

# Bearing Fault Identification Based on Deep Convolution Residual Network

Tong ZHOU\*, Yuan LI\*\*, Yijia JING\*\*\*, Yifei TONG\*\*\*\*

\*School of Mechanical Engineering, Nanjing University of Science and Technology, Nanjing 210000, People's Republic of China

\*\*Jiangyin Campus, Nanjing University of Science and Technology, Nanjing 214434, People's Republic of China

\*\*\*School of Mechanical Engineering, Nanjing University of Science and Technology, Nanjing 210000, People's Republic of China

\*\*\*\*School of Mechanical Engineering, Nanjing University of Science and Technology, Nanjing 210000, People's Republic of China, E-mail: tyf51129@aliyun.com

**crossref** <http://dx.doi.org/10.5755/j02.mech.28265>

## 1. Introduction

Bearings are important parts in industrial equipment, and the failure of bearing is one of the main factors leading to the shutdown of mechanical equipment [1]. Therefore, it is significantly important to implement real-time monitoring on the vibration signal of bearing and timely identify the fault type of the bearing from the signal.

Traditional fault identification is mostly based on signal processing. These classical models include two modules, feature extraction and fault classification based on machine learning (ML). Feature extraction maps the original signal to statistical parameters, which can convey information about the state of the machine. In order to obtain high-precision recognition results, the design of the feature extractor plays a very important role [2], because it is directly related to the performance of subsequent classification algorithms. Based on statistics [3], wavelet transform [4] and high order spectrums, impurities in the original data will be removed as much as possible. In this way, the feature specificity can be highlighted and the accuracy of fault diagnosis can be improved. However, these traditional methods often depend on professional prior knowledge [5], and the shallow structure of the conventional ML algorithm has very limited ability in learning to extract the nonlinear relationship of features [6].

In recent years, the rise of deep learning (DL) has promoted the development of artificial intelligence, and DL has obtained rich research results in the fields of speech recognition, image processing, and recommendation systems. Compared to traditional ML models, deep neural networks (DNN) contain more neural units and deep architecture, which can mine the more precious information from raw data. At present, there have been many related research results in the field of bearing fault identification. As deep feature extraction models, autoencoder (AE) [7, 8] and its variants such as stacked autoencoder (SAE) [9] and denoising autoencoder (DAE) [10], are used to extract fault features, and then, these extracted features are fed into classification model, which can obtain a high accuracy identification. In this situation, although DNN can obtain accurate feature representations, there is no difference from traditional learning mode. Actually, the models of DNN have a unique way of learning that is called end-to-end learning. In this way, DNN can directly give prediction results based on original data, avoiding the feature extraction [11].

Convolutional neural network (CNN), as a commonly used DNN, has achieved great success in the field of image recognition. According to the current literature, CNN has also been used for bearing fault identification. Because CNN is very good at processing two-dimensional images (2-D data), it is necessary to convert the vibration signal (1-D data) into 2-D data. A simple method is to directly treat the vibration signal of the bearing in time domain as an image with a width of 1, and use one-dimensional convolution for training [12, 13]. Sometimes the method of stacking the signal in time domain together to form a matrix can also be adopted [14]. Another method is to convert one-dimensional signals in the time domain into two-dimensional data in the frequency spectrum [15] or other 2-D format such as WPE image [16]. From the perspective of operation difficulty, the first method has a simpler data processing, and does not require two much prior knowledge about signal processing. Moreover, one-dimensional CNN is very effective in calculations and can be easily and cheaply implemented on hardware systems [17].

With advancement of research on DL, the architecture of DNN is continuously deepened, which causes the problem of gradient vanish and explosion and makes the DNN difficult to train. In order to solve this problem, He et al. [18] proposed residual network (ResNet) that is based on skip connection. Skip connection adds a degradation to DNN, because of the skip connection, DNN can degrade to the shallow neural networks, so the DNN can obtain performance equivalent to the shallow neural network. In the field of fault identification, ResNet have also been applied. Ma et al. [19] use the multi-objective optimization algorithm, fusing the ResNet with other neural networks and obtain a high accuracy. Jin et al. [20] proposes a decoupling attentional residual network for compound fault diagnosis and reach a very high accuracy on test set.

In these research above, the architecture of neural network is mostly shallow, which may make DNN unable to exert the powerful fitting ability. In addition, the design of the overall network is also very complicated.

In this paper, deep convolution residual network (DCRN) is proposed for bearing fault recognition, and the contributions of the proposed method are summarized below.

1. To ensure sufficient sample data, overlap sampling is used.
2. End-to-end learning is adopted, and input data is

one-dimension vibration signal in time domain, avoiding tedious signal processing and feature extraction.

3. By adding skip connection and stacking residual blocks, DCRN can achieve superior generalization performance on drive end data, the accuracy reaching about 99 %, and on fan end, the accuracy reached 100 %.

## 2. Methodology

### 2.1. Convolution

Different from the fully connected neural network, CNN has the characteristics of local connection and weight sharing [21], which greatly reduces the scale of network parameters and reduces the difficulty of training. In this paper, 1-D convolution will be applied, as shown in Fig.1.

