IMAGE RECOGNITION FOR BEARING FAULT DETECTION BASED ON CONVOLUTIONAL NEURAL NETWORK

Muhammad Harith Kamal* and Khairul Amirin Emar Azami

School of Mechanical Engineering, Faculty of Engineering,
Universiti Teknologi Malaysia,
81310 UTM Johor Bahru, Johor, Malaysia

*Corresponding email: harith93@graduate.utm.my

Article history
Received
17th July 2023
Revised
16th July 2024
Accepted
19th September 2024
Published
29th December 2024

ABSTRACT

Bearing is an essential component for rotary machinery. The bearing serves as a fixture for position and provides stability for rotation. Bearing failure has detrimental effects on production schedules and operations. Consequently, detecting and diagnosing bearing issues in advance ensures the safety and reliability of rotating equipment systems, which will definitely save production costs and time. Therefore, this paper proposes the use of image recognition based on a convolutional neural network (CNN) for machine fault detection. Recent years have seen the development of deep learning-trained artificial intelligence, which aims to reduce human-induced errors and expenses. Initially, we acquire the vibration signals of the bearing from a test rig under four different conditions. We consider four bearing conditions: healthy bearings, inner race defects, outer race defects, and ball bearing defects. Each of the four conditions is recorded in the vibration time-series data, then converted into spectrogram images before feeding it to the CNN model for training. The performance of the CNN model is based on the comparison of two different models, which are Model A and Model B. Model B is developed based on the performance of Model A, where hyperparameter tuning is implemented to improve the performance. The result shows that the proposed model is capable of detecting and classifying the bearing faults up to 99.9% accuracy.

Keywords: *Bearings, CNN, Image recognition, Fault detection*

1.0 INTRODUCTION

Artificial intelligence (AI) or machine learning methods had been widely introduced and implemented in various fields of study, especially in machine maintenance methods. Machines commonly encounter bearing problems, with bearings being the primary cause of fault in any rotating machinery. In rotating equipment, the most common source of failure is rolling bearings. They are to blame for 45–50% of all machine failures. These are caused by the cracks and pits on the bearing parts, such as the inner race, outer race, or ball [1]. Vibration analysis is an effective method for bearing fault diagnosis because of its sensitivity to failure and ability to apply nondestructive testing [2]. This particular case study focuses on image fault detection using artificial intelligence for machine maintenance. The goal of this research is to develop bearing fault detection using artificial intelligence (AI) and image recognition for machinery applications that rely on bearing systems.

In the context of image recognition, image recognition refers to the computer or program's ability to recognise objects, places, people, and writing in the form of imagery or pictures. Image recognition is based on deep learning in theory and a subclass of machine learning (ML). ML, a type of artificial intelligence (AI), enables software programs to improve their ability to process data and produce required outputs with high accuracy, even without explicit instructions [3]. The type of neural network that will be used is CNN due to the higher performance compared to other common methods [4]. CNN is widely used in image recognition and classification. CNN is also deemed to have relatively good performance in predicting defective engines [5] and structural damage detection in bridges [6]. Within ML, CNN is a powerful tool for pattern recognition, as in image, video, voice recognition, and face detection [7]. Moreover, a study on the CNN application for pathogen classification, which requires specialist-level observation, achieved up to 94% accuracy [8]. Theoretically, CNN is a neural network with multiple units integrated between the input and output layers that is used to analyse data and it relies heavily on convolutional operations. Furthermore, convolution kernels serve the same purpose as filters. The convolutional layer, pooling layer that, and fully connected layer comprise its composition [9]. CNN Image Classification reads the input image, assesses it, and categorizes it into several groups.

Next, Rectified Linear Unit (ReLu) is widely used as an activation function, especially in CNN models. The primary purpose of using Rectified Linear Unit (ReLU) is to introduce non-linearity into our CNN model [10], which boasts a fast convergence speed [11], and enables high-speed training due to its low computational requirement [12]. Next, when discussing hyperparameters, an epoch refers to the entire process of the training data being processed by the machine learning algorithm. It's a machine learning word for the number of passes the machine learning algorithm has made across the complete training dataset [13]. Other than that, Dropout is also used in CNN models. Dropout is a type of training that disregards randomly selected neurones. The term "dropped-out" refers to the random loss of neurones from the CNN model during training [14]. This prevents the model from overfitting.

