Bearing Fault Detection based on Few-Shot Learning in Siamese Network

DAEHWAN LEE, JONGPIL JEONG
Department of Smart Factory Convergence,
Sungkyunkwan University,
2066 Seobu-ro, Jangan-gu, Suwon 16419,
REPUBLIC OF KOREA

Abstract: - This paper executes bearing fault diagnosis with little data through few-shot learning. Recently, deep learning-based fault diagnosis methods have achieved promising results. In previous studies, fault diagnosis requires numerous training samples. However, in manufacturing, it is not possible to obtain sufficient training samples for all failure types under all working conditions. In this work, we propose a Few shot learning-based rolling bearing fault diagnosis that can effectively learn with limited data. Our model is based on the siamese network, which learns to use the same or different class of sample pairs.

Key-Words: - Few shot learning, Siamese network, Fault detection, WDCNN(Wide-first-layer kernels CNN), Bearing

Received: September 19, 2021. Revised: October 28, 2022. Accepted: November 25, 2022. Published: December 31, 2022.

1 Introduction

Manufacturing competitiveness is important in the era of global competition and the fourth industrial revolution. Product quality and facility management are important for securing manufacturing competitiveness. It is difficult to manage production facilities on-site and most companies do not have facilities maintenance workers. In many cases, production is often stopped because the equipment is stopped until repair workers arrive. Stable production is impossible because of equipment failure. Unstable production creates several losses. Additionally, it can significantly affect the quality of the product and cause significant losses to the company.

Most equipment failures occur in rotating equipment and bearing damage is the number one cause of failure in rotating equipment. As the most essential component of rotating mechanical equipment, the condition of rolling bearings significantly impacts the entire facility and manufacturing line, [1], [2], [3].

If the bearing is damaged while the rotating machine is running, the machine or the entire equipment may jam or malfunction. The bearing defects are caused y complex working conditions and long-term operation, resulting in microcracks inside the bearing and then internal microcracks accumulate, gradually starting with surface breakage. It is possible to detect the initial defect of a bearing by grasping the accident condition of the bearing from its vibration signal of the bearing, [1], [3].

Previous bearing defect studies have undertaken CNN, [8], [10], RNN, [11], [12], and Auto-encoder, [13], [14]. Other than that, there was a GAN, [16], [17], [20] study. In the above work, many data-based and deep learning-based technologies have been applied to increase accuracy and reliability, but most technologies require large amounts of training data, such as vibration, sound, motor and current signals. However, obtaining sufficient data samples of good quality to train all failure-type classifications in actual manufacturing sites is difficult. Therefore, there is limited data in actual manufacturing sites, so it is necessary to use a more effective algorithm.

This paper proposes a bearing fault diagnosis method for siamese networks based on Few shot learning. We compare accuracy and parameters according to the number of blocks in the WDCNN model. The method was experimented on Case Western Reserve University (CWRU) data, [18]. The composition of this paper is as follows. Section 2 describes CWRU-bearing data, a few shot learning, and the siamese networks. Section 3 describes the few-shot learning-based bearing fault detection. Section 4 of the relevant study describes the experimental procedures and results. Finally, Section 5 presents the conclusion and future research.

2 Related Work

2.1 Few-Shot Learning

Few-shot learning was first addressed in the 1980s, [4]. Recently, Few-shot learning has made great progress in solving the data shortage problem, [5]. Few samples have been used for classification or regression. Few-shot learning can categorize data well with literally few samples. Few-shot learning differs from conventional supervised learning methods and does not generalize the training set to a test set. It is divided into training, support and query sets in all data. We train the model in the training set, and the goal of training is to learn the similarities and differences between objects.

2.2 Siamese Network

Siamese networks were first introduced in the early 1990s by Bromley and LeCun to solve signature verification as an image-matching problem, [4]. Fig 1 shows the Siamese network structure. Unlike ordinary CNN, they consist of two CNN models but the two models have the same structure, [6], [19].

