DOI 10.2478/ama-2022-0017

BEARING FAULT DETECTION AND DIAGNOSIS BASED ON DENSELY CONNECTED CONVOLUTIONAL NETWORKS

Julius NIYONGABO*, Yingjie ZHANG, Jérémie NDIKUMAGENGE*****

*Doctoral School, University of Burundi, UNESCO Road No 2, Bujumbura 1550, Burundi
**College of Computer Science and Electronic Engineering, Hunan University, Lushan Road (S), Changsha 410082, China
***Faculty of Engineering Science, Department of Information and Communication Technology, University of Burundi, UNESCO Road No 2, Bujumbura 1550, Burundi

niyojule@gmail.com, zhangyj@hnu.edu.cn, jeremie.ndikumagenge@ub.edu.bi

received 18 December 2021, revised 3 February 2022, accepted 12 February 2022

Abstract: Rotating machines are widely used in today's world. As these machines perform the biggest tasks in industries, faults are naturally observed on their components. For most rotating machines such as wind turbine, bearing is one of critical components. To reduce failure rate and increase working life of rotating machinery it is important to detect and diagnose early faults in this most vulnerable part. In the recent past, technologies based on computational intelligence, including machine learning (ML) and deep learning (DL), have been efficiently used for detection and diagnosis of bearing faults. However, DL algorithms are being increasingly favoured day by day because of their advantages of automatically extracting features from training data. Despite this, in DL, adding neural layers reduces the training accuracy and the vanishing gradient problem arises. DL algorithms based on convolutional neural networks (CNN) such as DenseNet have proved to be quite efficient in solving this kind of problem. In this paper, a transfer learning consisting of fine-tuning DenseNet-121 top layers is proposed to make this classifier more robust and efficient. Then, a new intelligent model inspired by DenseNet-121 is designed and used for detecting and diagnosing bearing faults. Continuous wavelet transform is applied to enhance the dataset. Experimental results obtained from analyses employing the Case Western Reserve University (CWRU) bearing dataset show that the proposed model has higher diagnostic performance, with 98% average accuracy and less complexity.

Key words: bearing, deep learning, machine learning, transfer learning, fault detection and diagnosis, CWRU dataset

1. INTRODUCTION

With the rapid growing of industrialisation, electrical machines are always present in industries and electrified transportation systems. Often, these machines operate under harsh environments and complex conditions, such as high ambient temperature, high moisture and overload, which can eventually result in motor malfunctions that lead to high maintenance costs, severe financial losses and safety hazards [1]. Many research projects have revealed that rolling element bearings, also known as bearings, are the most vulnerable parts of electrical machines, accounting for high failure rate and downtime. These studies have also shown that bearing fault is the most common fault type and is responsible for 30–40% of all electrical machine failures [1,2]. Fig. 1. illustrates the structure of a rolling element bearing with four types of common scenarios of misalignment that are likely to cause bearing failures.

Since bearing is the most vulnerable component in a motor drive system, fault detection and diagnosis of bearings have become an essential part of development and engineering research [3]. Therefore, signal processing based–methods, machine learning (ML)–based techniques and deep learning (DL)–based approaches have been proposed and implemented. Signal processing–based methods implement fault diagnosis by detecting characteristic frequencies which are related to the faults [2]. For example, Li et al. [4] developed a variational mode decomposition–based bearing fault diagnosis method, which can effectively identify fault frequencies. However, the signal processing–based methods need to rely on professional knowledge, and it is arduous for these methods to realise accurate fault diagnosis under an actual strong noise environment [5]. On the other hand, ML-based approaches and DL, known also as intelligent methods, can perform the fault diagnosis task without the fault-related characteristics' frequencies and prior physical knowledge [2].