2.0 METHODOLOGY

The first step is dataset collection, and for this project, the dataset will be obtained from a test rig that has all the required equipment. Figure 2.1 below shows the test rig for acquiring the dataset. Each reading is taken from each of the four different types for analysis, and once the dataset is adequate for the training and testing, the next step would be classification and labelling.

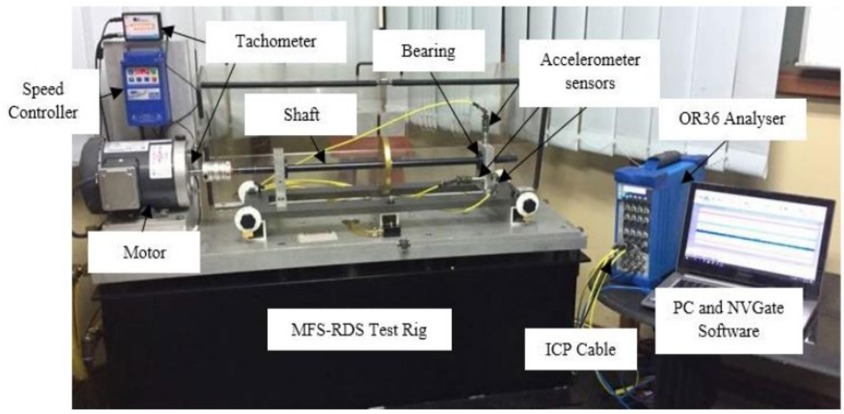

Figure 2.1: The test rig for the dataset preparation

2.1 Dataset Preparation

The next step for this project is to convert raw data signals into spectrogram images using MATLAB for signal processing based on each type. Figure 2.2 displays the raw data collected from the test rig operating at 1772 rpm. The preparation of the dataset essentially involves a series of steps or procedures designed to simplify the machine learning process by arranging the data in a way that is understandable for the CNN model.

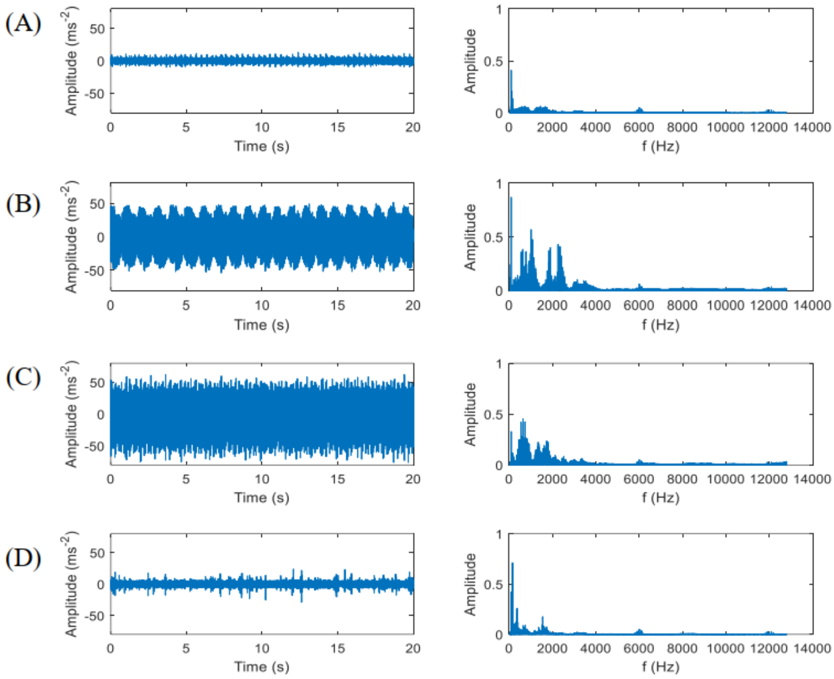

Figure 2.2: Raw vibration data signals from the test rig (A) healthy, (B) inner race fault, (C) outer race fault and (D) ball defect

Next, the dataset is formed from the vibration signals collected from the four groups of bearings: healthy bearing, inner race defect, outer race defect, and ball bearing defect. Table 2.1 shows the number of images for each variable. Figure 2.3 shows the sample of spectrogram images from all four categories.

