Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 4319074, 12 pages
https://doi.org/10.1155/2021/4319074

Research Article

Distilling the Knowledge of Multiscale Densely Connected Deep Networks in Mechanical Intelligent Diagnosis

Xiaochuan Wang,[1] **Aiguo Chen**,[1] **Liang Zhang**,[1,2] **Yi Gu**,[1] **Mang Xu**,[1] **and Haoyuan Yan**[1]

[1]*School of Artificial Intelligence and Computer Science, Jiangnan University, Wuxi, Jiangsu 214122, China*
[2]*School of Biotechnology, Jiangnan University, Wuxi, Jiangsu 214122, China*

Correspondence should be addressed to Aiguo Chen; agchen@jiangnan.edu.cn

Received 24 May 2021; Revised 20 June 2021; Accepted 22 June 2021; Published 7 July 2021

Academic Editor: Shan Zhong

At present, deep neural network (DNN) technology is often used in intelligent diagnosis research. However, the huge amount of calculation of DNN makes it difficult to apply in industrial practice. In this paper, an advanced multiscale dense connection deep network MSDC-NET is designed. A well-designed multiscale parallel branch module is used in the network. This module can greatly improve the acceptance domain of MSDC-NET, so as to learn useful information from input samples more effectively. Based on the inspiration of Densely Connected Convolutional Networks, MSDC-NET designed a similar dense connection technology, so that the model will not have the problem of gradient vanishing because of the deep network. The experimental data of MSDC-NET on MFPT, SEU, and Pu datasets show that our method has higher performance than other latest technologies. At the same time, we carried out knowledge distillation based on the high-precision classification level of MSDC-NET, which makes the diagnosis ability and robustness of the lightweight CNN model improve significantly.

1. Introduction

The mechanical equipment in modern industry often needs to work in the complex environment of high temperature, fatigue, and heavy load for a long time, which may cause incalculable production accidents and economic property losses. At present, most enterprises use the way of manual supervision, which consumes a lot of human and material resources. Since mechanical failures are often not so obvious, only technicians with professional knowledge can be sure to judge mechanical failures. As a mature unsupervised system, intelligent diagnosis can solve this problem for related enterprises. The traditional intelligent diagnosis method mainly uses a machine learning algorithm to classify the feature data from the sensor signal. For example, Zhang et al. used support vector machine (SVM) to identify the working state of bearing [1]; Banerjee and Das used SVM fusion multisensor signal to detect motor fault [2]; Bugharbee and Trendafilova used the nearest neighbor classifier to identify faults [3]; Keskes et al. studied the method of feature extraction using stationary wavelet packet transform, and support vector machine (SVM) is applied to rotor fault classification. [4]; Mustafa et al. proposed that the spectrum analysis method be used in feature extraction and SVM be used to detect mechanical faults. [5]. However, the signal obtained by sensor needs complex feature extraction, which is difficult to achieve instantaneity, and depends on professional knowledge and relevant experience.

With the rapid development of deep learning (DL) technology in many fields, researchers have begun to use some DL-based technologies to achieve intelligent diagnosis, such as Multilayer Perceptron (MLP) [6], Deep Belief-Net (DBN) [7], Convolutional Neural Net (CNN) [8],

Autoencoder-Net (AE) [9], and Recurrent Neural Network (RNN) [10]. The MLP model is a very simple perceptron model. The knowledge it can learn is very scarce, and it has been rarely used at present. As a probabilistic generation model relative to the traditional discriminant model, DBN is composed of multiple restricted Boltzmann machine layers; Shao et al. proposed to use the optimized DBN model to diagnose rolling bearings [11]; Tamilselvan and Wang proposed a DBN model for engine fault diagnosis [12]. The improvement of the CNN model is mainly in two directions. One is to use different input types, such as two-dimensional images [13], infrared thermal images [14], vibration spectrum images [15], and time-frequency images [16]. Another direction is to improve the structure of CNN model, such as combining the CNN with Generative Adversarial Network (GAN [17]) to generate more new labeled samples [18, 19]. Janssens et al. proposed a CNN model to classify the faults of rotating machinery [20]. Recently, some researchers use transfer learning [21, 22] combined with the CNN to obtain the prior knowledge of the original dataset [23–25]. The AE model is composed of multiple automatic encoders, and there are two common extended models, such as denoising AE (DAE) [26] and compression AE (CAE) [27]. The advantage of DAE is that it can learn useful information from damaged data, and CAE learns more stable feature representations through penalty items. The latest AE model also combines with the GAN network to generate labeled samples [28–30] and also embeds the semisupervised learning method into the VAE model [31, 32]. The Recurrent Neural Network has no advantage in classification, and it is more commonly used in mechanical life prediction.

