha L_c(D_{Label} \circ f_e) + (1\text{-}\alpha)(L_{SA}(f_e) + L_S(f_e)) \quad (18)$$

Where $\alpha$ is the weighting parameter to adjust the contributions of each loss. The proposed Siamese network architecture is described in Figure 12.

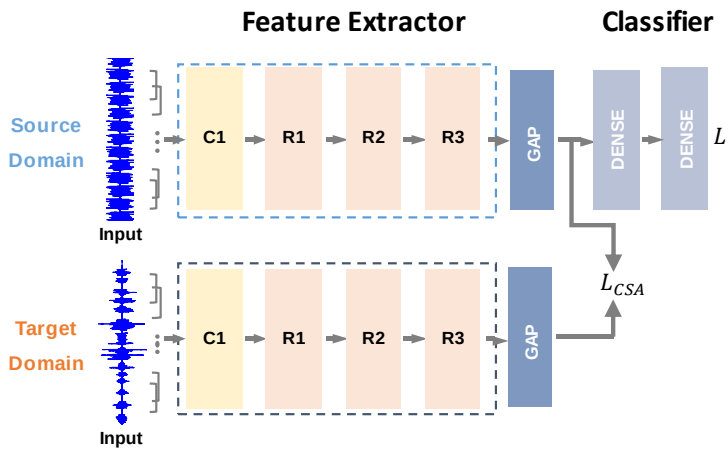


**Figure 12**. Constructed Siamese network with Res-CNN pre-trained network. Where C means the convolution block, and R means the residual block.

The overall pseudo code of the MAFS algorithm is in Figure 13. First, in order to generate a pre-trained model, the $N_P$ network is trained by randomly extracting mini-batch $D_S^m$ from the data of the source domain. And then, the samples of source and target domains are randomly selected with the number of m, n, respectively. Random sampling is updated per epoch, and learning is completed at the point where the loss converges. At this time, mini-batch $D_{S,T}^k$ is generated from the constructed pairwise data with point-wise comparisons of $D_{S,T}$ which is $\mathrm{m} \times \mathrm{n}$ dataset for distinguishing between cases. Finally, after constructing $N_S$ as a Siamese network using $N_P$, the network is trained by comparing feature similarity between the source and target data in mini-batch. And while the training procedure, the label discriminator $D_{Label}$ is fine-tuned on the few samples of the target domain.

**Algorithm 1.** Training procedure of MAFS. The *iter1* and *iter2 each* means the iterations in the training and domain adaptation stage; m, n, k is the batch size of $D_s, D_T, D_{S,T}^k$ respectively.

**Input** Source domain data $D_s$, Target domain data $D_T$,
Pre-trained network $N_P$, Siamese network $N_S$

**for** $i = 1$ ***to*** *iter1 do*
  Randomly sample from $D_s$ to obtain the mini-batch $D_S^m$;
  Train $N_P$ with $D_S^m$ using Adam optimizer by minimizing $L_C$;
**end for**
**for** $i = 1$ ***to*** *iter2 do*
  Constitute training data $D_{S,T} = [D_S^m \times D_T^n]$;
  Randomly sample from $D_{S,T}$ to obtain the mini-batch $D_{S,T}^k$;
  Construct $N_S$ using with $N_P$;
  Train $N_S$ with $D_{S,T}^k$ using Adam optimizer by minimizing $L_{total}$;
**end for**
**Return** $N_S$

**Figure 13.** Pseudo-code for training the proposed MAFS algorithm

As a result, based on point-wise comparison with latent vectors enables that 1) Initializing the network weights using a pre-trained model, and 2) Transferring the

fault-information of simple motions into complex motions to learn the shared cross-domain features from ripples. Also, we show that the proposed method can effectively transfer the fault-related information. A detailed description of the point-wise comparison is shown in Figure 14. In this case, blue-dot means normal data, and navy-dot means fault data, respectively. The dotted line is the $d_{Euc}$ on the embedding space, and blue-dot line corresponds to $L_{SA}$ for learning domain similarity between each normal and fault data of different domains. The gray-dot line corresponds to $L_C$ for classifying between different labels in the source domain. Finally, the black-dot line corresponds to $L_S$ for maximizing the $d_{Euc}$ with keeping the distance of $d_{Sep}$ between different labels in different domains.

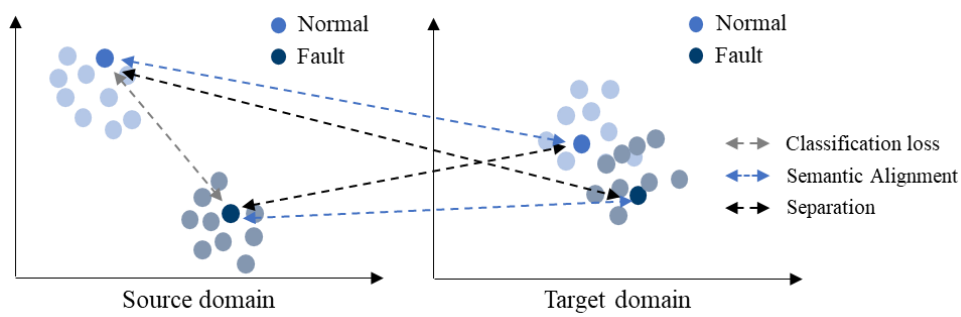


**Figure 14.** Pairwise comparisons for different domains on the latent space

# Chapter 4. Experimental Validations

In this chapter, the testbed setup is firstly discussed, and two case studies of the complex motions 1 and 2 are shown. Various comparative studies are performed to verify the effectiveness of the proposed method. In chapter 4.1, the industrial robot testbed that used to acquire the torque signals is presented. And in chapter 4.2, the pre-trained model generation was discussed. Finally, in chapter 4.3, the fault detection performances of each complex motions will be drawn.

## 4.1 Experimental Settings

The data used for the analysis was acquired from the robot controller with torque data of each joint and corresponding speed profiles from a controller described in Figure 15. For articulated robots, the number of DOF which corresponds to the number of joints determines how freely the robot arm can move. Conventionally, the industrial robot, which has six joints to perform sophisticated tasks like a human arm, is described in Figure 16. Industrial robots are composed of links and joints, and drivetrain components are coupled in each joint. While executing the specific tasks, the end-tip contour of an industrial robot requires the working path, and the rotating angles and speeds of each joint are determined by such a working path. The working path of each joint is programmable with a starting point, several via points and an endpoint. So, the motion starts from a starting point, undergoes several via points, and ends when the motion arrived at an endpoint.