Table 2.1: Total number of images for variables of the dataset

No	Variables	Number of images
1	Healthy bearing	1000
2	Outer race defect	1000
3	Inner race defect	1000
4	Ball bearing defect	1000

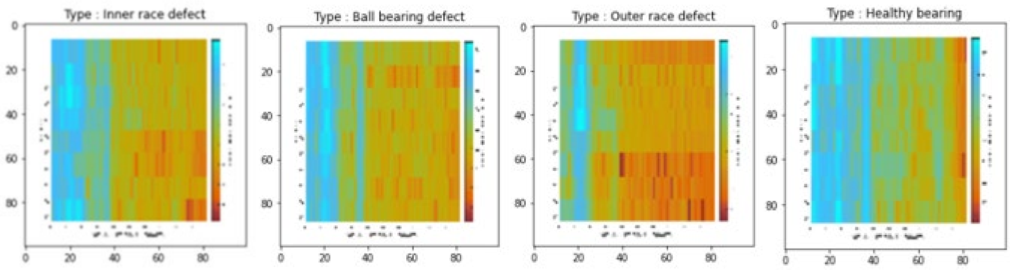

Figure 2.3: Spectrogram image that has been converted from the vibration signals

2.2 The Development of the Proposed CNN Model

Once the conversion of the raw vibration signals into spectrogram images is done, the converted image will be put into different folders based on their categories. This is to ensure that the dataset is organised and easily accessible for the model to interpret as inputs. Then, the proposed model will start by specifying the number of nodes based on the Conv2D layer. The number of nodes for both layers is 40. The input shape that will be used is 100x100x1 for the input, and the size of the kernel used is 3x3. As for the activation function for the hidden layer that is chosen, it is the rectified linear unit "ReLu.".

Next, the pooling type for this proposed model is max pooling, with a size of 2x2 for both layers. This size is commonly used, particularly in CNN, where the pooling layer reduces the number of values in each feature map by halving the dimension. Other than that, the number of nodes for a fully connected layer is set to 4 nodes. This is because the projected output has four different conditions for the bearing type. Next, when it comes to the activation function for the output layer, the activation function used is a sigmoid function [15]. Then, the final section of the model will compile with the usage of "categorical crossentropy" for the selection of the loss function, where the optimiser named "Adam" is also used for finalising the model once it has compiled all the features [16]. Table 2.2 below shows the summary of the model that will be developed.

Table 2.2: The summarized characteristics of the proposed CNN model

Layer number	Type	Nodes	Kernel Size	Activation functions
1	Convolution	40	3×3	Rectified Linear Unit
2	Max Pooling	40	3×3	Rectified Linear Unit
3	Convolution	40	3×3	Rectified Linear Unit
4	Max Pooling	40	3×3	Rectified Linear Unit
5	Fully Connected layer – Hidden layer (Dense)	4	-	Rectified Linear Unit
6	Fully Connected layer (Output)	4	-	Sigmoid Function

3.0 RESULT AND DISCUSSION

The analysis of the results of the proposed model is based on its accuracy and the loss value data from testing, which shows whether the model is able to classify each category accordingly. The result will also be plotted for better understanding, and it is where the model is underfitting, overfitting, or performing at optimum level. The proposed model will have two different models, which are Model A and Model B, where Model B is developed to improve based on the results from Model A. This is done to compare which model yields the best results based on its accuracy and the number of loss values. The loss function is used to evaluate the discrepancy between the output of a neural network's predictions and its actual value.

3.1 The Result of Model A

Model A uses the same characteristics as in Table 2.2. The number of iterations for Model A is 20; the result is shown in Figure 3.1. Based on Figure 3.1, the training accuracy for Model A after 20 epochs is 99.5%, and for validation accuracy, the result shows 91.75% accuracy. However, the validation accuracy surpasses 90% accuracy in the final iterations.