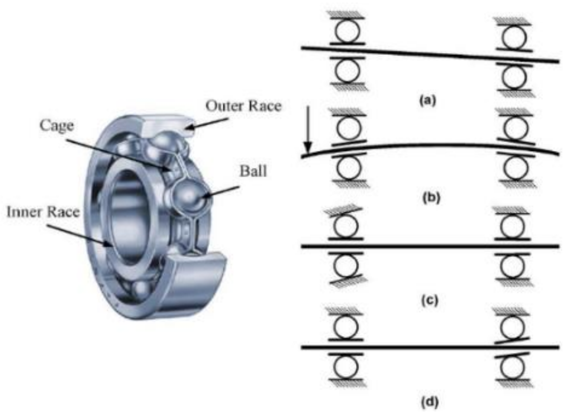

Fig. 1. Structure of a rolling element bearing with four types of common scenarios of misalignment that are likely to cause bearing failures: (a) Misalignment (out-of-line), (b) shaft deflection, (c) crooked or titled out race and (d) crooked or titled inner race [1]

Recently, various methods based on ML algorithms, including artificial neural networks (ANN), principal component analysis (PCA) and support vector machines (SVM), have been used successfully to make intelligent decisions regarding the presence of bearing faults [6-8]. Most of the literature applying these ML algorithms report satisfactory results with a classification accuracy of >90% [1]. However, one of the disadvantages of classical ML techniques is that they usually demand sophisticated manual feature engineering, which unavoidably requires expert domain knowledge and numerous human efforts.

To achieve better performance, DL methods with automated feature extraction capabilities have recently received much interest for bearing faults diagnosis [9,10]. DL is a subfield of ML that is inspired by ANN, which in turn are inspired by biological neural networks [11-15]. In the recent decade, deep Convolutional Neural Networks (CNN) have been classified as the most utilised models of DL and have been successfully applied to identify bearing faults. Moreover, many variations of CNN are employed in bearing fault diagnosis.

In this paper, after applying transfer learning techniques on DenseNet-121 architecture, which is one of the latest discoveries in neural network architectures and solves challenges caused by the depth of CNN, we propose a new approach to detect and diagnose bearing faults. We show the effectiveness of our model by conducting its evaluation and compare the experimental results to other previous CNN-based models.

The rest of this article is structured as follows: In section II, we briefly explain CNN architecture. Section III contains an overview of DenseNet algorithm and details of the proposed approach. In section IV, the most popular bearing dataset used in this work is presented. In section V, results are discussed. Finally, the conclusion is presented in section VI.

2. CNN ARCHITECTURE

CNN are feed-forward neural networks in which information flow takes place in one direction only, from their inputs to their outputs. They are composed by an input layer, convolutional layers, pooling layers, fully connected layers and an output layer [16]. The typical architecture of a CNN-based bearing fault classifier is illustrated in Fig. 2. The convolutional layer processes data input and passes the result to the next layer. The pooling layer has a function of compressing data in order to reduce its size and parameters. It is also charged to control overfitting. Often, max-pooling is the most-intensively used convolution operation in CNN. The fully connected layer is on the last position and looks like the perceptron. Its inputs come from the final convolutional layer. At the end of CNN, the SoftMax operator is used for classification problems. Since 2016, many papers applying CNN bearing fault diagnosis [17-25] have proliferated in the research domain.

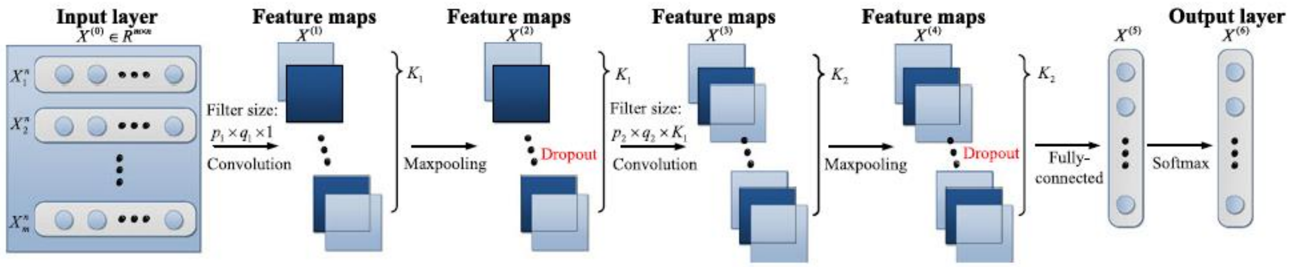

Fig. 2. Architecture of CNN-based fault diagnosis model (1). CNN, convolutional neural networks