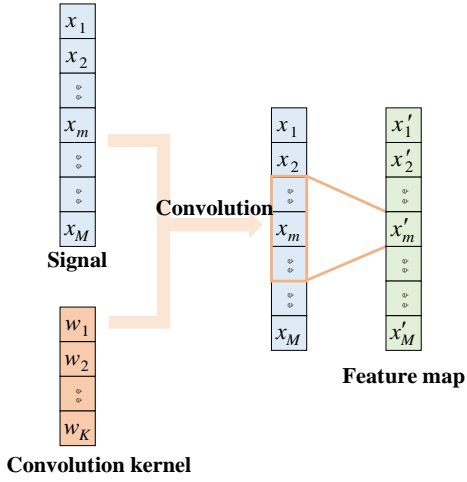


Fig. 1 1-D convolution

In Fig. 1, the blue bar represents the 1-D vibration signal which is viewed as the 2-D image with a width of 1, and the orange bar is convolution kernel, sometimes is called filter. The convolution kernel with weights  $\mathbf{w} = [w_1, w_2, \dots, w_K]^T$  slides on the signal  $\mathbf{x} = [x_1, x_2, \dots, x_M]^T$  according to a predetermined stride and calculating by Eq. (1), we can obtain feature map.

$$x'_m = f(\mathbf{w}^T \bar{\mathbf{x}} + b), \quad (1)$$

where:  $\bar{\mathbf{x}}$  is data points overlapping with the convolution kernel;  $b$  is bias;  $f(\cdot)$  is the nonlinear activation function, usually be the ReLU (recited linear unit) function.

Usually, when convolution is set to not change the size of input, padding will be necessary. Suppose kernel size is  $K \times 1$ , the padding will be  $\frac{K-1}{2}$ . To ensure padding number is an integer, kernel size must be odd number.

In Fig. 2, part (a) shows the two convolutions with  $3 \times 3$  kernel on  $5 \times 5$  image. Two  $3 \times 3$  convolution gives similar representation power as  $5 \times 5$  convolution. Those two convolution methods can reach the same receptive field, but first method, two  $3 \times 3$  convolution, is more popular, because only  $3 \times 3 \times 2 = 18$  parameters are required while  $5 \times 5$  convolution requires  $5 \times 5 = 25$  parameters. Generally speaking, fewer parameters are more conducive to updating the parameters of neural network. Moreover, two convolutional layers have more non-linear transformation than one convolutional layer.

The VGG network uses a large number of  $3 \times 3$  convolution kernels and obtain better performance on ImageNet. Thus,  $3 \times 3$  convolution gradually become mainstream.

However, when sample data is 1-D signal, something will be different. In Fig. 2b, two  $3 \times 1$  convolution layers can reach the same receptive field as one  $5 \times 1$  convolution layer. The former requires  $3 \times 1 \times 2 = 6$  parameters, the latter only requires  $5 \times 1 = 5$  parameters. The former has more parameters than the latter. Therefore, for 1-D signal, it is hard to say which kernel size is more suitable. In this paper, both convolution kernels will be tested

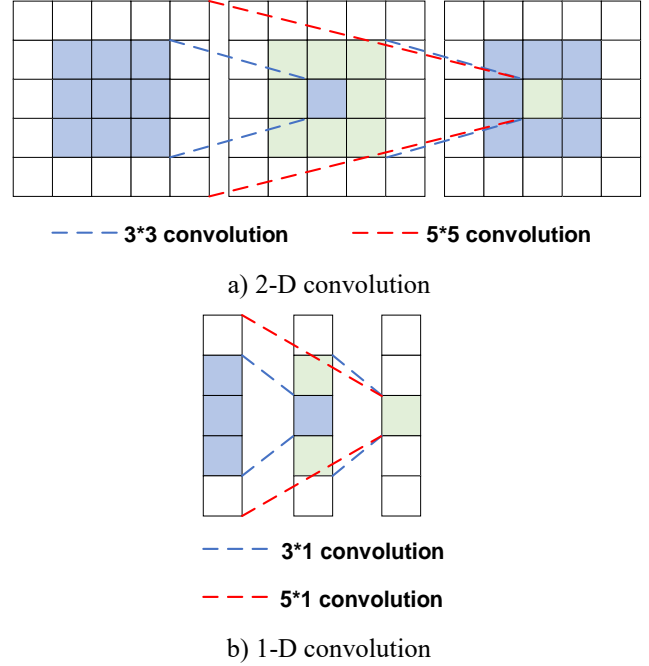


Fig. 2 Convolution on 2-D and 1-D data

### 2.2. Residual block

Based on design concept of residual network, skip connection is added between the input and output of neural layer, forming a residual block (ResBlock), as is shown in Fig. 3. After passing through neural layer, input  $\mathbf{x}$  is mapped to  $F(\mathbf{x})$ , then original input  $\mathbf{x}$  is added to  $F(\mathbf{x})$ , and output  $H(\mathbf{x})$ .

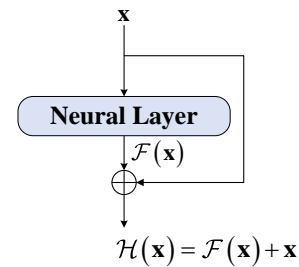


Fig. 3 Residual block

Sometimes skip connection may not bring original input  $\mathbf{x}$ , but  $\mathbf{x}'$  that is transformed from  $\mathbf{x}$ . Then,  $H(\mathbf{x})$  will be  $F(\mathbf{x}) + \mathbf{x}'$ .

In Section 2.1, two kinds of kernel size,  $3 \times 1$  kernel and  $5 \times 1$  kernel, are discussed. In this section, two kinds residual blocks with different convolution kernel sizes will be designed, as are shown in Fig.4. Because two  $3 \times 1$  conv layers can reach the same receptive field as one  $5 \times 1$  conv layer, in part (a), two  $3 \times 1$  conv layers are stacked and in part (c),

one 5\*1 conv layer is used. For further comparison, part (b) is also designed. Meanwhile, a batch normalization layer (BN) is added after each conv layer. The effect of BN layer is to obtain a smoother optimized landscape, which can improve optimization efficiency [22]. Besides, it is also a method of regularization, which can improve the generalization performance of the network [23]. Thus, dropout is not used in this paper, following the practice in [24].