In this thesis, the term *simple motion* means the unit-axis accelerates from a starting point with a specific posture, rotates with constant angular speed, and

decelerates to the next via point from the previous via point. For trajectory generation, a linear segment with parabolic blends shape is used. Detailed explanations will be covered in the next chapter 4.2. Also, the term *complex motion* means that the working path of a real-industrial welding motion. Compared to the simple motion, while executing complex motion, the joints of the robot rotate simultaneously with constant angular speed and transient angular speed on a welding cycle composed of the complicated working path. In this thesis, the complex motions are configurated as following two motions. *Complex 1 motion* means a welding motion in a standard speed state that contains constant speed region, and *complex 2 motion* means a welding motion in a fast speed state that includes only a transient speed region. Detailed explanations will be covered in the next chapter 4.3.

The proposed method was validated using a 6-DOF industrial robot testbed with 80kgf of payload described in Figure 16. The cycloidal gearbox was used in each joint of the industrial robot, and the faulty specimen of $4^{th}$ joint is brought from real-industrial sites, which is frequently used for almost all movements for high precision control and takes the role of the wrist part of the industrial robot. The dataset is described in Table 2, and acquired with different temperatures and operating speed conditions under the design of experimental (DOE) method. In addition, the normal and fault data were obtained without imbalance for all cases. For deep network training, we used CPU of Intel i7-8250U, 64Gb RAM, and one GPU of NVIDIA GEFORCE RTX 2080 Ti. The code is constructed with tensorflow2.0 using python.

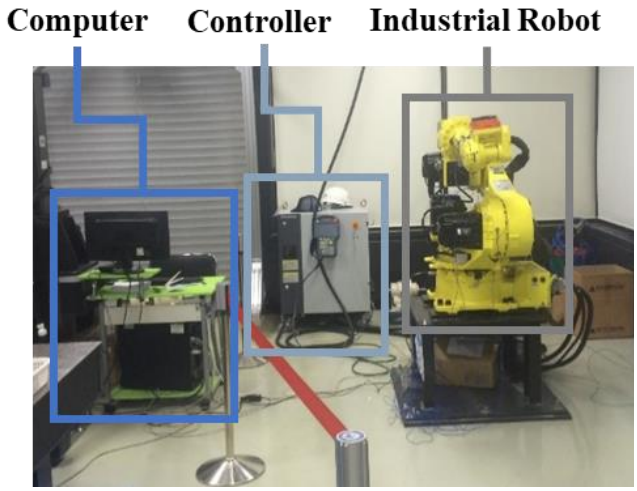


**Figure 15.** The overall configuration of the industrial robot testbed with a computer, and controller

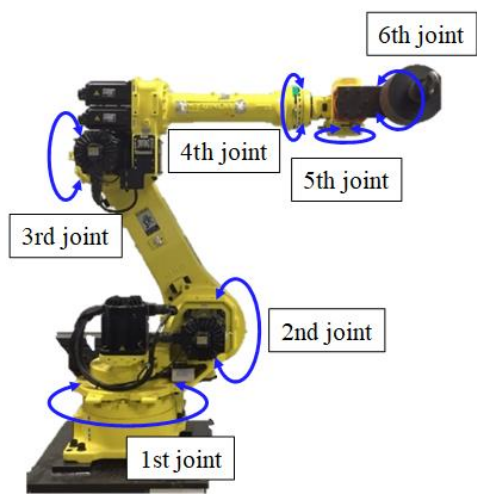


**Figure 16.** Industrial robot testbed of 6-DOF articulated robot [2]

**Table 2**. Dataset compositions of the source and the target domain

| Motions | Temperatures (℃) | Operating Speeds | # of datasets |
|---|---|---|---|
| Simple(1-axis) | 40, 50, 60 | 40, 60, 80, 100 | 24/24 |
| Complex1(6-axis) | 40, 50, 60 | - | 9/9 |
| Complex2(6-axis) | 40, 50, 60 | - | 6/6 |

## 4.2 Pre-trained Network Generation

To construct a robust diagnostic network, data is acquired with the variants of operating speeds and temperatures, where operating speeds are range from 40%, 60%, 80%, 100% of the full-angular speed, and temperatures range from 40℃, 50℃, 60℃. The parameters of the convolutional and pooling layers are shown in Table 2. And Adam [38] optimizer is applied to train the Res-CNN model as it is a straightforward, memory-saving, and computationally effective. The number of total samples is divided from the whole data used to train the network. For the training dataset, 50% is used composed of 3457 samples for normal and 3457 samples for fault, and validation data is the same as training data. In Table 2, the number of data acquired from the simple motion is noted. Also, as can be seen from the speed profile in Figure 17, the amplitude of the torque data changes in the chapter where the speed changes.

In Figure 18, a convergence plot while training the network is composed of the 3 candidate networks of Dense, one-dimensional (1D) CNN, and the proposed network. First, a Dense network did not converge while training due to low feature extraction performance. Second, a 1DCNN network composed of three conv-activ-maxpool

blocks converged lately because of the slow gradient propagation speed. On the other hand, the proposed Res-1DCNN network converged very fast and stabilized after almost 10 epochs.