The most common CNN architectures developed in the last decade are Alexnet (2012), Inception (2014), VGGNet (2014), ResNet (2015) and DenseNet (2017). Since AlexNet, the number of layers has increased and CNN architecture went deeper and deeper. However, deep networks become complex and difficult to be trained because of the vanishing gradient problem or dead neurons; the accuracy starts saturating and then degrades also. DenseNet has been proposed [26] to overcome this problem and easily train deep CNN.

3. OVERVIEW OF DENSENET AND THE PROPOSED MODEL

3.1. Overview of DenseNet

DenseNets or Densely Connected Convolution networks were introduced in 2017 by Gao Huang Liu, Zhuang Liu, Laurens van der Maaten and Kilian Q. Weinberger in their paper "Densely Connected Convolutional Networks" [26].

In comparison with traditional convolutional networks, which have L layers with L connections, one between each layer and its subsequent layer, DenseNet has L(L+1)/2 direct connections, and each layer is directly connected to every other layer.

DenseNet architecture is mainly constituted by Dense Block and the transition layers. The transition layers consist of a Batch-Normalisation layer, and 1 × 1 convolution followed by a 2 × 2 average pooling layer. The first part of DenseNet architecture is constituted by 7 × 7 convolution, and a stride 2 layer followed by a 3 × 3 maxpool, stride 2 layer. After the four dense blocks comes a classification layer. Inside each Dense Block and transition layer, the convolution operations are performed [26].

The main advantages of DenseNet are that it overcomes the vanishing gradient problem and does not require many parameters to train the model. Moreover, variation in the input of layers as a result of concatenated feature maps prevents the model from succumbing to the overfitting problem. The most popular DenseNet architectures are DenseNet-121, DenseNet-169, DenseNet-201 and DenseNet-264. Tab. 1 shows that each DenseNet architecture has four dense blocks with a varying number of layers, and Transition Layers are added between these blocks. Thus, DenseNet-121 consists of [6, 12, 24, 16] layers in the four dense blocks, DenseNet-169 has [6, 12, 32, 32] layers and DenseNet-201 consists of [6, 12, 48, 32] layers, whereas DenseNet-161 has [6, 12, 36, 24] layers.

DOI 10.2478/ama-2022-0017

Tab. 1. DenseNet Architectures for ImageNet [26]

Layers	Output Size	DenseNet-121	DenseNet-169	DenseNet-201	DenseNet-264
Convolution	112×112	7×7 conv, stride 2			
Pooling	56×56	3×3 max pool, stride 2			
Dense Block (1)	56×56	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ vonv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$
Transition Layer (1)	56×56	1×1 conv			
	28×28	2×2 average pool, stride 2			
Dense Block (2)	28×28	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$
Transition Layer (2)	28×28	1×1 conv			
	14×14	2×2 average pool, stride 2			
Dense Block (3)	14×14	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 24$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 48$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 36$
Transition Layer (3)	14×14	1×1 conv			
	7×7	2×2 average pool, stride 2			
Dense Block (4)	7×7	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 16$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 1 \times 1 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 24$
Classification Layer	1×1	7×7 global average pool			
		1000 D fully − connected, softmax			

3.2. The Proposed Model

In this article, by using transfer learning techniques, we fine-tuned DenseNet-121 [26]. To be precise, the following strategies are used:

– DenseNet-121's original architecture is implemented.
– Weights are initialised randomly.
– The last fully connected layer of DenseNet-121 is replaced with another new layer adapted to our classification problem.
– A flatten layer is added before the last fully connected layer.