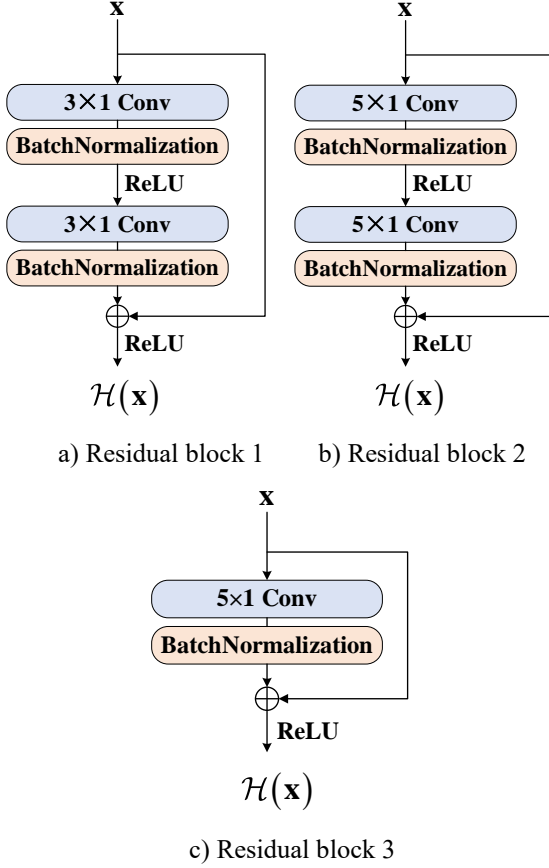


Fig. 4 Three kinds of residual blocks

### 2.3. The architecture

In CNN, commonly used down sampling methods are pooling and convolution with the stride equal to 2. Pooling on feature map can achieve more accurate feature representation, but this method is very time consuming in training. In ResNet [18], few pooling layer is used in the part of convolution. Inspired by the thought of ResNet, we build deep convolution residual networks by stacking residual blocks for bearing fault identification, as is shown in Fig. 4. There are three options for the residual block, as is shown in Fig. 4. The convolution layers follow two simple design rules [18]: i) for convolutional layers with the same output feature map size, the layers have the same number of filters; ii) if the feature map size is halved, the number of filters is double so as to preserve the time complexity. In Fig. 5, blue cells represent weighted layers, and green cells represent the un-weighted layers. Down sampling is mostly performed by convolution with a stride of 2. The network ends with a global average pooling layer and a dense layer with softmax. The dense layer possesses several units, the number of which is same as the number of fault type.

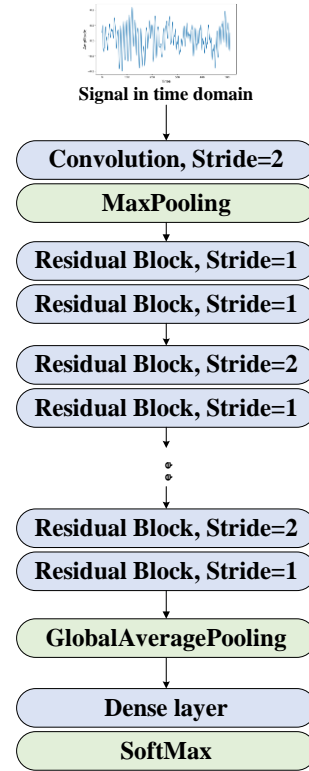


Fig. 5 Architecture of the DCRN

### 2.4. Updating weights of network

The DCRN in Fig. 5 is a feedforward neural network (FNN), and for this kind of network, the commonly used method of updating network weights is backpropagation algorithm. By using this algorithm, weights are updated according to the Eq. (2) until the network converges.

$$\theta' = \theta - \alpha \frac{\partial L}{\partial \theta}, \quad (2)$$

where:  $\theta$  and  $\theta'$  are the weights before and after updating respectively;  $L$  is the loss function of predicted value and ground-true value,  $\alpha$  is learning rate and decays over time.

## 3. Case study

### 3.1. Data pre-process

The data in this paper is the part of the open-source bearing data from Case Western Reserve University, using the drive end fault vibration signal and the normal bearing vibration signal that are collected at 12000 samples/second. In the original dataset, there are 3 types of bearing faults, containing inner raceway fault, rolling element fault and outer raceway fault. The signal data of the four kinds of signal is shown in Fig. 6. All fault types are artificially manufactured before the experiment.

In all working conditions, the lowest speed is 1730 rpm. Because data is sampled at a 12000 samples/second, when the shaft rotates for one cycle, about 416 ( $60/1730 \times 12000 = 416$ ) data points will be collected, that means one data period is 416 data points. The time span of a single sample should be related to the data period. Generally speaking, the time span of the sample needs to be longer than one data period, and  $k \times 2^n$  data points have the priority

to be selected as a sample point. Therefore, in this paper, continuous 512 data points ( $k=1, n=9$ ) are extracted as a single sample.

In order to obtain as many data samples as possible, overlapping sampling is considered, as is shown in Fig. 7. In this figure., stride represents the interval between two adjacent samples.