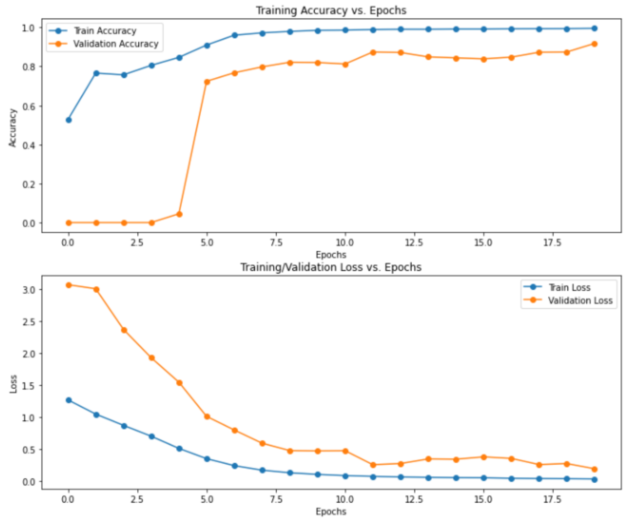

Figure 3.1: The plotted graph of the result for training and testing of Model A

Both training and validation losses show significant declines that correlate with each other. This indicates that the model's classification errors decrease with each iteration during testing. This indicates a positive outcome, as the model is able to generalize the output rather than simply memorizing the training dataset. However, the result from Model A could be improved with some tuning where the result could converge faster and the testing accuracy could be improved to match with the training accuracy and could possibly go up to 98% accuracy.

3.2 The Result of Model B

Moreover, Model B is developed to improvise the result from Model A and compare the performance between both models. The changes that can be seen from Model B apart from Model A are the extra layers in the model itself, the number of iterations, and data augmentation. The number of iterations is increased to 50 epochs. This is to ensure that the neural network learns the structure of the data appropriately, especially if the model uses a very large dataset or has a high record of failed validation. As for data augmentation, it is commonly used to increase the diversity of the training data by applying random transitions such as rotating the image to a certain degree or randomly flipping the images [5]. Figure 3.2 shows the sample of data augmentation of the dataset before proceeding to the CNN model for classification.

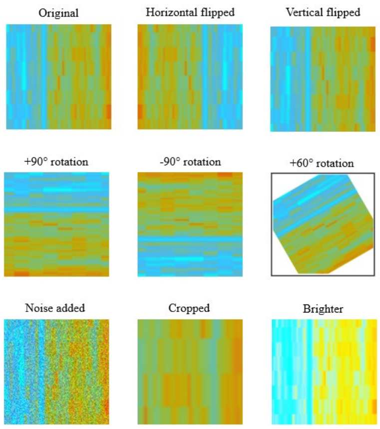

Figure 3.2: Sample of augmentation of the dataset

Next, the extra layer that is added is called the dropout layer. The dropout layer functions as a mask, suppressing the connections of some neurons to the subsequent layer, while preserving the functionality of all other neurons. It is commonly used in CNN models to prevent overfitting. The number of nodes for this model is maintained at 40 nodes for each convolution layer. The summary of the characteristics for Model B can be seen in Table 3.1 below.

Table 3.1: The summarized characteristics of the model B

Layer number	Type	Nodes	Kernel Size	Activation functions	Rate
1	Convolution	40	3×3	Rectified Linear Unit	-
2	Max Pooling	40	3×3	Rectified Linear Unit	-
3	Convolution	40	3×3	Rectified Linear Unit	-
4	Max Pooling	40	3×3	Rectified Linear Unit	-
5	Dropout	-	-	-	0.15
6	Fully Connected layer – Hidden layer (Dense)	64	-	Rectified Linear Unit	-
7	Fully Connected layer (Output)	4	-	Sigmoid Function	-