Before training and evaluation of the proposed model, a transformation of the time domain data to wavelet domain is realised, and thereafter, we train and evaluate the new architecture on wavelet domain image for fault diagnosis. As the number of classes for Case Western Reserve University (CWRU) bearing data is 10, in the proposed approach we use a 10 Dense fully connected layer to detect and diagnose different faults of bearing. SoftMax is utilised for a classification layer. A comparison with other CNN-based models is also done. The architecture of the proposed model is shown in Tab. 2 and Fig. 3.

Tab. 2. DenseNet Architecture of the proposed model

Layers	Output Size	Proposed Model
Convolution	112×112	7×7 conv, stride 2
Pooling	56×56	3×3 max pool, stride 2
Dense Block (1)	56×56	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$
Transition Layer (1)	56×56	1×1 conv
	28×28	2×2 average pool, stride 2
Dense Block (2)	28×28	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$
Transition Layer (2)	28×28	1×1 conv
	14×14	2×2 average pool, stride 2
Dense Block (3)	14×14	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 24$
Transition Layer (3)	14×14	1×1 conv
	7×7	2×2 average pool, stride 2
Dense Block (4)	7×7	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 16$
Classification Layer	1×1	7×7 global average pool
	Flattening	
		10 Dense fully − connected, softmax

4. DATASET AND IMPLEMENTATION

4.1. Dataset

Data is the fundamental unit and the foundation for all ML or DL architectures. Deep networks and DL algorithms are influential ML structures that work most excellently if trained on vast amounts of data. With a small training dataset, we get limited sample variations, and the network efficiency decreases. So, the quantity and the frequency of data availability have a vital role in DL applications. Generally, the more the data, the better the accuracy [2].

To evaluate the performance of the model, in this research work, we used the popular bearing dataset from CWRU Bearing Data Center. The CWRU bearing dataset has become one of the

DOI 10.2478/ama-2022-0017 acta mechanica et automatica, vol.16 no.2 (2022)

most popular datasets for machine fault diagnosis since it was published. The experimental setup for collecting the CWRU bearing dataset, consisting of an electric motor on left, a torque transducer/encoder in the middle and a dynamometer on the right, is illustrated in Fig. 4.