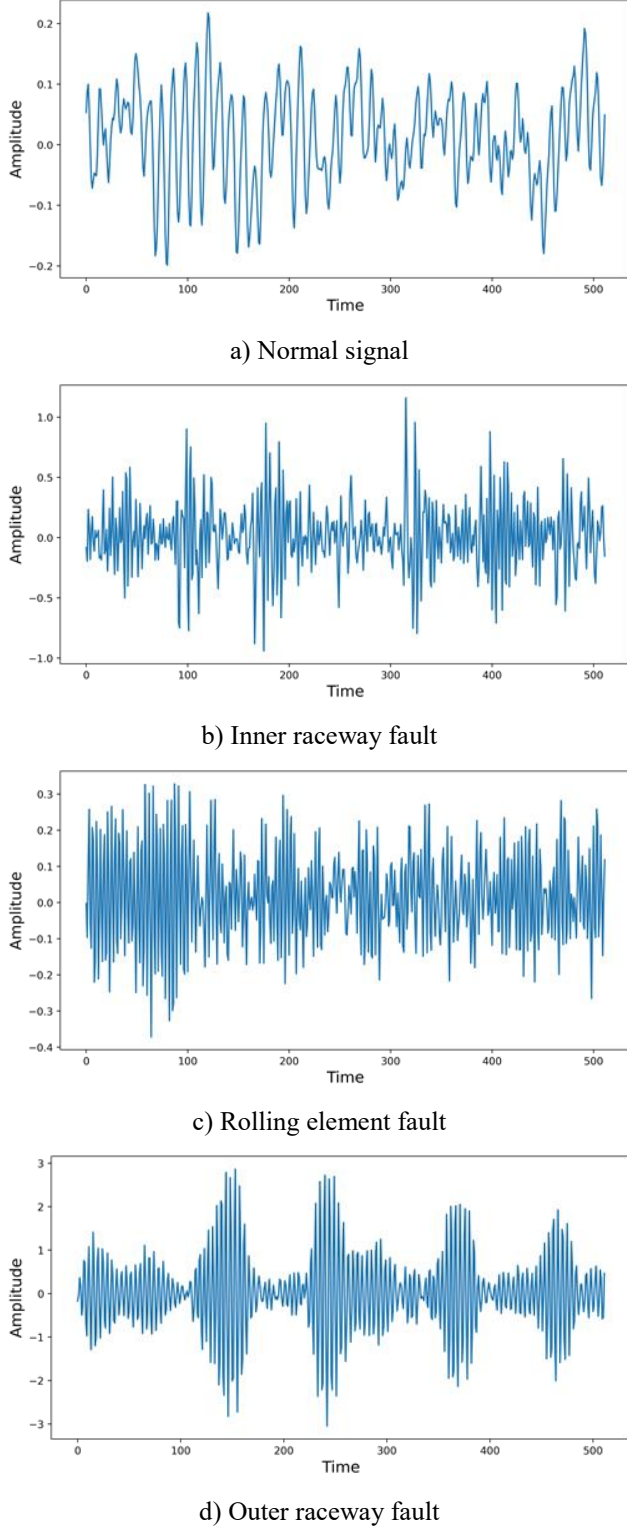


Fig. 6 Four kinds of vibration signals

Overlapping sampling will damage the independence between sample points, and the smaller the stride is, the weaker the independence between sample points will be. Therefore, the stride needs to be set to an appropriate value.

The number of data points in each mat file is inconsistent, so stride should change with different mat file, and can be adjusted according to Eq. (3):

$$\left\lfloor \frac{L-512}{stride} \right\rfloor + 1 = 1000, \quad (3)$$

where:  $L$  is the number of data points in the mat file,  $stride$  is the interval between two adjacent sample points.

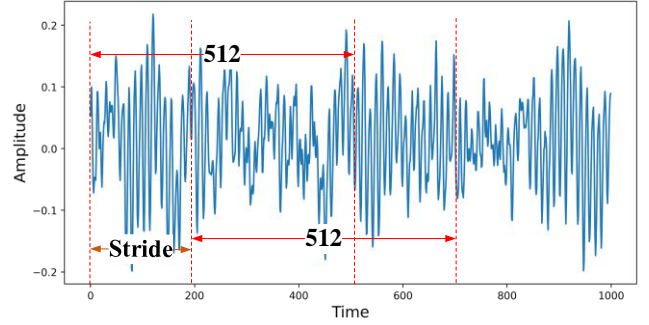


Fig. 7 Overlapping sampling

In order to improve the performance of network, a larger proportion of sample data can be selected as the training set. After the sample data set shuffled, 90 % of the sample data is randomly selected for training and the remaining 10 % for validation. The results of the division are shown in Table 1.

Table 1

Result of Dataset division

Fault type	Training set	Validation set	Total
Normal	3604	400	4004
Inner raceway	3613	387	4000
Ball	3607	393	4000
Outer raceway	3583	421	4004

According to Eq. (4), the sample data can be linearly scaled into  $[0,1]$  interval.

$$x\_pre_i = \frac{x_i - x\_train\_min}{x\_train\_max - x\_train\_min}, \quad (4)$$

where:  $x_i$  is  $i^{th}$  sample,  $x\_train\_min$  and  $x\_train\_max$  are the minimum and the maximum values in the training set.