So as to evaluate the effectiveness and fairness of the model, our experiment is carried out in the benchmark code base proposed by Zhao et al. [33]. They conducted a comprehensive benchmark study on the realization of intelligent diagnosis based on the DL model. The study evaluated the intelligent diagnosis performance of nine commonly used models on nine publicly available datasets. The nine models include three encoder models, such as AE, DAE, and CAE, including the MLP model and CNN model and two advanced deep classification networks AlexNet [34], ResNet18 [35], and finally include an LSTM [36] model. The results of these studies show the significant advantages of DL in intelligent diagnosis, especially the models that have outstanding performance in other classification problems such as ResNet18, which also have good results when transplanted to intelligent diagnosis. However, the scale of these deep classification networks is very large, and the computing power of the equipment is relatively high. How to improve the accuracy of model intelligent diagnosis in different accuracy requirements and equipment environment is a worthy research direction.

The main target of our method is to use a new and powerful deep classification network as the powerful model to guide the lightweight model to realize mechanical intelligent diagnosis. Therefore, our main contribution can be outlined in two aspects:

(1) We designed a new multiscale densely connected deep network MSDC-NET, and the experimental results of five comparison algorithms on three types of datasets show that the model has higher superiority

(2) We innovatively use the knowledge distillation technology in the mechanical intelligent diagnosis subject, which makes the classification accuracy of the lightweight CNN model significantly improve and provides an effective solution to achieve the highest level of diagnosis in the case of limited computing power

The experiments in this paper are carried out on the open MFPT bearing dataset (https://mfpt.org/fault-data-sets/), Pu bearing dataset (https://mb.uni-paderborn.de/kat/forschung/datacenter/bearing-datacenter/), and SEU transmission dataset (https://github.com/cathysiyu/Mechanical-datasets).

The rest of the paper is arranged as follows: Section 2 retrospects the related work of deep classification network and knowledge distillation technology, and then in Section 3, we introduce our MSDC-NET model, knowledge distillation process, and its algorithm in detail. After that, the experimental results obtained on three common datasets are discussed and analyzed in Section 4. Finally, the article is summarized in the Section 5.

2. Related Work

2.1. Deep Convolution Network. The CNN, which was proposed in 1997, is a classification network for processing labeled data. It can transfer, extract, and learn information through convolution and pool operations. AlexNet won the championship in the 2012 ImageNet competition, and the ResNet proposed in 2016 has surpassed human classification accuracy for the first time.

The CNN model is generally composed of a convolution layer, pooling layer (maximum pooling layer, average pooling layer), and fully connected layer. We often use the convolution layer and maximum pooling layer with a step size of 2 to learn features and finally classify features through full connection layer.

The convolution layer operation is equivalent to the multiplication of input x and convolution kernel w, which is then represented by the activation function mapping:

$$h_k^l = \phi\left(w_k^l \cdot x + b_k^l\right), \tag{1}$$

where ϕ is the convolution operator, h_k^l is the output of convolution operation, k is the k-th convolution kernel, l is the l -th layer, and w_k^l and b_k^l are the weight and bias.

The purpose of maximum pooling layer is to learn the most essential knowledge in the local acceptance domain, focus on texture information, and reduce the bias of estimation mean caused by convolution layer parameter error. The purpose of average pooling layer is to consider the characteristics of step size, focus on the background information, and reduce the estimation variance.