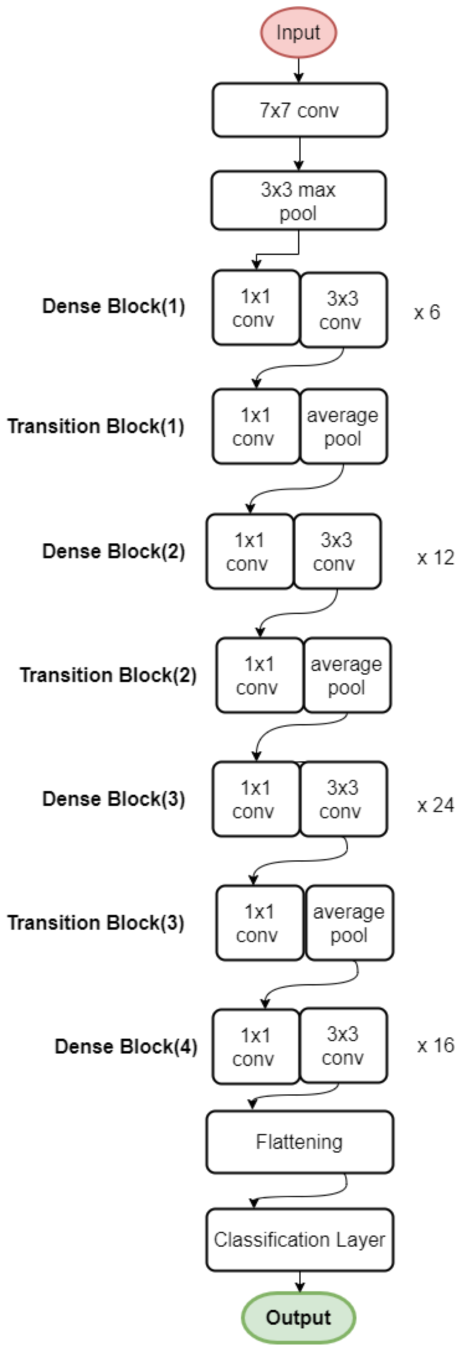

Fig. 3. Structure of the proposed model

The single point motor bearing faults simulated by the electro-discharge machining were tested in this platform, including inner-race fault (IF), outer-race fault (OF) and ball fault (BF). Each fault has three types: 7 mils, 14 mils, 21 mils (1 mil = 0.001 inches). The sampling frequencies of 12 kHz and 48 kHz were used for the collection of data. For the drive-end bearing experiments, data were collected at 12,000 and 48,000 samples/s. Fan-end data were collected at 12,000 samples/s. For the normal-baseline, the data collection rate was 48,000 samples/s [2].

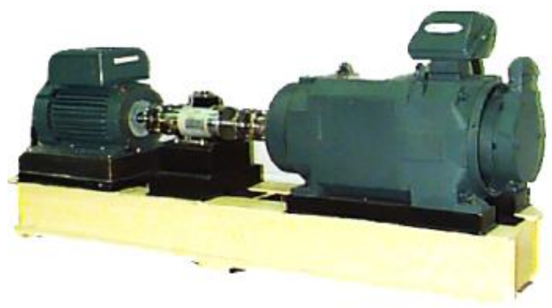

Fig. 4. Experimental setup for collecting the CWRU bearing dataset. CWRU, Case Western Reserve University

In this work, we selected 48 kHz drive-end bearing fault data. Normal data collected with 1 hp load have also been used. We applied continuous wavelet transform on the dataset consisting of 4,600 data samples and 10 classes are considered: Ball defect (0.007 inch, load: 1 hp), Ball defect (0.014 inch, load: 1 hp), Ball defect (0.021 inch, load: 1 hp), IF (0.007 inch, load: 1 hp), IF (0.014 inch, load: 1 hp), IF (0.021 inch, load: 1 hp), Normal (load: 1 hp), OF (0.007 inch, load: 1 hp, data collected from 6 O'clock position), OF (0.014 inch, load: 1 hp, 6 O'clock) and OF (0.021 inch, load: 1 hp, 6 O'clock) [13].

4.2. Implementation

In this research paper, we implemented our model with Tensorflow (2.3.0) environments using Jupyter Notebooks and Python on a computer with the following properties:
– Processor: Intel® Core (TM) i5-8250U CPU @ 1.60GHz 1.80 GHz
– Installed RAM: 12.0 Go (11.9 Go Usable)

The number of parameters of the model is as follows: total: 7.047.370, trainable: 6.965.898 and non-trainable: 81.472. The following hyperparameters are used: number of filters: 128, kernel size:1, strides: 2, zero-padding: same, batch size: 64 and learning rate: 0.007.

The activation function used in the model is ReLU. The size of the data is (4600, 32, 32). Eighty percent of the total data is taken as training data and 20% as test set [13,24]. The number of epochs for all models in this paper is 50, the optimiser is SGD and momentum is initialised at 0.5.

5. RESULTS AND DISCUSSION

In this part, we are going to present and discuss the experimental results. After 10 iterations of running the proposed model under the above-mentioned conditions, an average accuracy of 98.57% is obtained. The curves of the training and validation accuracy are shown in Fig. 5. In Fig. 6 the train and validation loss curves are illustrated.