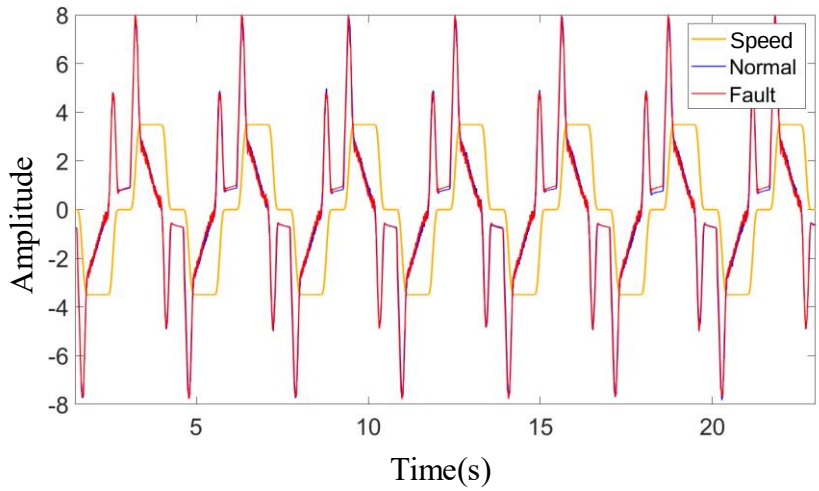


**Figure 17.** Torque data with speed profile on the simple motions

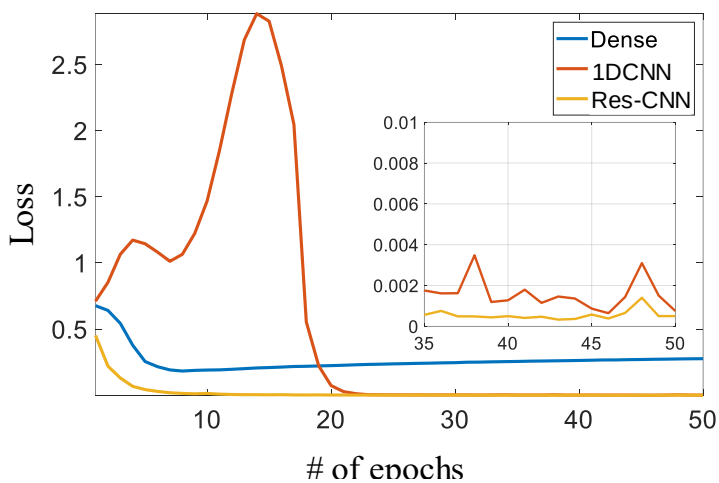


**Figure 18.** Convergence plot of validation loss on (a) Dense, (b) 1DCNN, (c) Proposed Res-CNN network

After training, the visualization of the classification performance via Res-CNN is shown using t-SNE plot in Figure 19. t-SNE converts the distance between data into stochastic probability and uses it for embedding, and expresses the manifold of high-dimensional data as a two-dimensional map by learning neighbor structures between data represented by high-dimensional embedding vectors.

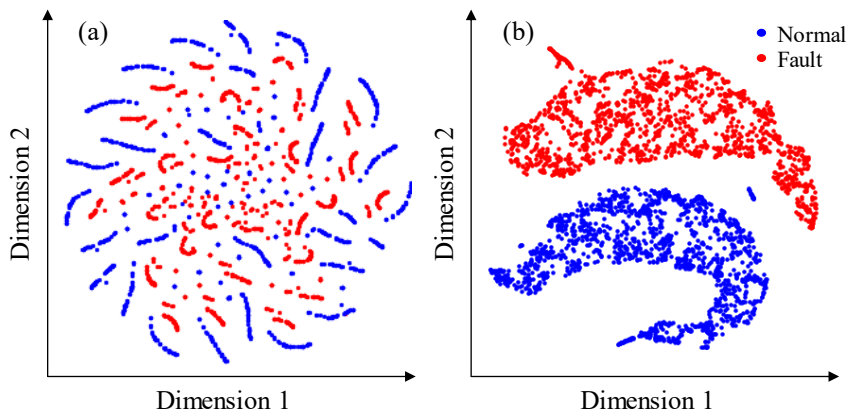


**Figure 19.** Visualization of embedding space of simple motions with t-SNE. Where blue dots mean the normal samples, and red dots mean the fault samples of total validation data.

## 4.3 Motion-Adaptation with Few-Shot Learning

In this chapter, the performance of the proposed method is examined by conducting two case studies with some of the comparative models. When training a network, only one-cycle of normal and fault data is used for setting a few-shot learning problems. And the performance of each model was evaluated using the average values of accuracies on samples of cycle data on the test dataset. Also, as a result of some ablation studies, the proposed method showed superior performances over other methods on such few-shot learning conditions.

**Without transfer learning.** Based on the same network used in the source domain, a network trained from scratch from the other target domain was applied to evaluate the performance. So, it is a measure of how different the discrepancy of data distributions on each complex motion.

**Transfer learning with fine-tuning.** For transfer learning, the pre-trained weight was frozen up to the first block of the model, and the training was performed with the trainable subsequent layers.

**Proposed MAFS with 1DCNN.** To validate the performance of MAFS by using 1DCNN as a feature extractor, a feature extractor is switched as 1DCNN to confirm the effectiveness of BSDA.

**Proposed MAFS with Res-CNN w/o $L_{SA}$**. To investigate the performance of the network according to the loss terms, modified loss term that without $L_{SA}$ is used while BSDA procedure.

**Proposed MAFS with Res-CNN w/o $L_S$**. Similar to above case, modified loss term that without $L_S$ is used while BSDA procedure.

**Proposed MAFS with Res-CNN.** By switching the backbone feature extractor as Res-CNN, the effects of residual propagation on information flow on SDA is

discussed, which is the proposed architecture on this thesis.

**Baseline.** By training the Res-CNN from scratch with sparse data in the target domain, it is conducted to check the learning performance for test data of complex motions.

### 4.3.1 Case Study 1: Standard speed Welding Motion

In this case study, a case of standard speed welding motion data is used. In one-cycle, each normal and fault data is composed of 200 samples. In Figure 20, the angular speed and torque profile of the 4th joint for one cycle are shown. There are three major regions of the constant speed region between 8(s) to 10(s), 13(s) to 14(s), and 14(s) to 16(s). Data is acquired with the variants of temperatures range from 40℃, 50℃, 60℃. The number of samples is divided from the whole data, only one cycle data of normal and the fault is randomly selected to train the network. Test data is the remainder of the whole dataset except the training dataset.