### 3.2. Training strategy

Fig. 5 shows the rough network architecture, and some details can be seen in Table 2. In this experiment, three kinds architecture is designed. Tensor flow 2.3.0, a deep learning framework based on python, is used to write the program. For classification problems, the commonly used loss function is cross-entropy loss function, as shown in Eq. (5).

$$loss = -\sum_i y_i \log \hat{y}_i, \quad (5)$$

where:  $y_i$  is the real label of the sample that is encoded in one-hot encoding;  $\hat{y}_i$  is the probability that the sample belongs to the  $i^{th}$  fault type. Weights and biases are initialized as in [25]. Adam optimizer with a mini-batch size of 512 is

selected, two moving average decay rates are 0.9 and 0.999. The learning rate starts from 0.01 and is set with piecewise constant decay, as Fig. 8 shows.

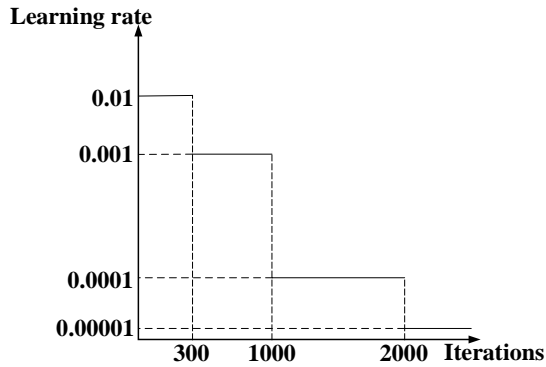


Fig. 8 Learning rate with piecewise constant decay

### 3.3. Results

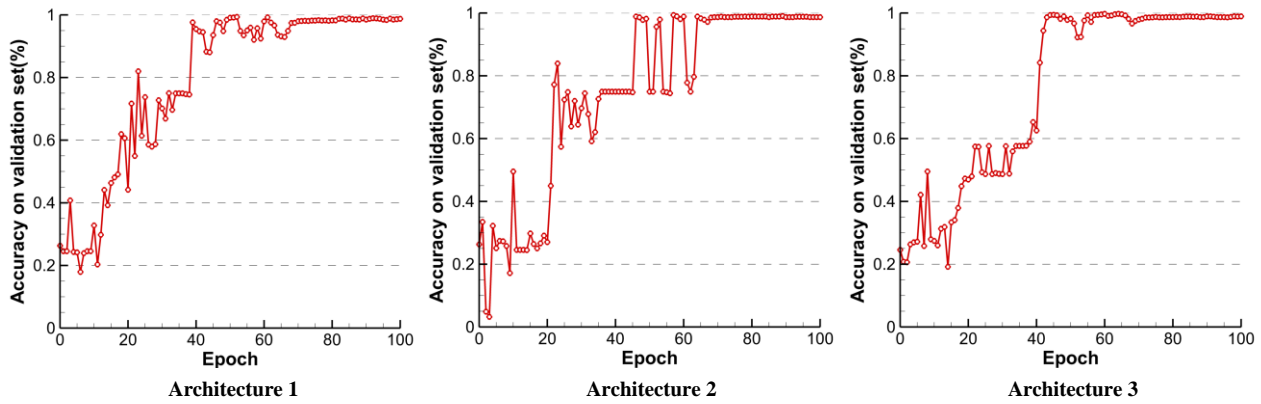
In this experiment, each architecture is trained for 100 epochs three times, and one of the results is in Fig. 9. The upper part of the figure is the accuracy on validation set, and the bottom is the confusion matrix on validation set. From these three training procedures, we can see that three different architectures can achieve very high accuracy on validation set and the average accuracy is shown in Table 3.

From the results, for convolution on 1-D signal, 5\*1 kernel size, instead of 3\*1, is more suitable. Both architecture 2 and architecture 3 have slightly better generalization performance than architecture 1. Besides, generalization performance of architecture 3 is slightly superior to architecture 2, this is because architecture 2 has three times as many parameters as architecture 3, and in this situation, architecture 2 is prone to slight overfitting.