For the reliability of the experiment, confusion matrix of the considered model is created and can be seen in Fig. 7. The comparison of fault diagnosis accuracy with three previous DL Algorithms based on CNN is presented in Tab. 3, from which it is clear that the accuracy of the proposed model is higher when compared with the AlexNet, VGG-16 and ResNet-50 models. Moreover, from Fig. 8, we can confirm that the proposed approach achieves high accuracy.

DOI 10.2478/ama-2022-0017

Tab. 3. Fault diagnosis average accuracy results of AlexNet, VGG-16, ResNet-50 and the proposed model

Method	Iter1	Iter2	Iter3	Iter4	Iter5	Iter6	Iter7	Iter8	Iter9	Iter10	Average
AlexNet	0.9110	0.9430	0.9620	0.9040	0.9280	0.9690	0.9710	0.9280	0.9690	0.9440	0.9349
VGG-16	0.9010	0.9390	0.9560	0.9640	0.9640	0.9650	0.9680	0.9680	0.9680	0.9660	0.9559
ResNet-50	0.9340	0.9440	0.9030	0.9630	0.9760	0.9700	0.9780	0.9840	0.9700	0.9760	0.9603
Proposed model	**0.9750**	**0.9620**	**0.9730**	**0.9670**	**0.9780**	**0.9810**	**0.9790**	**0.9850**	**0.9760**	**0.9810**	**0.9857**