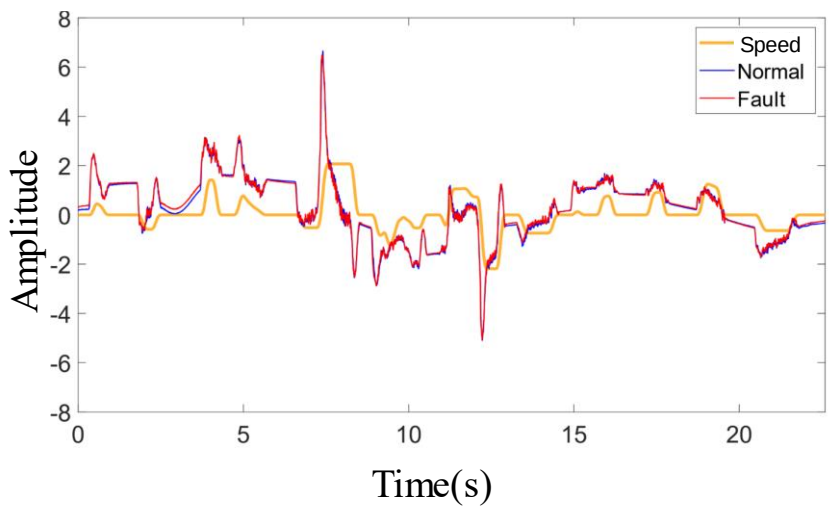


**Figure 20.** Torque data with speed profile on the complex 1 motion

Next, the results of the fault detection performances are summarized in Table 3. The accuracy means the averaged accuracies of each cycle expressing the variations with -sigma to +sigma. For a pre-trained case that trained only with a baseline model on the complex 2 motion, the classification accuracy is low as a pre-trained weight from the complex 1 motion is not fit for complex 2 motion. For a fine-tuned case, which uses weights of a pre-trained network on the shallow layer trained on the complex 2 motion, accuracy is greatly improved as 67.1%, but the performance is still low as the low-level features extracted from a shallow layer are different compared to the simple motion case. So, it implies that even if the network is fine-tuned on the simple motions, overfitting has occurred on sparse data on the complex motions, so fine-tuning is not a valid choice.

On the other hand, based on the proposed MAFS method, the classification accuracy using 1DCNN as a feature extractor is 61.8%, which is lower than the fine-tuned case. Also, proposed MAFS with Res-CNN w/o $L_{SA}$ and $L_S$ each shows improved performance as 89.7% and 88.1%, respectively, but proposed MAFS with Res-CNN using total loss showed the most superior results than other methods as 95.3%. Especially, the accuracy is drastically improved compared to the MAFS with 1DCNN case because the information flow is reformed based on residual propagation during motion-adaptation. So, high-dimensional feature representations of the ripples could effectively be learned through passing the deep-layers. Furthermore, the baseline model, which trained only using the target domain data, showed similar performance compared to the proposed MAFS with Res-CNN.

In Figure 21, visualizations of embedding space via proposed MAFS with Res-CNN are shown as (a) before adaptation, and (b) after adaptation, where blue dots mean the normal samples and red dots mean the fault samples. In (a) of Figure 21,

embedding space of a pre-trained network using only unit-axis motions is shown. Properly clustered distributions are composed of the normal and fault samples of the unit-axis motion. On the other hand, in (b) of figure 21, the overlapped distributions, which originated from the domain discrepancy, are well distinguished after adaptation. Therefore, the cross-domain features on torque ripples were effectively learned on the motion-adaptation procedure only using few-shot samples on the target domain, while maintaining the classification accuracy.

Furthermore, using 1d guided grad-CAM, visualization of localization effects on the signal is shown in Figure 22. In each plot, it shows the highlighted regions localized as informative regions to classify each class. As we can see, the non-adapted network is localized to a specific part of the ripple, resulting in low fault detection performance, whereas the adaptive network sees the shape of the ripple in a broader range and detects the fault with higher accuracy. Also, the highlighted region after adaptation means the transferable region, which means the shared region with the source domain.

**Table 3.** Comparative results of the fault detection performances on the complex 1 motion test data

| **Methods** | **Accuracy(%)** |
|---|---|
| Pre-trained(2-1) | $46.5 \pm 9.2$ |
| Fine-tuned(2-1) | $67.1 \pm 7.6$ |
| Proposed MAFS(1DCNN) | $61.8 \pm 5.5$ |
| Proposed MAFS(Res-CNN) w/o $L_{SA}$ | $89.7 \pm 5.9$ |
| Proposed MAFS(Res-CNN) w/o $L_S$ | $88.1 \pm 5.5$ |
| **Proposed MAFS(Res-CNN)** | **$95.3 \pm 4.4$** |
| Baseline | $93.9 \pm 4.7$ |

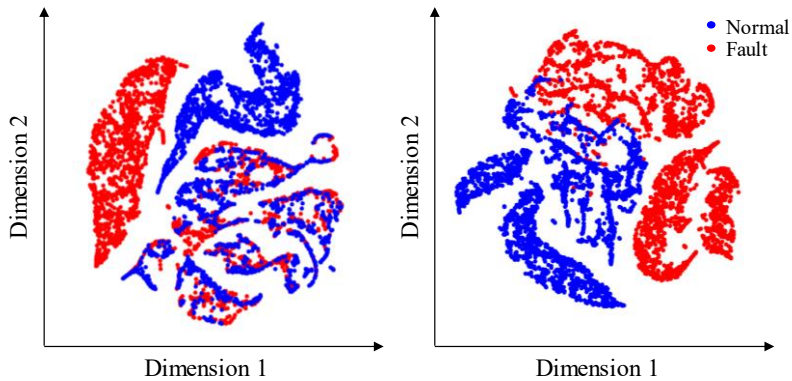


**Figure 21.** Visualization of embedding space with t-SNE with (a) before adaptation, and (b) after adaptation

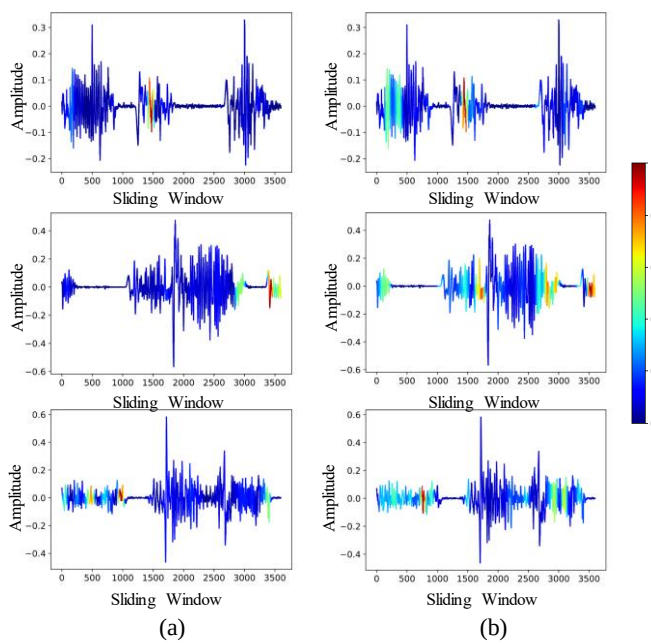


**Figure 22.** Visualization of 1D guided grad-CAM on complex 1 motion (a) before MAFS with Res-CNN, and (b) after MAFS with Res-CNN

### 4.3.2 Case Study 2: Fast Speed Welding Motion

In this case study, the same methodology of case study 1 is used, but the speed profile is quite different because it includes only the transient regions. In one-cycle, each normal and fault data is composed of 180 samples. In Figure 23, the angular speed and torque profile of the 4th joint for one cycle are shown, and there is no constant speed region on a cycle. For the training dataset, only one cycle data of normal and the fault is randomly selected, and test data is the remainder of the whole dataset except the training dataset. In Table 2, the number of data acquired from the complex motion 2 is noted, and results of the comparative fault detection performance are summarized in Table 4. The accuracy means the averaged accuracies of each cycle expressing the variations with -sigma to +sigma. Compared to the case study 1, the samples of the one-cycle data are less and the similarity between the source and the target domain is low.