As the number of layers of CNN tends to be deeper, the problem of model degradation may arise, and jump connection technology is often used to avoid gradient disappearance. Meanwhile, the multiscale technology widely used in

CV field can also be transplanted into the CNN to improve the receptive field of network model.

2.2. Distilling the Knowledge. In the industrial application, in addition to the requirement that the model should have as high a prediction level as possible, it is also expected that the expenditure of the model should be as small as possible, so that the deployment needs the least computing resources (computing power, storage space) and has a lower delay. Knowledge distillation [37] is a solution to this problem. First, we need a powerful and effective pretraining model (T-Model), then train a lightweight model (S-Model) from scratch and pass on the knowledge contained in a T-Model to an S-Model during the training process.

The above process of knowledge transfer only requires that the "SoftMax" output distribution of S-Model and T-Model under a given input be fully closed.

$$q_i = \frac{\exp\left(z_i/\mathrm{TEMP}\right)}{\sum_j \exp\left(z_j/\mathrm{TEMP}\right)}, \tag{2}$$

where q_i is the output distribution of S-Model, z_i is the calculated probability for each class in S-Model, and TEMP is a parameter usually set to 1. When TEMP becomes larger, we get a softer output. For cross-entropy loss, the gradient for a logit of S-Model is as follows:

$$\frac{\partial C}{\partial z_i} = \frac{1}{\mathrm{TEMP}}\left(q_i - p_i\right) = \frac{1}{\mathrm{TEMP}}\left(\frac{e^{z_i/\mathrm{TEMP}}}{\sum_j e^{z_j/\mathrm{TEMP}}} - \frac{e^{c_i/\mathrm{TEMP}}}{\sum_j e^{c_j/\mathrm{TEMP}}}\right), \tag{3}$$

where p_i is the output distribution of the T-Model, c_i is the calculated probability for each class in the T-Model, and when x tends to 0, it is equivalent to infinitesimal with $e^x - 1$. Easy to know, when TEMP is sufficiently large, there are

$$\frac{\partial C}{\partial z_i} \approx \frac{1}{\mathrm{TEMP}}\left(\frac{1 + \left(z_i/\mathrm{TEMP}\right)}{N + \sum_j z_j/\mathrm{TEMP}} - \frac{1 + \left(c_i/\mathrm{TEMP}\right)}{N + \sum_j c_j/\mathrm{TEMP}}\right). \tag{4}$$

Suppose that every sample has Logits with zero mean, that is $\sum_j z_j = \sum_j c_j = 0$, then:

$$\frac{\partial C}{\partial z_i} \approx \frac{1}{N\mathrm{TEMP}^2}\left(z_i - c_i\right). \tag{5}$$

3. Our Proposed Method

3.1. The Composition of the Blocks. To ensure the performance of the model, we have carefully designed the neural network modules shown in Figures 1–3, through the stacking of these modules constituting the final MSDC-NET.

Go_Down cell is proposed to achieve a similar dense connection effect, which can maximize the retention of the original lore of the previous data, and realize the superposition of the previous and subsequent feature maps in the channel dimension. The "times" represents the multiple relationship of the size of the previous and subsequent feature maps. After "times" max pooling, the dimension of the feature map will be reduced to 1/times of the original size.

DS cell is proposed for downsampling of the feature map, which is implemented by convolution operation (the

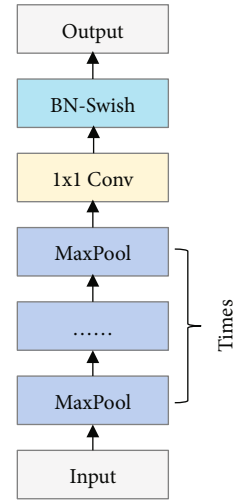


Figure 1: The construction of a Go_Down cell.

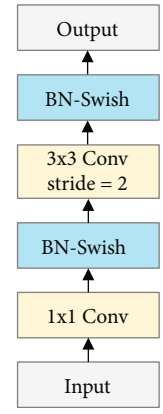


Figure 2: The construction of a DS cell.