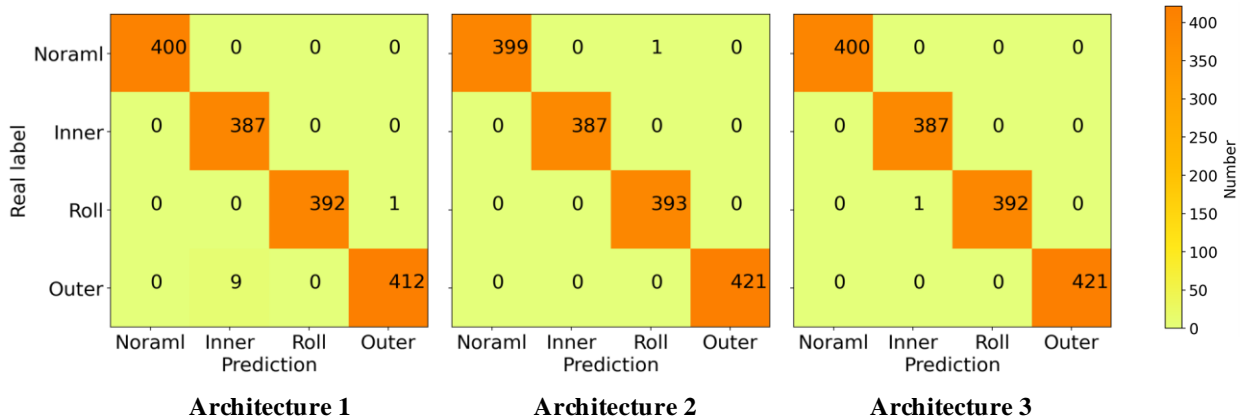
Table 2

Three different architectures of DCRN

Layer name	Output size	Architecture 1	Architecture 2	Architecture 3
Conv1	256×1×32	Conv(7×1, 32), stride 2		
Conv2	128×1×32	MaxPooling(3×1), stride 2		
Conv3	64×1×64	2×ResBlock(2×3×1, 32)	2×ResBlock(2×5×1, 32)	2×ResBlock(1×5×1, 32)
Conv4	32×1×128	2×ResBlock(2×3×1, 64)	2×ResBlock(2×5×1, 64)	2×ResBlock(1×5×1, 64)
Conv5	16×1×256	2×ResBlock(2×3×1, 128)	2×ResBlock(2×5×1, 128)	2×ResBlock(1×5×1, 128)
Conv6	8×1×512	2×ResBlock(2×3×1, 256)	2×ResBlock(2×5×1, 256)	2×ResBlock(1×5×1, 256)
Conv7	4×1×512	2×ResBlock(2×3×1, 512)	2×ResBlock(2×5×1, 512)	2×ResBlock(1×5×1, 512)
Pooling	1×1×512	GlobalAveragePooling		
	4×1	Dense layer with SoftMax		
Layer number		22	22	12
Total parameters		3,857,284	6,294,404	2,800,708



a) Accuracy on validation set



b) Confusion matrix on validation set

Fig. 9 Training procedure and prediction of three architecture

Table 3

Average accuracy of three architecture

	Architecture 1	Architecture 2	Architecture 3
Average accuracy, %	99.60	99.71	99.81

### 3.4. Discussion

Although deep convolution residual network achieves very good result, we cannot know the internal operation mechanism of the network. A popular explanation is that, deep neural network has a very powerful feature extraction ability. The network can be seen as a combination of feature extractor and classifier. Take architecture 3 as an example, the dense layer is dropped and the output of the global average pooling layer is visualized by T-SNE algorithm [26], as is shown in Fig. 10. We can clearly see that, after passing through several residual blocks, various fault types form clusters in high-dimensional space, which makes it easy for dense layer to identify fault types.

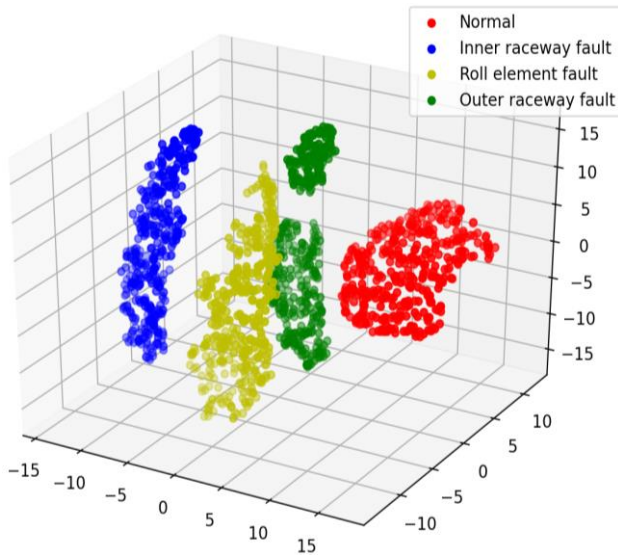


Fig. 10 Visualization by T-SNE algorithm

### 3.5. Validation on fan end

In order to further verify the robustness of the proposed method, the data on fan end is also chosen to test. We use the trained model of architecture 2. The model is trained on the data of drive end, so it has learned the knowledge about how to identify the fault type. With transfer learning [27], we do not need too many sample points on fan end, and only 882 sample points are required to obtain an excellent model (the batch size is 128 and learning rate starts from 0.01 with piecewise constant decay is also adopted). The result on the validation is shown in Fig. 11. Compared with the curve in Fig. 9, the curve in Fig. 11 possesses a high accuracy in the initial stage of training, because the initialized model has learned relevant knowledge. Throughout the entire training process, the highest identification accuracy can reach 100 %.

### 3.6. Comparison with other methods

In [20], multiple methods are tested on the data of fan end. Li et al. [28] used wavelet packet transform (WPT) to extract features and applied Support Vector Machine

(SVM) to classify faults for these features. Dhamande et al. [29] applied artificial neural network (ANN) on the signal in time domain. Some other DNN model is also tested, such as CNN [30] without skip connection and long-short term memory (LSTM), and the results is shown in Fig. 12. Compared with some other methods, the method proposed in this paper has the highest recognition accuracy.