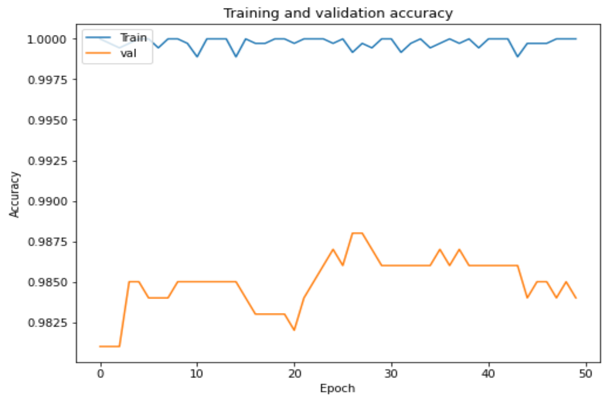

Fig. 5. Curves of training and validation accuracy during training

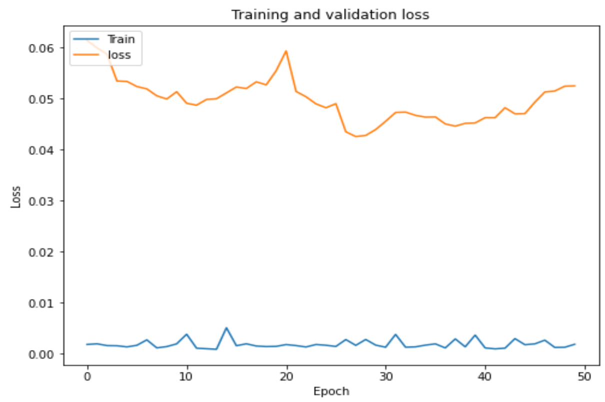

Fig. 6. Curves of training and validation loss during training

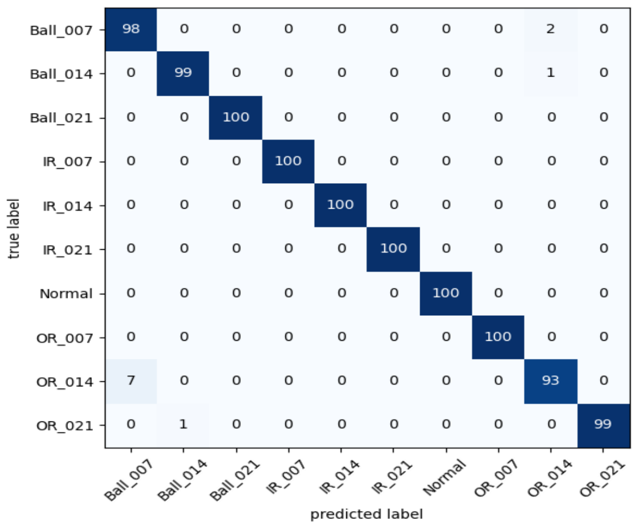

Fig. 7. Confusion matrix of DenseNet model

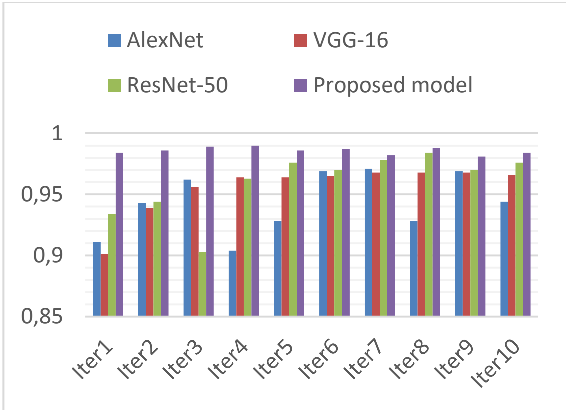

Fig. 8. The average classification accuracies

6. CONCLUSION

The early detection, diagnosis and visualisation of bearing faults can significantly reduce downtime and maintenance costs in industry. In this research, using DL technologies and transfer learning, we presented an intelligent fault diagnosis method based on deep CNN applied to rolling element bearings. We applied the model to the CWRU bearing dataset, and the simulation results reveal that the fine-tuned DenseNet-121 model performs an efficient classification for bearing faults. Moreover, it is observed that, in comparison with three previous DL models of the same family, this method has a higher classification accuracy and identifies different faults of bearing correctly.

Future researches will consist of improving the structure of the model and applying the new model on other bearing datasets.

REFERENCES

1. Zhang S, Zhang S, Wang B, Habetler TG. Deep Learning Algorithms for Bearing Fault Diagnosticsx - A Comprehensive Review. IEEE Access. 2020;8:29857–81.
2. Zhang Z, Li H, Chen L, Han P. Shrinkage Networks. 2021;2021(DI).
3. Neupane D, Seok J. Bearing fault detection and diagnosis using case western reserve university dataset with deep learning approaches: A review. IEEE Access. 2020;8:93155–78.
4. Li G, Tang G, Luo G, Wang H. Underdetermined blind separation of bearing faults in hyperplane space with variational mode decomposition. Mech Syst Signal Process. 2019;120:83–97. https://doi.org/10.1016/j.ymssp.2018.10.016
5. Huang T, Fu S, Feng H, Kuang J. Bearing fault diagnosis based on shallow multi-scale convolutional neural network with attention. Energies. 2019;12(20).
6. Awadallah MA, Morcos MM. Application of AI tools in fault diagnosis of electrical machines and drives - An overview. IEEE Trans Energy Convers. 2003;18(2):245–51.