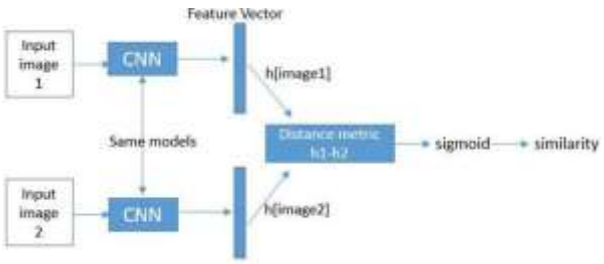


Fig. 1: Siamese network

That is why it is also called the twin network. Siamese networks usually use CNN models, but other models can be used as well.

A Siamese network takes two images as input and receives two images. The neural network outputs two feature vectors extracted from two input images and obtains a difference result vector between the two feature vectors. Multiple dense layers are used to process the resulting difference between vectors, and finally, a number between scalar 0 and 1 is obtained by applying a sigmoid activation function. The output is close to 1 if the two images are of the same class, and close to 0 if they are of different classes. A network that vectorizes and returns the similarity between two vectors (similarity in [0,1]).

The method to learn pairwise similarity is to train with positive and negative samples using a training set and randomly sample images from the training samples. If the two samples are the same, it is a positive sample, and if the two samples are different, it is a negative sample, [2], [7], [8], [9]. For example, there are the Husky, Elephant, Tiger, Parrot, Tea Class, and Tiger classes. First, select one sample from the tiger class and then select another type of tiger sample from the same class. Both samples are of the same class and are marked as 1. Conversely, other classes can also be selected. First, select one from the tiger class, then another sample from the other class. The two samples are of different classes, and if they are of different types, 0 is displayed.

2.3 Fault Detection

Fig 2 shows the bearing components. The basic components consist of an outer ring, an inner ring, a ball, and a cage (or retainer). A bearing is the basic element of the machine that supports the rotating shaft inside the machine and aids in the rotation of objects by reducing friction.

Fig. 2: Components of rolling bearing

As the most essential component of rotary mechanical equipment, the condition of the bearing has a significant impact on the entire facility. Bearing defects are caused by internal microcracks and surface damage due to the accumulation of microcracks. Fig 3 shows the bearing crack. Then,

Fig. 3: Bearing crack

due to the lack of a lubricant, contact between bearing surfaces and abnormally excessive external force is applied to the bearing. Defects in bearings mainly appear in the inner diameter, outer diameter or ball, [10]. The bearing defects are caused by complex working conditions and long-term operation, resulting in microcracks inside the bearing and then internal microcracks accumulate, gradually starting with surface breakage. It is possible to detect the initial defect of a bearing by grasping the accident condition of the bearing from its vibration signal of the bearing, [1], [3].

3 Few Shot Learning based Bearing Fault Detection

It is a Siamese network few-shot learning classification method based on our proposed WDCNN model. Fig 4 consists of three stages with the system structure presented in the paper. The data preparation stage (Top), the Few-shot-learning training & test (Middle), and the last is a siamese network structure based on the WDCNN model (Bottom).

Fig 4 shows the system structure. The first step is data preparation. To verify the performance, we selected 12k drive end-bearing fault data from the Case Western Reserve University (CWRU) bearing datasets as the experiment data. In the experiment, each sample is extracted from two vibration signals. Half of the vibration signal is used to generate the training sample and the other half is used to generate the test sample. Training samples were generated with a window size of 2048 points and 80 shift steps. The test samples are also created without overlap with the same window size.

The second is the training and testing phase. During training, the model is trained with a set of sample pairs of the same or different categories. The input is a sample pair with the same or different classes. The WDCNN model uses the two vibration signals prepared above as inputs. Each neural network outputs.

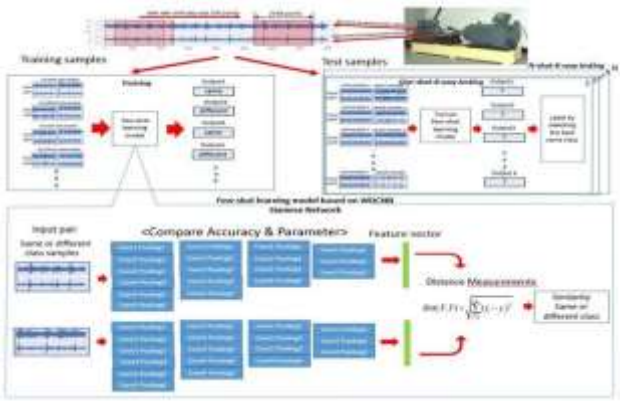


Fig. 4: System structure

Each neural network outputs two feature vectors extracted from two input images. After the output, the difference (distance) between the two feature vectors was obtained. After that, a dense layer was used to process differences between vectors. A number is obtained between 0 and 1 by applying the sigmoid activation function. Two similarities are measured and the output is that if the two images are of the same class, the output is close to 1 and the other class is close to 0. The difference between the target value and the predicted scalar is measured using the loss function.