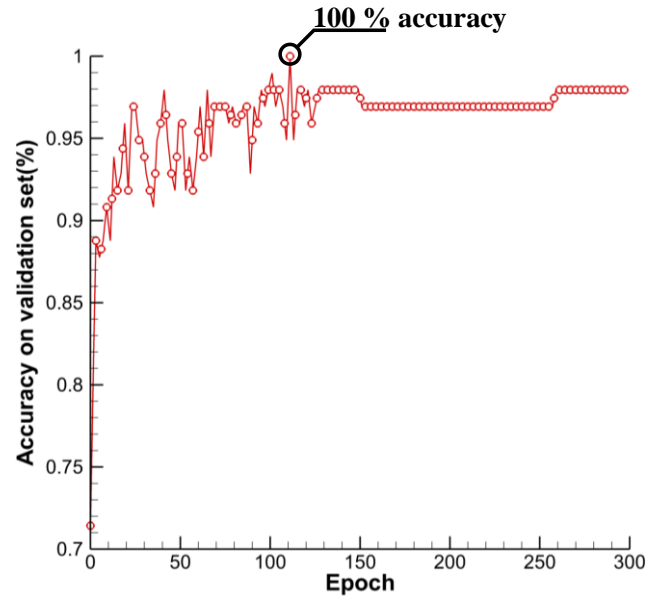


Fig. 11 Accuracy on validation set of the fan end

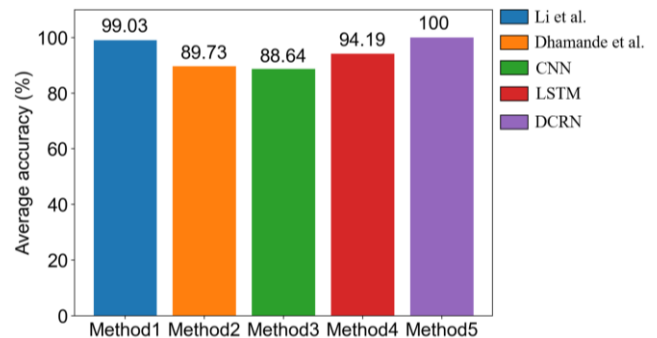


Fig. 12 Comparison with other methods

## 4. Conclusions

1. With the end-to-end learning, DCRN can directly be trained on 1-D signal in time domain, avoiding tedious signal processing and feature extraction.
2. In this paper, three different architecture of DCRN are designed, and all three achieve very high accuracy, respectively 99.60 %, 99.71 % and 99.81 %. Furthermore, the model is further tested on the fan end signal and the final result reached 100 %.
3. By simply stacking the residual blocks and using skip connection, DCRN can take better advantage of DNN. Therefore, DCRN has a good application prospect in the field of bearing fault identification.

## Acknowledgements

This work was financially supported by Fundamental Research Funds for the Central Universities (No. 30919011205, No. 30919011402).

## Statement

The author(s) declare(s) that there is no conflict of interests and the mentioned received funding in the Acknowledgement section did not lead to any conflicts of interest regarding the publication of this manuscript.

## References

1. **Li, J.; Yao, X.; Wang, X.; Yu, Q.; Zhang, Y.** 2020. Multiscale local features learning based on BP neural network for rolling bearing intelligent fault diagnosis, *Measurement* 153: 107419. <https://doi.org/10.1016/j.measurement.2019.107419>.
2. **Zhao, Z. Q; Huang, D. S.; Sun, B. Y.** 2004. Human face recognition based on multi-features using neural networks committee, *Pattern recognition letters* 25(12): 1351-1358. <https://doi.org/10.1016/j.patrec.2004.05.008>.
3. **Li, W.; Zhu, Z.; Jiang, F.; Zhou, G.;** 2015. Fault diagnosis of rotating machinery with a novel statistical feature extraction and evaluation method, *Mechanical Systems and Signal Processing* 50: 414-426. <https://doi.org/10.1016/j.ymsp.2014.05.034>.
4. **Wang, D.; Zhao, Y.; Yi, C.; Tsui, K. L.; Lin, J.** 2018. Sparsity guided empirical wavelet transform for fault diagnosis of rolling element bearings, *Mechanical systems and signal processing* 101: 292-308. <https://doi.org/10.1016/j.ymsp.2017.08.038>.
5. **Hoang, D. T.; Kang, H. J.** 2019. A survey on deep learning based bearing fault diagnosis, *Neurocomputing* 335: 327-335. <https://doi.org/10.1016/j.neucom.2018.06.078>.
6. **Jia, F.; Lei, Y.; Lin, J.; Zhou, X.; Lu, N.** 2016. Deep neural networks: A promising tool for fault characteristic mining and intelligent diagnosis of rotating machinery with massive data, *Mechanical Systems and Signal Processing* 72: 303-315. <https://doi.org/10.1016/j.ymsp.2015.10.025>.
7. **Mao, W.; He, J.; Li, Y.; Yan, Y.** 2017. Bearing fault diagnosis with auto-encoder extreme learning machine: A comparative study, *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science* 231(8): 1560-1578. <https://doi.org/10.1177/0954406216675896>.
8. **Shao, H.; Jiang, H.; Zhao, H.; Wang, F.** 2017. A novel deep autoencoder feature learning method for rotating machinery fault diagnosis, *Mechanical Systems and Signal Processing* 95: 187-204. <https://doi.org/10.1016/j.ymsp.2017.03.034>.
9. **Chen, Z.; Li, W.** 2017. Multisensor feature fusion for bearing fault diagnosis using sparse autoencoder and deep belief network, *IEEE Transactions on Instrumentation and Measurement* 66(7): 1693-1702. <https://doi.org/10.1109/TIM.2017.2669947>.
10. **Xia, M.; Li, T.; Liu, L.; Xu, L.; de Silva, C. W.** 2017. Intelligent fault diagnosis approach with unsupervised feature learning by stacked denoising autoencoder, *IET Science, Measurement & Technology* 11(6): 687-695. <https://doi.org/10.1049/iet-smt.2016.0423>.
11. **LeCun, Y.; Bengio, Y.; Hinton, G.** 2015. Deep learning, *nature* 521(7553): 436-444. <https://doi.org/10.1038/nature14539>.
12. **Ince, T.; Kiranyaz, S.; Eren, L.; Askar, M.; Gabbouj, M.** 2016. Real-time motor fault detection by 1-D convolutional neural networks, *IEEE Transactions on Industrial Electronics* 63(11): 7067-7075. <https://doi.org/10.1109/TIE.2016.2582729>.
13. **Eren, L.** 2017. Bearing fault detection by one-dimensional convolutional neural networks, *Mathematical Problems in Engineering*. <https://doi.org/10.1155/2017/8617315>.
14. **Guo, X.; Chen, L.; Shen, C.** 2016. Hierarchical adaptive deep convolution neural network and its application to bearing fault diagnosis, *Measurement* 93: 490-502. <https://doi.org/10.1016/j.measurement.2016.07.054>.
15. **Che, C.; Wang, H.; Ni, X.; Lin, R.** 2020. Hybrid multimodal fusion with deep learning for rolling bearing fault diagnosis, *Measurement* 108655. <https://doi.org/10.1016/j.measurement.2020.108655>.
16. **Ding, X.; He, Q.** 2017. Energy-fluctuated multiscale feature learning with deep convnet for intelligent spindle bearing fault diagnosis, *IEEE Transactions on Instrumentation and Measurement* 66(8): 1926-1935. <https://doi.org/10.1109/TIM.2017.2674738>.
17. **Farabet, C.; Poulet, C.; Han, J. Y.; LeCun, Y.** 2009. Cnp: An fpga-based processor for convolutional networks, 2009 International Conference on Field Programmable Logic and Applications. IEEE, 32-37. <https://doi.org/10.1109/FPL.2009.5272559>.
18. **He, K.; Zhang, X.; Ren, S.; Sun, J.** 2016. Deep residual learning for image recognition, *Proceedings of the IEEE conference on computer vision and pattern recognition* 770-778. <https://doi.org/10.1109/CVPR.2016.90>.
19. **Ma, S.; Chu, F.** 2019. Ensemble deep learning-based fault diagnosis of rotor bearing systems, *Computers in Industry* 105: 143-152. <https://doi.org/10.1016/j.compind.2018.12.012>.
20. **Jin, Y.; Qin, C.; Huang, Y.; Liu, C.** 2020. Actual bearing compound fault diagnosis based on active learning and decoupling attentional residual network, *Measurement* 108500. <https://doi.org/10.1016/j.measurement.2020.108500>.
21. **LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P.** 1998. Gradient-based learning applied to document recognition, *Proceedings of the IEEE* 86(11): 2278-2324. <https://doi.org/10.1109/5.726791>.
22. **Santurkar, S.; Tsipras, D.; Ilyas, A.; Madry, A.** 2018. How does batch normalization help optimization? *Advances in Neural Information Processing Systems* 2483-2493. <https://arxiv.org/abs/1805.11604v5>.
23. **Luo, P.; Wang, X.; Shao, W.; Peng, Z.** 2018. Towards understanding regularization in batch normalization, *arXiv preprint*. <https://arxiv.org/abs/1809.00846>.
24. **Ioffe, S.; Szegedy, C.** 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift, *arXiv preprint*. <https://arxiv.org/abs/1502.03167>.
25. **He, K.; Zhang, X.; Ren, S.; Sun, J.** 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification, *Proceedings of the IEEE international conference on computer vision* 1026-1034. <https://arxiv.org/abs/1502.01852>.