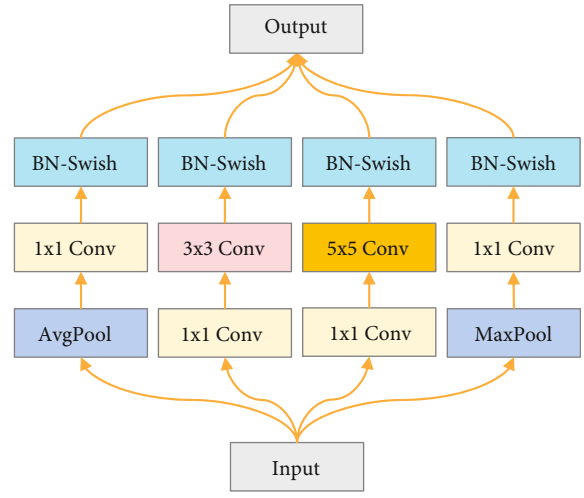


Figure 3: The construction of an MS_Block cell.

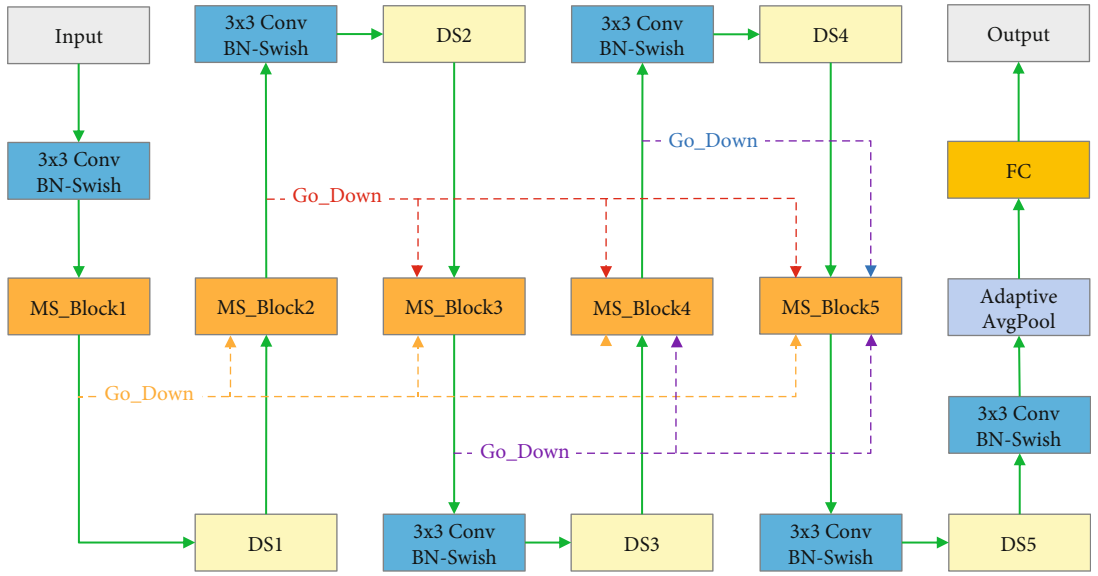


Figure 4: The structure of the MSDC-NET.

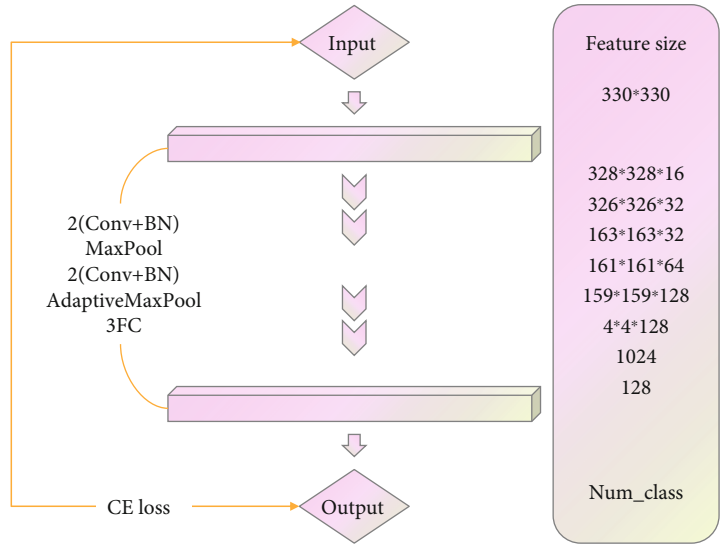


Figure 5: The structure of the CNN.

parameters are as follows: stride = 2 and kernel_size = 3 × 3). The dimension of the input data will be decreased by half, and the number of channels will be doubled.