The test is carried out using several one-shot k-way tests. In the N-shot K-way test, the model is provided with a support set of K different classes with N samples each. Determine which support set class the test sample belongs to. In this paper, since we proceeded with 5 Shot, the one-shot K-way test is repeated 5 times each time the support set is randomly selected from the training data. After five trials, five probability factors (P1, P2, P3, P4, P5) are calculated, and then the sum is calculated to obtain the largest value, [2]. The third is the structure of WDCNN according to the number of blocks in the Siamese network. 1 Block consists of a convolutional layer and a pooling layer. Each model consists of 6, 5, 4 or 3 blocks. Compare the accuracy of each model and the number of parameters.

Table 1 shows the structure of the few-shot learning model based on WDCNN

Table 1. Structure of few-shot learning model based on WDCNN

No	Layer Type	Kernel Size/Stride	Kernel Number	Output Size (width * Depth)	padding
1	Convolution1	64*1/16*1	16	128*16	same
2	Pooling1	2*1/2*1	16	64*16	valid
3	Convolution2	3*1/1*1	32	64*32	same
4	Pooling2	2*1/2*1	32	32*32	valid
5	Convolution3	3*1/1*1	64	32*64	same
6	Pooling3	2*1/2*1	64	16*64	valid
7	Convolution4	3*1/1*1	64	16*64	same
8	Pooling4	2*1/2*1	64	16*64	valid
9	Convolution5	3*1/1*1	64	6*64	valid
10	Pooling5	2*1/2*1	64	3*64	valid
11	Fully-connected	100	1	100*1	

It consists of 5 convolutional layers and a pooling layer, initially setting the kernel size to 64. One block means the sum of one convolution layer and one pooling layer, and the configuration of one block follows the size, stride, and padding of the kernel of con 5 in Table 1 and the size, stride, and padding of the kernel of pooling 5.

Block 3, 4, 5, 6 models have reduced number of blocks or added 1 block from the 5 blocks in the base model. When reducing the number of blocks, reduce sequentially from Conv5+Pooling5. For example, in the case of block4, it means conv4+pooling4 by reducing Conv5+Pooling5 in Table 1. When adding block counts use the same

kernel size, number and stride for conv5+pooling5. For example, for block6, conv6 +pooling6 is added after con5+pooling5.

4 Experiment and Results

4.1 Experiment Environments

Table 2 shows the experimental environment. The hardware used in this study consisted of an Intel Core i7- 8700k processor and GeForce GTX 3080ti. The software uses Windows, Tensorflow 2.4 and Python 3.6.

Table 2. System specification

Hardware Environment	Software Environment
CPU: Intel Core i7-8700K CPU@ 3.70GHZ Six-core	window, Tensorflow 2.4
GPU: NVIDIA Geforce GTX 3080ti	Python 3.6

In this paper, the CWRU dataset was used. The CWRU dataset is data collected for normal bearings, Drive end, and Fan end defects. Drive end was collected in samples measured 12k per second (12k – 12000 vibrations per second) and 48k per second (48k – 48000 vibrations per second) and Fan end in samples measured 12k per second (12k – 12000 vibrations per second). There are three types of bearing fault: Inner race, Outer race, and Ball and an independent data set exist according to the size of each bearing fault. Each fault size consists of 0.007 inches, 0.014 inches and 0.021 inches, respectively. For each failure size, 0-3 hp was configured. Outer Raceway Faults measured vibration for fault conditions at 3 o'clock, 6 o'clock and 12 o'clock positions, [12], [15].