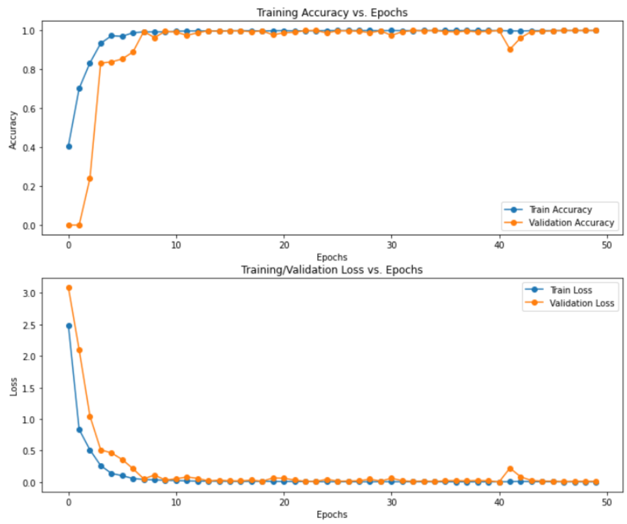

Figure 3.3: The plotted graph of the result for training and testing of Model B

From Figure 3.3, it can be seen that the accuracy of Model B improved compared to Model A, where it reaches up to 100% accuracy for both training and testing accuracy after 50 iterations. Other than that, the dropout rate is set at 15%, which shows a positive result where the training accuracy maintains above 95%. It is essential to set the dropout rate at the optimum value by trial and error, as the higher the dropout rate, the higher the chance that the model could be underfitting. Moreover, both the training and validation loss results have improved, particularly for the validation loss, where the majority of the loss value has been below 0.05 for all 50 iterations. For Model A, the validation loss is significantly higher than that of Model B. Figure 3.4 below shows the comparison between the results of Model A and Model B.

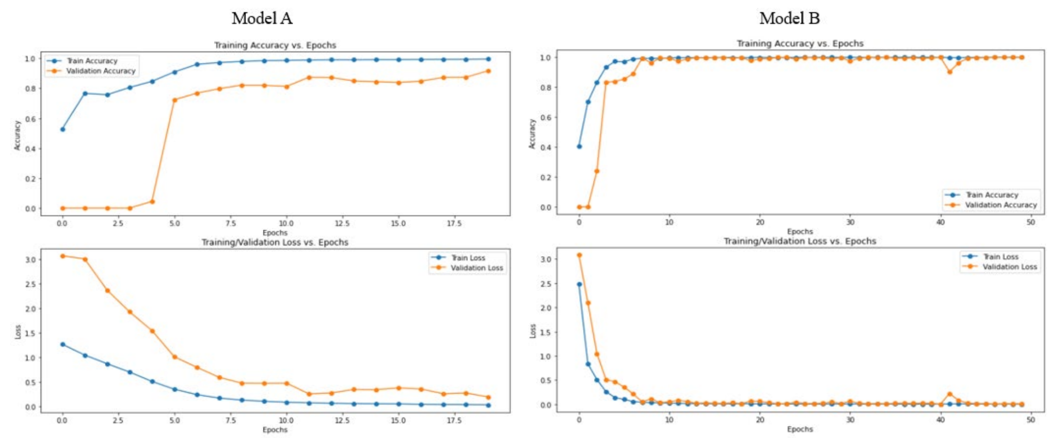

Figure 3.4: The comparison of the performance between Model A and Model B

3.3 Comparison of the Performance of Model A and Model B

In summary, the CNN model performs better when the hyperparameters are tuned and data is augmented in Model B based on the results of Model A. The addition of the dropout layer at the rate of 15% is at the optimum level as the training does not show any sign of decrement, which can cause the model to be underfitting and continue to maintain above 98%. This demonstrates that the dropout layer effectively prevents some neurones from converging towards the same goal, which could otherwise lead to overfitting of the model. Next, increasing the number of iterations to 50 epochs improves the model's performance, ensuring that both plotted graphs converge and maintain accuracy without any fluctuations. The validation accuracy also shows no sign of overfitting during testing, as the value of validation loss continues to drop for both models without any signs of sudden increases or deflections. The number of epochs is subjective; if the number is too few, it may lead to underfitting, while too many epochs can also lead to overfitting.

4.0 CONCLUSION

In conclusion, this project has achieved its objective, which is to develop fault detection artificial intelligence (AI) using image recognition for machinery applications based on bearing systems. After a few tunings and adjustments based on the two models, the proposed CNN model performs well, particularly in image recognition for fault detection, achieving an accuracy of more than 98%. However, the CNN model can be improved by adding more variety for its classification, and increasing the number of datasets will help the model with more options and helps generalise the idea better.