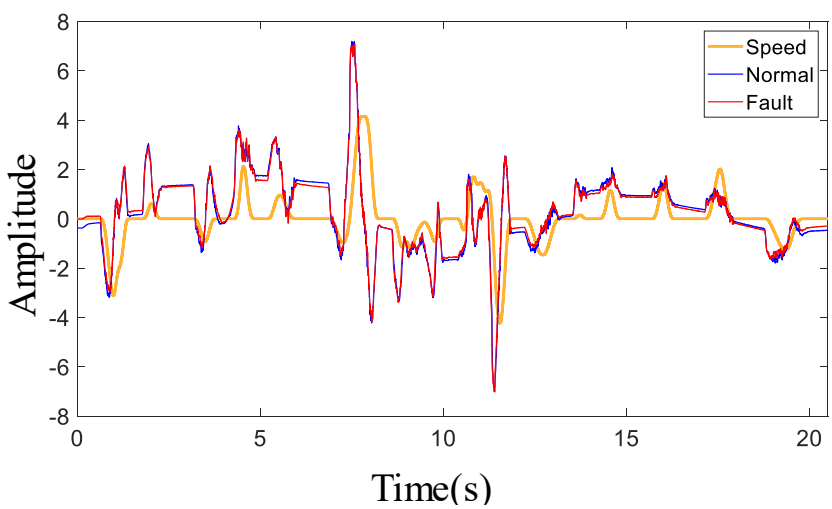


**Figure 23.** Torque data with speed profile on the complex 2 motion

Based on the proposed MAFS method, the classification accuracy using 1DCNN as a feature extractor is 79.6%, which is higher than the fine-tuned case. Also, proposed MAFS with Res-CNN w/o $L_{SA}$ and $L_S$ each shows improved performance as 75.6% and 86.4%, respectively, but proposed MAFS with Res-CNN showed the most superior results than other methods as 92.8%. Especially, the accuracy is drastically improved compared to the MAFS with 1DCNN case. Compared to the case study 1, the baseline model trained only using the target domain data, showed significantly low accuracy because of the low learning performance.

In Figure 24, visualizations of embedding space via proposed MAFS with Res-CNN are shown as (a) before adaptation, and (b) after adaptation, where blue dots mean the normal samples and red dots mean the fault samples. In (a) of Figure 24, properly clustered distributions are composed of the normal and fault samples of the source domain. On the contrary, in (b) of Figure 24, the overlapped distributions of target motions become well-distinguished after motion-adaptation. Therefore, similar to the above case study 1, the cross-motion features on torque ripples are effectively learned through the motion-adaptation procedure using few-shot samples on the target domain.

Furthermore, using 1d guided grad-CAM, visualization of localization effects on the signal is shown in Figure 25. As a result, the network localized to informative regions while training. As we can see, the non-adapted network is localized to a specific part of the ripple, resulting in low fault detection performance, whereas the adaptive network sees the shape of the ripple in a broader range and detects the fault with higher accuracy. Also, the highlighted region after adaptation means the transferable region, which means the shared region with the source domain.

**Table 4.** Comparative results of the fault detection performances on the complex 2 motion test data

| Methods | Accuracy(%) |
|---|---|
| Pre-trained(2-1) | $50.2 \pm 1.2$ |
| Fine-tuned(2-1) | $77.8 \pm 18.3$ |
| Proposed MAFS(1DCNN) | $79.6 \pm 7.7$ |
| Proposed MAFS(Res-CNN) w/o $L_{SA}$ | $75.6 \pm 7.5$ |
| Proposed MAFS(Res-CNN) w/o $L_S$ | $86.4 \pm 9.3$ |
| **Proposed MAFS(Res-CNN)** | $\mathbf{92.8 \pm 8.2}$ |
| Baseline | $67.4 \pm 19.5$ |

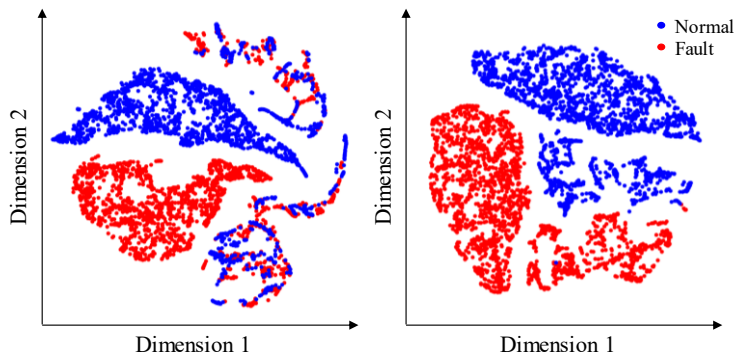


**Figure 24.** Visualization of embedding space with t-SNE of complex 2 motion.

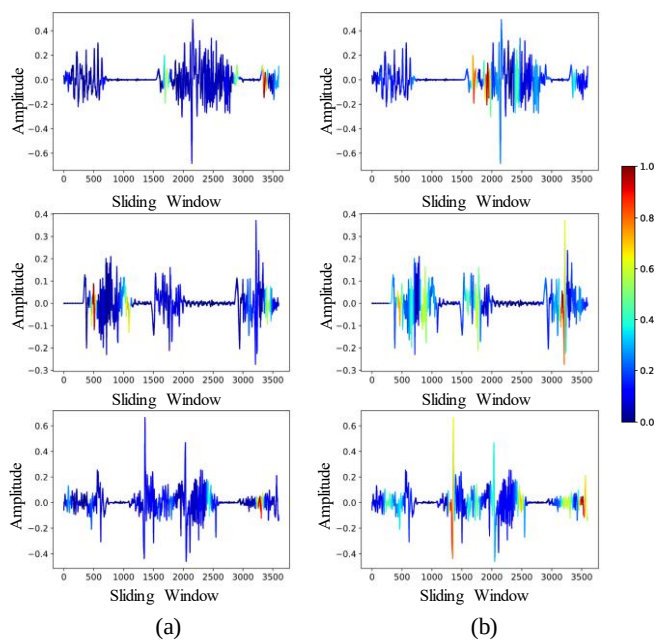


**Figure 25.** Visualization of 1D guided grad-CAM on the complex 2 motion. (a) before MAFS with Res-CNN, and (b) after MAFS with Res-CNN