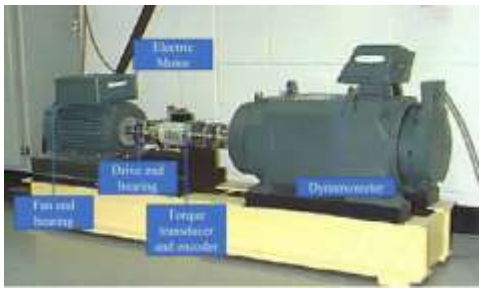


Fig. 5: Bearing simulator of CWRU

Fig 5 shows the bearing simulator of CWRU. The CWRU simulator is composed of the dynamometer, Electric motor, Drive end bearing, Fan end bearing and Torque transducer and encoder.
Table 3 shows the description of rolling bearing datasets. There are 10 types of fault labels, as shown in Table 3. Dataset A combined 660 tests of Load 1, 2 and 3 training and 25 tests to create 1980 training sets and 75 test sets.
In this experiment, the test set of dataset A is set as this and 60, 90, 120 and 200 samples are randomly sampled from the training samples of dataset A, respectively.

Table 3. Description of rolling bearing datasets

Fault Location	None	Ball			Inner Race			Outer Race			Load
Fault Diameter (inch)	0	0.007	0.014	0.021	0.007	0.014	0.021	0.007	0.014	0.021	
Fault Labels	1	2	3	4	5	6	7	8	9	10	
Dataset A	Train	1980	1980	1980	1980	1980	1980	1980	1980	1980	1,2,3
	Test	75	75	75	75	75	75	75	75	75	

4.2 Evaluation Metric

Accuracy is the most intuitive indicator. The problem, however, is that unbalanced data labels can skew performance**.** The equation for this parameter is:

$$\text{Accuracy} = \frac{|TP|+|TN|}{|TP|+|FP|+|FN|+|TN|} \quad (1)$$

The recall is the ratio of a class to what the model predicts as true among those that are actually true. The recall can be expressed by the following equation:

$$\text{Recall(sensitivity)} = \frac{|TP|}{|TP|+|FN|} \quad (2)$$

Precision is the proportion of what the model classifies as true that is actually true. Precision can be expressed by the following equation:

$$\text{Precision} = \frac{|TP|}{|TP|+|FP|} \quad (3)$$

The f1-score is the harmonic average of precision and recall. When the data labels are unbalanced, the performance of the model can be accurately evaluated. The f1 score can be expressed in the following equation:

$$\text{F1} - \text{score} = 2 * \frac{Precision*Recall}{Precision+Recall} \quad (4)$$

4.3 Results

In this paper, we proceeded with 60, 90, 120, and 200 samples from the training sample of dataset A, respectively and each experiment was set to a batch size of 32. All experiments used the most important

accuracies mentioned in the evaluation index. To cover compensate for the shortcomings of accuracy, an f-1 score was used. The experimental results are shown in Fig 5 and Fig 6 show the change in accuracy according to the number of blocks and Fig 5 shows the change in the parameters according to the number of blocks.

Fig 6 shows a graph of the change in accuracy with the number of blocks. The graph in Fig 6 shows the number of blocks 3, 4, 5 and 6 in the WDCNN model, respectively and the x-axis shows the number of samples 60, 90, 120 and 200. The blue graph represents block 3, the orange graph represents block 4, the gray graph represents block 5 and finally, the yellow graph represents block 6. It can be seen that the accuracy of Block 5 is high in most of the samples. However, it can be seen that the accuracy of block 4 is higher in samples 120 and 200.

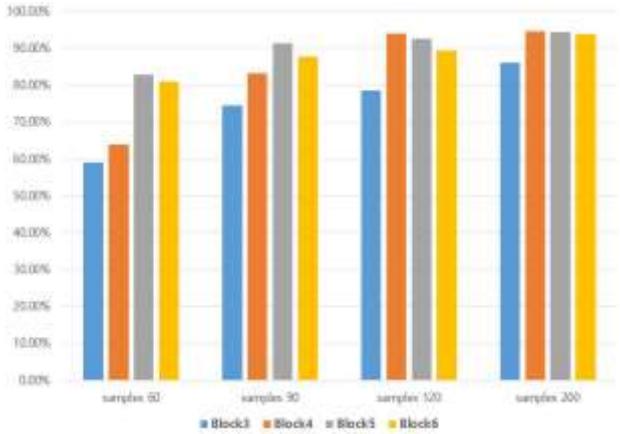


Fig. 6: Accuracy changes according to the number of blocks

Fig 7 shows the Parameter changes according to the number of blocks graph.