MS_Block cell draws on the idea of inception cell [38], in which 3 × 3 and 5 × 5 convolution kernels are used for multiscale learning feature map, and average pooling and maximum pooling are used to extract useful information from multiple angles. After the multibranch network structure has learned the input data, a complete feature map of each branch superimposed and fused in the channel dimension will be obtained, and then, this feature map will be placed in the MSDC network to continue learning.

3.2. The Structure of MSDC-NET. For the purpose of get a high-precision Teacher-Model, we integrated the most prevalent multiscale and dense connection technology to advance a split-new deep classification network MSDC-NET. It is made up of our carefully designed stack of various modules. Taking the input data size AB as an example, five MS_Block cells are set in the model to realize multiscale and multiangle learning sample data information. At the same time, five DS cells are set for downsampling, and the size of the top feature map is AC. In practical applications, we can adjust the network model reasonably according to the input data size. When the input data is large, more stacking blocks can be set, and on the contrary, the number of layers can be set fewer.

However, only through the one-way propagation of this order often fails to achieve a good classification effect, because the deep network will forget and lose a lot of previous

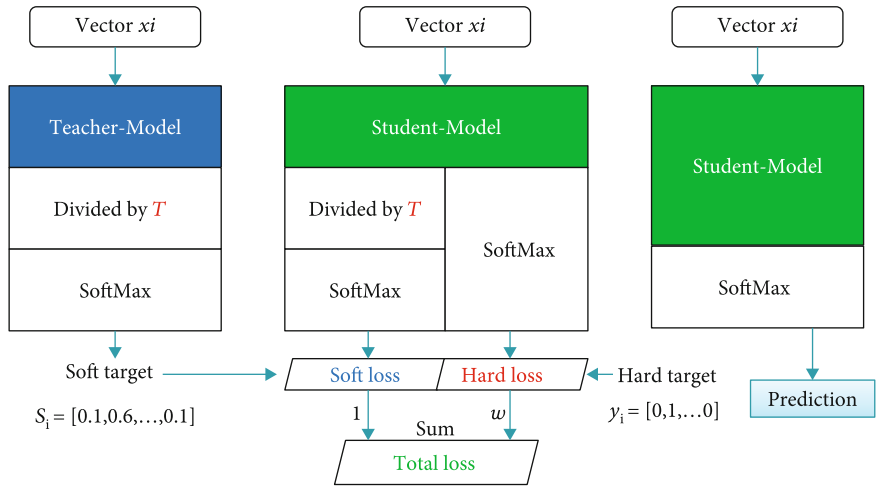


Figure 6: Flow chart of knowledge distillation technology.

Input: Sample X, Sample category, Test sample $\tilde{x}$
Step1: The training process of the MSDC- NET model:
1) Processing sample input data:
Unify the sample data X into the same segmentation method, input type and input size, and finally we get the input as (x, y).
2) Train the MSDC-NET model:
Train MSDC-NET through the processed training data, and the objective function (6) as:
$\min loss = -\sum_{i=1}^{n}(l \log(\tilde{l}) + (1-l)\log(1-\tilde{l}))$
Step2: The training process of the CNN model:
1) Processing sample input data:
2) Train the CNN model:
Train CNN through the processed training data, and the objective function (10) as
$\min loss = \omega * loss\_hard + loss\_soft$
Step3: Fault classification
1) Output of MSDC-NET (or CNN):
Input the test data sample $\tilde{x}$ into the pre-trained model (MSDC-NET or CNN) to get the output $\tilde{y}$.
2) Category determination
After obtaining the output $\tilde{y}$ of the model (MSDC-NET or CNN), use Formula (7) to predict the category of the test sample:
$label(\tilde{x}) = \arg\max(\tilde{y})$
Output: Predict the category of input x