# Chapter 5. Conclusion and Future Work

## 5.1 Conclusion

In this paper, we confirmed that the proposed MAFS with Res-CNN method showed superior performance to detect the faults on the complex motions. First, pre-processing minimizes the motion effects by extracting the torque ripples. Second, the proposed Res-CNN propagated the information of torque ripples efficiently. Third, even if the few-shot tasks, BSDA transfers motion-invariant features of the gearboxes effectively under constant and transient speed welding motions. Also, via 1D guided grad-CAM, it is shown that transferable local regions became broader after domain adaptation. In addition, the proposed method is generalized to resolve domain discrepancy for different robot motions through learning the shared physical mechanisms on the torque ripples under few-shot conditions. Therefore, this study can be used to quickly determine whether the faults of the gearbox of an industrial robot is has occurred or not on the complex motions both on constant and transient speeds.

## 5.2 Contribution

Based on the above procedures, this study has the following three contributions: (1) extracting torque ripples by pre-processing to minimize motion effects on the low-frequency band, (2) generating a pre-trained network for robust feature extraction by improved information flow via Res-CNN, (3) generalizing a few-shot fault detection method on industrial motions by motion-adaptive learning. The proposed method is expected to apply to other industrial motions such as assembly,

painting, inspection, and so on.

**Contribution 1: Extracting Torque Ripples By Pre-Processing To Minimize Motion Effects On The Low-Frequency Band**

The effects of the motion were minimized by pre-processing the acquired torque signal. Phases are matched through cross-correlation, and torque ripples are extracted through MAF. This pre-processing minimizes the influences of the low-frequency band due to robot motion independent of the health index. After then, input data as torque ripples is generated through a sliding window-based data augmentation.

**Contribution 2: Generating A Pre-Trained Network For Robust Feature Extraction By Improved Information Flow Via Res-CNN**

To pre-train the network, data were collected differently depending on the operating and environmental conditions according to the experimental design method. In addition, the proposed Res-CNN network showed the best feature representation performance due to attention effects and reformed information flow.

**Contribution 3: Generalizing a Few-Shot Fault Detection Method on Industrial Motions By Motion-Adaptive Learning**

A fault detection method through learning the motion-invariant features under complex motions is generalized using BSDA. This method is powerful in exploiting sparsely labeled target data to learn the domain-invariant knowledge by using point-wise comparisons of feature distribution distances and similarities. Furthermore, the performance is visualized in the embedding space through t-SNE, and the transferable region is highlighted through 1D guided grad-CAM. As a result, motion-

adaptation minimized the physical mismatches or inconsistencies on the torque ripples on the different motions.

## 5.3 Future Work

In this chapter, several future works are summarized. First, to verify the effectiveness of the proposed method, adjusting the number of samples for few-shot tasks is required. Also, changes in the model performance when the data scarcity is different by varying the sliding window interval are not dealt. Second, architecture studies of DA are not enough to compare, so there is a possibility for further improvement in the detection performance by adopting advanced divergence measures of extracted features such as MMD [38], CORAL [39], GCN [40]. Finally, the case where the faults occurred on the $4^{th}$ axis was only dealt in this thesis, but it is expected that the proposed method is applicable to detect the gearbox faults not only for the $4^{th}$ axis but also other axes. In addition, the proposed method based on the torque ripples is applicable to other types of articulated robot systems such as cooperative robots, and so on.

# Bibliography


[1] A. W. K. To, G. Paul, and D. Liu, "A comprehensive approach to real-time fault diagnosis during automatic grit-blasting operation by autonomous industrial robots," *Robotics and Computer-Integrated Manufacturing,* vol. 49, pp. 13-23, 2018.

[2] Y. Kim, J. Park, K. Na, H. Yuan, B. D. Youn, and C.-s. Kang, "Phase-based time domain averaging (PTDA) for fault detection of a gearbox in an industrial robot using vibration signals," *Mechanical Systems and Signal Processing,* vol. 138, p. 106544, 2020.

[3] F. Cheng, Y. Peng, L. Qu, and W. Qiao, "Current-based fault detection and identification for wind turbine drivetrain gearboxes," *IEEE Transactions on Industry Applications,* vol. 53, no. 2, pp. 878-887, 2016.

[4] F. Cheng, A. Raghavan, D. Jung, Y. Sasaki, and Y. Tajika, "High-Accuracy Unsupervised Fault Detection of Industrial Robots Using Current Signal Analysis," in *2019 IEEE International Conference on Prognostics and Health Management (ICPHM)*, 2019: IEEE, pp. 1-8.

[5] A. C. Bittencourt, K. Saarinen, and S. Sander-Tavallaey, "A Data-driven method for monitoring systems that operate repetitively-applications to wear monitoring in an industrial robot joint," *IFAC Proceedings Volumes,* vol. 45, no. 20, pp. 198-203, 2012.

[6] I.-M. Kim, H.-S. Kim, and J.-B. Song, "Design of joint torque sensor with reduced torque ripple for a robot manipulator," *International Journal of Precision Engineering and Manufacturing,* vol. 13, no. 10, pp. 1773-1779, 2012.

[7] R. Mitra and Y. Sozer, "Torque ripple minimization of switched reluctance motors through speed signal processing," in *2014 IEEE Energy Conversion Congress and*

*Exposition (ECCE)*, 2014: IEEE, pp. 1366-1373.

[8] C. N. N. Rao and G. Sukumar, "Design and analysis of torque ripple reduction in Brushless DC motor using SPWM and SVPWM with PI control," *European Journal of Electrical Engineering,* vol. 20, no. 1, p. 7, 2018.