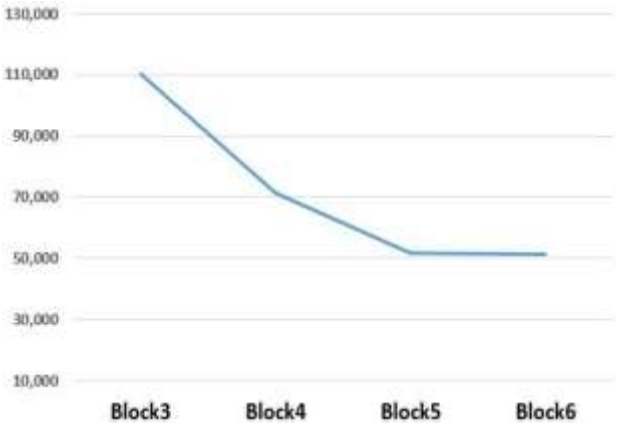


Fig. 7: Parameter changes according to the number of blocks

You can see that it decreases rapidly as you go through Block 3 and Block 4, and there is almost no change in Block 5 and Block 6. In Sample 120, block 4 is about 1% higher than block 5, but block 5 is more efficient because the number of block 5 parameters is 20,000 less than the number of block 4 parameters.

Table 4. Block accuracy & F1-score

Sample60	Block3	Block4	Block5	Block6
Accuracy	59.01	66.75	82.80	81.02
F1-score	58.28	68.72	78.90	79.58

Table 5. Block accuracy & F1-score

Sample90	Block3	Block4	Block5	Block6
Accuracy	74.47	83.22	91.37	85.42
F1-score	75.73	87.39	91.50	85.96

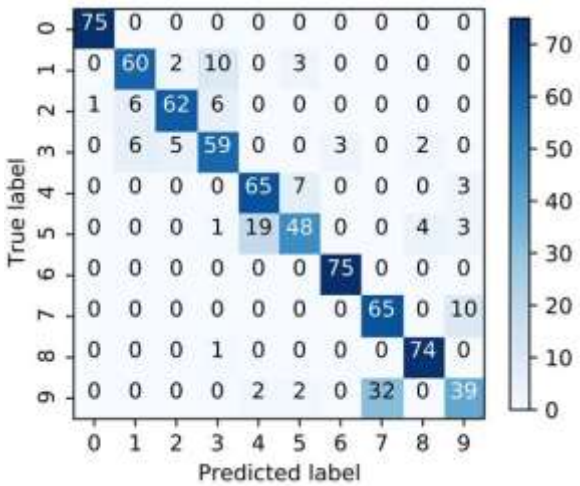
Table 6. Block accuracy & F1-score

Sample120	Block3	Block4	Block5	Block6
Accuracy	78.46	94.03	92.66	89.28
F1-score	78.10	94.25	78.28	87.00

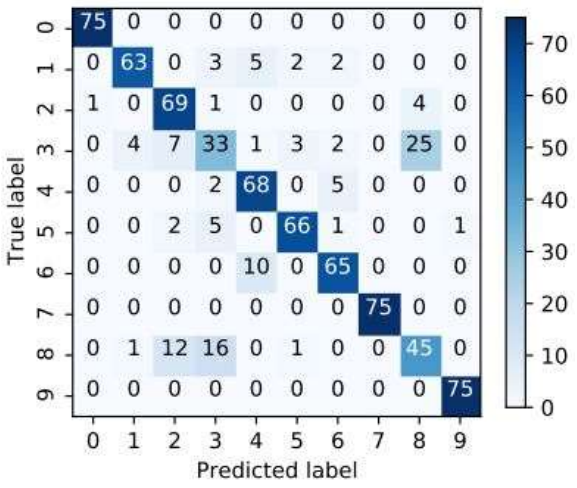
Table 7. Block accuracy & F1-score

Sample200	Block3	Block4	Block5	Block6
Accuracy	86.16	94.63	94.32	93.69
F1-score	85.97	94.43	90.83	88.87