7. Batista L, Badri B, Sabourin R, Thomas M. A classifier fusion system for bearing fault diagnosis. Expert Syst Appl [Internet]. 2013;40(17):6788–97. http://dx.doi.org/10.1016/j.eswa.2013.06.033
8. Liu R, Yang B, Zio E, Chen X. Artificial intelligence for fault diagnosis of rotating machinery: A review. Mech Syst Signal Process. 2018;108:33–47. https://doi.org/10.1016/j.ymssp.2018.02.016
9. Bansal N, Sharma A, Singh RK. A Review on the Application of Deep Learning in Legal Domain. IFIP Adv Inf Commun Technol. 2019;559:374–81.
10. Zhao R, Yan R, Chen Z, Mao K, Wang P, Gao RX. Deep Learning and Its Applications to Machine Health Monitoring: A Survey. 2016;14(8):1–14. Available from: http://arxiv.org/abs/1612.07640
11. Brownlee J. What is Deep Learning? August 14, 2020 . Available from: https://machinelearningmastery.com/what-is-deep-learning/, October 15 2021.
12. Great Learning Team. Introduction to Resnet or Residual Network. Sep 28. 2020, Available online: https://www.mygreatlearning.com/blog/resnet/, October 15, 2021. 2021;2021.
13. Sahoo B. Fault Diagnosis using Deep Learning on raw time domain data. July 15 2020. Available on line: https://github.com/biswajitsahoo1111/cbm_codes_open/blob/master/notebooks/Deep_learning_based_fault_diagnosis_using_CNN_on_raw_time _domain_data. 2021.
14. Chen Z, Cen J, Xiong J. Rolling Bearing Fault Diagnosis Using Time-Frequency Analysis and Deep Transfer Convolutional Neural Network. 2020;8.
15. Singhal G. Introduct ion to DenseNet with TensorFlow. May 6, 2020, Available online: https://www.pluralsight.com/guides/introduction-to-densenet-with-tensorflow, October 25, 2021.
16. Si L, Xiong X, Wang Z. Tan C. A Deep Convolutional Neural Network Model for Intelligent Discrimination between Coal and Rocks in Coal Mining Face. Math Probl Eng. 2020.
17. Guo X, Chen L, Shen C. Application To Bearing Fault Diagnosis. Measurement [Internet]. 2016; Available from: http://dx.doi.org/10.1016/j.measurement.2016.07.054
18. Janssens O, Slavkovikj V, Vervisch B, Stockman K, Loccufier M, Verstockt S, et al. Convolutional Neural Network Based Fault Detection for Rotating Machinery. J Sound Vib. 2016;377:331–45.
19. Liu R, Meng G, Yang B, Sun C, Chen X. Dislocated Time Series Convolutional Neural Architecture: An Intelligent Fault Diagnosis Approach for Electric Machine. IEEE Trans Ind Informatics. 2017;13(3):1310–20.
20. Lu C, Wang Z, Zhou B. Intelligent fault diagnosis of rolling bearing using hierarchical convolutional network based health state classification. Adv Eng Informatics. 2017;32:139–51.
21. Wang H, Xu J, Yan R, Sun C, Chen X. Intelligent bearing fault diagnosis using multi-head attention-based CNN. Procedia Manuf. 2020;49:112–8. https://doi.org/10.1016/j.promfg.2020.07.005
22. Zilong Z, Wei Q. Intelligent fault diagnosis of rolling bearing using one-dimensional multi-scale deep convolutional neural network based health state classification. ICNSC 2018 - 15th IEEE Int Conf Networking, Sens Control. 2018;(April):1–6.
23. Li S, Liu G, Tang X, Lu J, Hu J. An ensemble deep convolutional neural network model with improved D-S evidence fusion for bearing fault diagnosis. Sensors (Switzerland). 2017;17(8).
24. Magar R, Ghule L, Li J, Zhao Y, Farimani AB. FaultNet: A Deep Convolutional Neural Network for Bearing Fault Classification. IEEE Access. 2021;9(October):25189–99.
25. Guo S, Yang T, Gao W, Zhang C, Zhang Y. An intelligent fault diagnosis method for bearings with variable rotating speed based on pythagorean spatial pyramid pooling CNN. Sensors (Switzerland). 2018;18(11).
26. Huang G, Liu Z, Van Der Maaten L, Weinberger KQ. Densely connected convolutional networks. Proc - 30th IEEE Conf Comput Vis Pattern Recognition. CVPR 2017. 2017:2261–9.

Acknowledgements: This research was jointly funded by National Key Research and Development Program of China, grant number 2019YFE0105300.

Julius Niyongabo: https://orcid.org/0000-0002-1429-9918

Yingjie Zhang: https://orcid.org/0000-0002-4170-6152

Jérémie Ndikumagende: https://orcid.org/0000-0003-1094-0066