26. **Van der Maaten, L.; Hinton, G.** 2008. Visualizing data using t-SNE, *Journal of machine learning research*, 9(Nov): 2579-2605. <http://jmlr.org/papers/v9/vandermaaten08a.html>.
27. **Pan, S. J.; Tsang, I. W.; Kwok, J. T.; Yang, Q.** 2010. Domain adaptation via transfer component analysis, *IEEE Transactions on Neural Networks* 22(2): 199-210. <https://doi.org/10.1109/TNN.2010.2091281>.
28. **Yonglong, L.; Renping, S.; Jingming, C.** 2010. A New and Effective Method of Gear Fault Diagnosis Using Wavelet Packet Transform Combined with Support Vector Machine, *Journal of Northwestern Polytechnical University*, 4. <https://doi.org/10.4304/jcp.6.11.2502-2509>.
29. **Dhamande, L. S.; Chaudhari, M. B.** 2018. Compound gear-bearing fault feature extraction using statistical features based on time-frequency method, *Measurement* 125: 63-77. <https://doi.org/10.1016/j.measurement.2018.04.059>.
30. **Appana, D. K.; Ahmad, W.; Kim, J. M.** 2017. Speed invariant bearing fault characterization using convolutional neural networks, *International Workshop on Multi-disciplinary Trends in Artificial Intelligence*. Springer, Cham 189-198. [https://doi.org/10.1007/978-3-319-69456-6\\_16](https://doi.org/10.1007/978-3-319-69456-6_16).

T. Zhou, Y. Li, Y. Jing, Y. Tong

## BEARING FAULT IDENTIFICATION BASED ON DEEP CONVOLUTION RESIDUAL NETWORK

### S u m m a r y

Bearings are important parts in industrial production and are related to the normal operation of mechanical equipment. For bearing fault identification, traditional method often includes feature extraction, which involves professional prior knowledge and is time-consuming. This paper proposes the deep convolution residual network (DCRN) to identify the bearing fault. Based on the end-to-end learning characteristics of deep neural networks, this method can directly use raw data for training, and does not require feature extraction. Moreover, under the effect of skip connection, DCRN can exert the powerful fitting ability of deep neural network. In this paper, by stacking residual blocks, three different architecture of DCRN are designed and all three achieve very high accuracy, respectively 99.60%, 99.71% and 99.81%. Compared with other methods, DCRN have better generalization performance.

**Keywords:** bearing fault identification; residual block; deep convolution residual network; end-to-end learning; deep learning.

Received January 06, 2021

Accepted June 02, 2021



This article is an Open Access article distributed under the terms and conditions of the Creative Commons Attribution 4.0 (CC BY 4.0) License (<http://creativecommons.org/licenses/by/4.0/>).