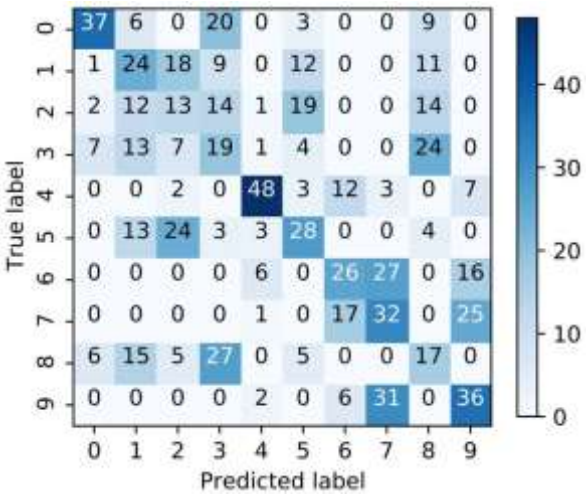
Tables 4, 5, 6, and 7 show the accuracy and f1-score. Samples 60, 90, and 200 do not drop significantly in the F1-score relative to accuracy. However, unlike blocks 3, 4, and 6 on sample 120, block 5 performs poorly due to its low f1 score compared to its accuracy.



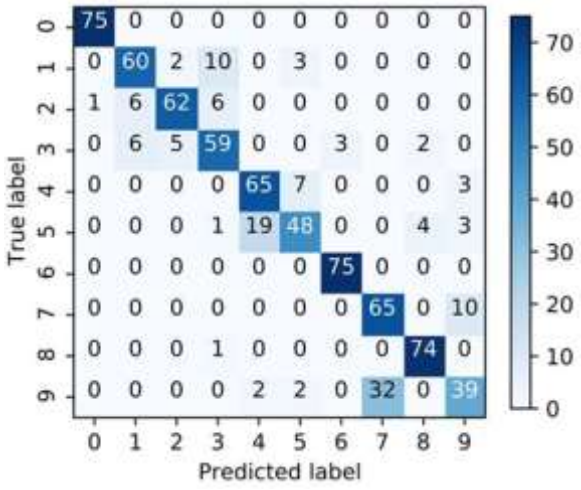
(a) Sample 60



(b) Sample 90



(c) Sample 120



(d) Sample 200

Fig. 8: Cofusion Matrix of block 6

The confusion matrix plots the sample's predicted results on the horizontal axis and the actual labels of the samples on the vertical axis. Fig 8(a), (b), (c), (d) shows the confusion matrix results for block 6. It is difficult to diagnose in sample 120 compared to other samples. In particular, it can be seen that it is difficult to diagnose in other categories except category 4.

5 Conclusion

In this paper, we propose a siamese network structure for classifying bearing defects through Few shot learning on the CWRU data set and see the changes in accuracy and parameters according to the number of blocks in WDCNN.

In future studies, in addition to the CWRU dataset, it can be considered in future research as a dataset with noise added to a dataset in the actual field. Additionally, we plan to conduct research focusing on improving bearing fault diagnosis accuracy while reducing the number of parameters through other models other than WDCNN.

Acknowledgement:
This research was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2021R1F1A1060054), the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2022-2018-0-01417) and the ITC Creative Consilience Program (IITP-2022-2020-0-01821) supervised by the IITP (Institute for Information Communications Technology Planning Evaluation) supervised by the IITP (Institute for Information Communications Technology Planning Evaluation)" Corresponding author: Professor Jongpil Jeong.

References:

[1] S. Lee and J. Jeong, "SSA-SL Transformer for Bearing Fault Diagnosis under Noisy Factory Environments", *Journal of electronics*, Vol.11, Issue. 9, May. 2022, pp. 1-21.

[2] A. Zhang, S. Li, Y. Cui, W. Yang, R. Dong and J. Hu, "Limited Data Rolling Bearing Fault Diagnosis with Few-Shot Learning", *IEEE Access*, Vol.7, Aug. 2019, pp. 110895-110904.