ACKNOWLEDGEMENTS

The author would like to extend their greatest gratitude to UTM Encouragement Re-search Grant Scheme (UTM-ER), Q.J130000.3824.31J20.

REFERENCES

[1] M. Momeny and Ali Mohammad Latif, 2021. A noise robust convolutional neural network for image classification, *Results in Engineering*, 10.

[2] H. Li, Q. Zhang, X. Qin, and S. Yuantao, 2021. Raw vibration signal pattern recognition with automatic hyper-parameter-optimized convolutional neural network for bearing fault diagnosis, *Proc Inst Mech Eng C J Mech Eng Sci*, vol. 234, no. 1, pp. 343–360, doi: 10.1177/0954406219875756.

[3] A. Geron, 2019. Hands-on Machine Learning with Scikit-Learn, Keras & Tensorflow, *Hands-on Machine Learning with Scikit-Learn, Keras & Tensorflow, O'reilly.*: 3–15.

[4] W. Zhang, G. Peng, and C. Li, 2017. Bearings Fault Diagnosis Based on Convolutional Neural Networks with 2-D Representation of Vibration Signals as Input, *MATEC Web of Conferences, vol. 95*, p. 13001. doi: 10.1051/matecconf/20179513001.

[5] N. Günnemann, J. Pfeffer, L. Torgo, B. Krawczyk, P. Branco, and N. Moniz, 2017. Predicting Defective Engines using Convolutional Neural Networks on Temporal Vibration Signals.

[6] Y. Zhang, Y. Miyamori, S. Mikami, and T. Saito, 2019. Vibration-based structural state identification by a 1-dimensional convolutional neural network, *Computer-Aided Civil and Infrastructure Engineering*, vol. 34, no. 9, pp. 822–839, doi: 10.1111/mice.12447.

[7] . Albawi, T. A. Mohammed, and S. Al-Zawi, 2017. Understanding of a convolutional neural network, *2017 International Conference on Engineering and Technology (ICET), IEEE*, pp. 1–6. doi: 10.1109/ICEngTechnol.2017.8308186.

[8] B. B. Traore, B. Kamsu-Foguem, and F. Tangara, 2018. Deep convolution neural network for image recognition, *Ecol Inform*, vol. 48, pp. 257–268, doi: 10.1016/j.ecoinf.2018.10.002.

[9] Shao Haidong and Jiang Hongkai, 2018. A novel method for intelligent fault diagnosis of rolling bearings using ensemble deep auto-encoders, *Mechanical Systems and Signal Processing*, 102, pp. 278-297.

[10] Wenhua and Zhijian Wang, 2021. Data-driven fault diagnosis method based on the conversion of erosion operation signals into images and convolutional neural network, *Process Safety and Environmental Protection*, 149, pp. 591-601.

[11] G. Lin and W. Shen, 2018. Research on convolutional neural network based on improved Relu piecewise activation function, *Procedia Computer Science, Elsevier B.V.*, pp. 977–984. doi: 10.1016/j.procs.2018.04.239.

[12] D. T. Hoang and H. J. Kang, 2019. Rolling element bearing fault diagnosis using convolutional neural network and vibration image, *Cogn Syst Res*, vol. 53, pp. 42–50, doi: 10.1016/j.cogsys.2018.03.002.

[13] Jin Woo Oh and Jongpil Jeong, 2020. Data augmentation for bearing fault detection with a light weight CNN, *Procedia Computer Science*, 175, pp. 72-79.

[14] Youngjun Yoo and Seongcheol Jeong, 2022. Vibration analysis process based on spectrogram using gradient class activation map with selection process of CNN model and feature layer, *Displays*, 73.

[15] H. Pokharna, 2016. The best explanation of Convolutional Neural Networks on the Internet, *Medium.com*, pp. 1–9.

[16] Nikhil A. Sonkul, 2021. Single and Multi-Label Fault Classification in rotors from unprocessed multi-sensor data through deep and parallel CNN architectures, *Expert Systems with Applications*, 185.