[9] T. Ohnuma, S. Doki, and S. Okuma, "Analysis of Torque Ripple Caused by Current Signal Injection on the Maximum Torque Control Frame for Sensorless Control," in *The 2010 International Power Electronics Conference-ECCE ASIA-*, 2010: IEEE, pp. 1383-1387.

[10] B. Niu, "Parasitic torque pulsation suppression in drive train of flexible joint industrial robot," in *IEEE ISR 2013*, 2013: IEEE, pp. 1-5.

[11] W.-S. Lin, Y.-P. Shih, and J.-J. Lee, "Design of a two-stage cycloidal gear reducer with tooth modifications," *Mechanism and Machine Theory,* vol. 79, pp. 184-197, 2014.

[12] K. Wu, C. Krewet, and B. Kuhlenkötter, "Dynamic performance of industrial robot in corner path with CNC controller," *Robotics and Computer-Integrated Manufacturing,* vol. 54, pp. 156-161, 2018.

[13] V. Sathish, M. Orkisz, M. Norrlof, and S. Butail, "Data-driven gearbox failure detection in industrial robots," *IEEE Transactions on Industrial Informatics,* 2019.

[14] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE transactions on pattern analysis and machine intelligence,* vol. 35, no. 8, pp. 1798-1828, 2013.

[15] F. Wang *et al.*, "Residual attention network for image classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 3156-3164.

[16] M. Lin, Q. Chen, and S. Yan, "Network in network," *arXiv preprint arXiv:1312.4400,*

2013.

[17] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 807-814.

[18] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167,* 2015.

[19] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770-778.

[20] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556,* 2014.

[21] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1-9.

[22] X. Wang, L. Li, W. Ye, M. Long, and J. Wang, "Transferable attention for domain adaptation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019, vol. 33, pp. 5345-5352.

[23] H. Kim and B. D. Youn, "A new parameter repurposing method for parameter transfer with small dataset and its application in fault diagnosis of rolling element bearings," *IEEE Access,* vol. 7, pp. 46917-46930, 2019.

[24] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," *arXiv preprint arXiv:1409.7495,* 2014.

[25] Y. Ganin *et al.*, "Domain-adversarial training of neural networks," *The Journal of Machine Learning Research,* vol. 17, no. 1, pp. 2096-2030, 2016.

[26] S. Motiian, M. Piccirilli, D. A. Adjeroh, and G. Doretto, "Unified deep supervised domain adaptation and generalization," in *Proceedings of the IEEE International*

*Conference on Computer Vision*, 2017, pp. 5715-5725.

[27] X. Li, W. Zhang, and Q. Ding, "Cross-domain fault diagnosis of rolling element bearings using deep generative neural networks," *IEEE Transactions on Industrial Electronics,* vol. 66, no. 7, pp. 5525-5534, 2018.

[28] W. Lu, B. Liang, Y. Cheng, D. Meng, J. Yang, and T. Zhang, "Deep model based domain adaptation for fault diagnosis," *IEEE Transactions on Industrial Electronics,* vol. 64, no. 3, pp. 2296-2305, 2016.

[29] J. Jiao, M. Zhao, J. Lin, and C. Ding, "Classifier Inconsistency based Domain Adaptation Network for Partial Transfer Intelligent Diagnosis," *IEEE Transactions on Industrial Informatics,* 2019.

[30] S. Motiian, Q. Jones, S. Iranmanesh, and G. Doretto, "Few-shot adversarial domain adaptation," in *Advances in Neural Information Processing Systems*, 2017, pp. 6670-6680.

[31] R. Zhang, T. Che, Z. Ghahramani, Y. Bengio, and Y. Song, "Metagan: An adversarial approach to few-shot learning," in *Advances in Neural Information Processing Systems*, 2018, pp. 2365-2374.

[32] J. Kim, T. Kim, S. Kim, and C. D. Yoo, "Edge-labeling graph neural network for few-shot learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 11-20.

[33] T. Hu, T. Tang, R. Lin, M. Chen, S. Han, and J. Wu, "A Simple Data Augmentation Algorithm and A Self-Adaptive Convolutional Architecture for Few-Shot Fault Diagnosis under Different Working Conditions," *Measurement,* p. 107539, 2020.

[34] S. Ma, F. Chu, and Q. Han, "Deep residual learning with demodulated time-frequency features for fault diagnosis of planetary gearbox under nonstationary running conditions," *Mechanical Systems and Signal Processing,* vol. 127, pp. 190-

201, 2019.

[35] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *European conference on computer vision*, 2016: Springer, pp. 630-645.

[36] W. Zhang, G. Peng, C. Li, Y. Chen, and Z. Zhang, "A new deep learning model for fault diagnosis with good anti-noise and domain adaptation ability on raw vibration signals," *Sensors,* vol. 17, no. 2, p. 425, 2017.

[37] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, 2005, vol. 1: IEEE, pp. 539-546.

[38] A. Rozantsev, M. Salzmann, and P. Fua, "Beyond sharing weights for deep domain adaptation," *IEEE transactions on pattern analysis and machine intelligence,* vol. 41, no. 4, pp. 801-814, 2018.

[39] B. Sun and K. Saenko, "Deep coral: Correlation alignment for deep domain adaptation," in *European conference on computer vision*, 2016: Springer, pp. 443-450.

[40] X. Ma, T. Zhang, and C. Xu, "GCAN: Graph Convolutional Adversarial Network for Unsupervised Domain Adaptation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8266-8276.

[41] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 618-626.

# Appendix A. 1D Guided Grad-CAM

Nowadays, explainable AI (XAI) has been receiving lots of attention, which is used for interpreting and understanding the deep learning architectures based on final-results and the validity of the derivation process. Also, for PHM, it is important to know the most probable region to classify each class of healthy states to prevent the wrong localization problem of the deep network. It is because the wrong decision caused by AI could induce disasters in the real-world. So, people were would like to see how the proposed method highlights an informative region. In PHM, much data is acquired by sensor signals, which is 1D not a 2D, so 1D guided-grad CAM is needed to be applied. In this thesis, 1D guided grad-CAM was implemented with simple signal-processing techniques for the binary case, and it is further applicable to other applications such as bearing, gearbox, and so on. In Figure 26, Net 1 represents a pre-trained network before adaptation, and Net 2 represents an adapted network after the MAFS method by adjusting the weights through a constructed Siamese network.