[3] S. Han, S. Oh and J. Jeong, "Bearing Fault Diagnosis Based on Multiscale Convolutional

neural network Using Data Augmentation", *Journal of Sensors*, Feb. 2021, pp. 1-14.

[4] K. Yip and G. Sussman, "Sparse Representation
s for fast, One-Shot Learning", *National Conference on Artificial Intelligence*, July 1997, pp. 1-29.

[5] G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese neural networks for one-shot image recognition", *ICML Deep Learning Workshop*, pp. 1–30, July 2015.

[6] Y. Wang, Q. Yao, J. Kwok and L. M. Ni, "Generalizing from a few examples: A survey on few-shot learning", *Arxiv*, Apr. 2019, pp 1-33.

[7] Z. Cui, X. Kong and P. Hao, "Few-shot Learning for Rolling Bearing Fault Diagnosis Based on Residual Convolution Neural Network", *2021 4th International Conference on Artificial intelligence and Big Data*, May. 2021, pp. 320-324.

[8] Y. Yang, H. Wang, Z. L and Z. Y, "Few Shot Learning for Rolling Bearing Fault Diagnosis Via Siamese Two-dimensional Convolutional Neural Network", *2020 11th International conference on Prognostics and System Health Management*, Oct. 2020 pp. 373-378.

[9] D. Wu, F. Zhu, L. Shao, "One shot learning gesture recognition from RGBD images", 2012 IEEE *Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, Jun. 2012. pp.7-12.

[10] S. Oh, S. Han and J. Jeong, "Multi-Scale Convolutional Recurrent Neural Network for Bearing Fault Detection in Noisy Manufacturing Environments", *Journal of Applied Sciences*, Vol.11, Issue.9, May. 2021, pp. 1-16.

[11] M. Alrifaey, W. Lim and C. Ang, "A Novel Deep Learning Framework Based RNN-SAE for Fault Detection of Electrical Gas Generator", *IEEE Access*, Vol.9, Jan. 2021, pp. 21433-21442.

[12] Q. yu, Z. Peng, X. cheng and F. dong, "RNN – based Method for Fault Diagnosis of Grinding System", *2017 IEEE 7th Annual International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER)*, Aug. 2018, pp. 673-678.

[13] X. Lin, B. Li, X. Yang and J. Wang "Fault Diagnosis of Aero-engine Bearing Using a Stacked Auto-Encoder Network", *2018 IEEE 4th Information Technology and Mechatronics Engineering Conference (ITOEC)*, Dec. 2018, pp. 545-548.

[14] C. Liu, B. Chen, H. Zhang and X. Wang, "Fault Diagnosis Application of Short Wave Transmitter Based on Stacked Auto-Encoder", *IEEE 4th International Conference on Computer and Communications(ICCC)*, Dec. 2018, pp.119-123.

[15] D. Neupane and J. Seok, "Bearing Fault Detection and Diagnosis Using Case Western Reserve University Daataset With Deep Learning Approaches: A review", *IEEE Access*, Vol.8, Apr. 2020, pp. 93155-93178.

[16] Q. Guo, Y. Li, Y. Song, D. Wang and W. Chen, "Intelligent Fault Diagnosis Method Based on Full 1-D Convolutional Generative Adversarial Network", *IEEE Transactions on Industrial Informatics*, Vol.16, Issue.3, Aug. 2019, pp.2044-2053.

[17] F. Zhou, S. Yang, H. Fujita, D. Chen and C. Wen, "Deep learning fault diagnosis method based on global optimization GAN for unbalanced data", *Knowledge-Based Systems*, Vol.187, Jan. 2020, pp.1-19.

[18] Case Western Reserve University(CWRU) (https://engineering.case.edu/bearingdatacenter).

[19] A. Parnami, M. Lee "Learning from Few Examples: A Summary of Approaches to Few-Shot Learning", *Arxiv*, Mar. 2022, pp. 1-32.

[20] C. Chen, Z. Liu, G. Yang, C. Wu and Q. Ye "An Improved Fault Diagnosis Using 1D-Convolutional Neural Network Model", *Journal of electronics*, Vol.10, Issue.1, May. 2022, pp. 1-21.