By Selvaraju et al. [41], the concept of guided-grad CAM was introduced. As grad-CAM detects roughly the parts related to the class on the image, but the details of those parts were difficult to catch due to the effects of bilinear upsampling. So, by element-wise multiplication of guided back-propagation and grad-CAM, it showed a more effective end-to-end localization method. In more detail, for example, in the classification problem, the expressions of the guided grad-CAM are as follows. The $y^c$ score for each class of the softmax step and the gradient for the $k^{th}$ layer on CNN structure where feature map is $A^k$. So, the importance weights could be expressed as follows.

$$\alpha_k^c = \frac{1}{Z}\Sigma_i\Sigma_j \frac{\partial y^c}{\partial A_{i,j}^k} \tag{19}$$

In equation (20), after multiplying the importance weights with each feature map, grad-CAM heat-map could be calculated as passing by ReLU of linear summation, since the interested region of the class is in the effect of positive values.

$$L_{Grad-CAM}^c = \mathrm{ReLU}\left(\Sigma_k \alpha_k^c A^k\right) \tag{20}$$

By combining with the above heatmap and guided back-prop in Figure 26, we can get the guided grad-CAM. In this thesis, the method of visualizing a 1D signal with grad-CAM is also considered. The signal envelope is extracted by applying the Hilbert transform and smooth it with a low pass filter. After then, 1D guided-grad CAM is plotted with line intensity according to the amplitude of torque ripples.

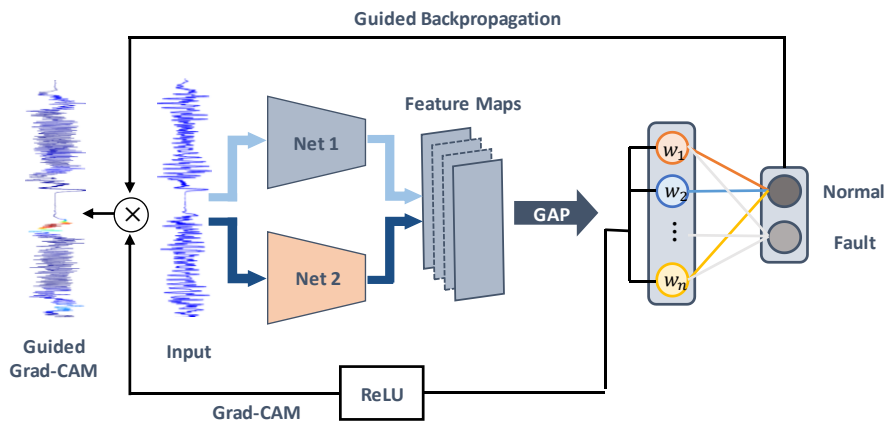


**Figure 26.** 1D Guided gradient-weighted class activation map

# 국문 초록

오늘날 산업용 로봇은 일관된 정밀도와 정확성으로 반복적인 작업을 수행할 수 있는 자동화된 제조 공정의 핵심 장비이다. 그러나, 산업용 로봇에 고장이 발생하면 생산 라인의 예기치 않은 종료를 야기하여 많은 경제적 손실이 발생할 수 있으므로 고장을 감지하는 것이 중요하다.

산업용 로봇의 주 동력 전달 요소 중 하나인 기어박스에는 높은 토크 부하가 걸리기 때문에 고장이 자주 발생하며, 기어박스에 고장이 발생하면 토크 신호의 진폭 및 주파수가 변조되어 토크 신호의 특성이 달라지게 된다. 이전의 여러 연구에서 토크 신호를 통한 산업용 로봇의 고장 감지 방법들이 제안되었지만, 다양한 환경 및 운영 조건에서 고장 관련 특징을 추출하고 산업 현장에서 사용되는 복잡한 동작에서 고장을 감지하는 것은 여전히 어려운 과제이다.

이러한 어려움을 해결하기 위하여, 본 논문에서는 토크의 리플을 사용하여 1차원 잔차-컨볼루션 신경망(Res-CNN)과 이진 감독 도메인 적응을 통한 산업용 로봇 기어박스의 동작 적응형 퓨샷 고장 감지 방법을 제안한다. 제안된 방법의 전체적인 절차는 다음과 같다. 먼저, 취득된 토크 신호에 이동 평균 필터링을 적용하여 데이터의 경향성을 추출하고, 원 신호 및 필터링 된 신호와의 잔차로서 고주파 대역의 토크 리플을 얻는다. 둘째, 온도와 회전 속도와 같은 다양한 환경 및 운영 조건에서 잔차-컨볼루션 신경망을 통해 토크 리플의 상태를 분류한다. 이는, 1) 정상과 고장 토크 리플의 작은 차이를 효과적으로 구분하였고, 2) 어텐션에 의해 입력 데이터의 중요한 영역에 초점을 두어 학습하는 효과를 나타냈다. 셋째, 단순 동작으로 구성된 소스 도메인에서 사전

훈련된 네트워크로 샴 네트워크를 구축한 후, 이진 감독 도메인 적응을 통해 복잡 동작으로 구성된 대상 도메인에서 고장을 감지한다. 결과적으로, 1) 산업용 로봇의 관절 축 내에서 토크 리플 간의 교차되는 물리적 특성의 유사성을 학습하였으며, 2) 로봇이 복잡 동작을 수행하는 동안, 동작에 적응적으로 기어박스의 고장을 감지하는 데 효과적임을 확인하였다. 제안된 방법은 복잡한 동작에서 정상과 고장 상태의 사이클 데이터가 각각 하나씩만 있는 퓨샷 조건에서, 다른 딥러닝 기반 방법들보다 가장 높은 정확도로 고장을 감지할 수 있음을 보였다. 또한, 토크 리플에서 전이 가능한 영역은 1차원 가이드 된 경사 가중치 기반 클래스 활성화 맵을 통해 강조되었다.

제안된 방법의 효과는 자동차 제조 라인과 같은 실제 산업 분야에서 일반적으로 사용되는 일정 및 과도 속도의 다축 용접 동작의 실험 데이터를 통해 검증되었으며, 제안된 방법은 추후 검사, 도장, 조립 등의 다른 유형의 동작에서도 기어박스의 고장을 감지하는 데 적용 가능할 것으로 예상된다.

**주요어:** 산업용 로봇
기어박스 고장 감지
토크 리플
잔차-컨볼루션 신경망
이진 감독 도메인 적응
퓨샷 학습
일차원 가이드 된 경사 가중치 기반 클래스 활성화 맵

**학 번:** 2018-27498