Algorithm 1: Process of MSDC-NET (or CNN) for fault classification

knowledge and information. We need a unique cascade operation to fuse the same resolution features of different stages in the channel dimension, so we designed a technology similar to dense connection. The implementation of this technology relies on Go_Down cell, through which we save the output feature map of each MS_Block cell. The feature maps obtained by Go_Down of different "times" can be merged with the small feature map of matching size later, which is able to speed up the convergence of the network model, while improving the robustness of the model and avoiding problems such as gradient disappearance.

The structure of the MSDC-NET model we proposed is shown in Figure 4. The main road after the data sample enters the model is the green one-way line. A network training procedure using reasonable and effective loss function is crucial; the output $\tilde{l}$ is obtained from the top-level feature graph through Adaptive AvgPool and FC layers. If the class label of data sample is l, the loss function of model training is as follows:

$$loss(Cross\ entropy) = -\sum_{i=1}^{n}\left(l \log\left(\tilde{l}\right) + (1-l)\log\left(1-\tilde{l}\right)\right). \tag{6}$$

Since the quantity of layers of the DNN is relatively deep, the useful information contained in the first data entering the network and the first feature map may be forgotten and lost. We borrowed the idea of DenseNet [39] in the MSDC-NET model and, after the foregoing features of FIG different sizes corresponding to multiples by scaling Go_Down cell extract, then merged them with the feature maps of the back of the

Table 1: Fault classification of PU dataset.

Code	Fault mode: description
K001	Health state: launched 50 hours in advance
K002	Health state: launched 19 hours in advance
K003	Health state: launched 1 hours in advance
K004	Health state: launched 5 hours in advance
K005	Health state: launched 10 hours in advance
K006	Health state: launched 16 hours in advance
KA01	Outer ring: man-made damage caused by EDM (Level 1)
KA03	Outer ring: man-made damage caused by electric engraver (Level 2)
KA05	Outer ring: man-made damage caused by electric engraver (Level 1)
KA06	Outer ring: man-made damage caused by electric engraver (Level 2)
KA07	Outer ring: man-made damage caused by drilling (Level 1)
KA08	Outer ring: man-made damage caused by drilling (Level 2)
KA09	Outer ring: man-made damage caused by drilling (Level 2)
KI01	Inner ring: man-made damage caused by EDM (Level 1)
KI03	Inner ring: man-made damage caused by electric engraver (Level 1)
KI05	Inner ring: man-made damage caused by electric engraver (Level 1)
KI07	Inner ring: man-made damage caused by electric engraver (Level 2)
KI08	Inner ring: man-made damage caused by electric engraver (Level 2)
KA04	Outer ring: damage caused by fatigue and pitting (single point and single damage and Level 1)
KA15	Outer ring: damage caused by plastic deform and indentation (single point and single damage and Level 1)
KA16	Outer ring: damage caused by fatigue and pitting (single point and repetitive damage and Level 2)
KA22	Outer ring: damage caused by fatigue and pitting (single point and single damage and Level1)
KA30	Outer ring: damage caused by plastic deform and indentation (distributed and repetitive damage and Level 1)
KB23	Outer ring, inner ring: damage caused by fatigue and pitting (single point and multiple damage and Level 2)
KB24	Outer ring, inner ring: damage caused by fatigue and pitting (distributed and multiple damage and Level 3)
KB27	Outer ring, inner ring: damage caused by fatigue and pitting (distributed and multiple damage and Level 1)
KI04	Inner ring: damage caused by fatigue and pitting (single point + single damage + Level 1)
KI14	Inner ring: damage caused by fatigue and pitting (single point + multiple damage + Level 1)
KI16	Inner ring: damage caused by fatigue and pitting (single point + single damage + Level 3)
KI17	Inner ring: damage caused by fatigue and pitting (single point + repetitive damage + Level 1)
KI18	Inner ring: damage caused by fatigue and pitting (single point + single damage + Level 2)
KI21	Inner ring: damage caused by fatigue and pitting (single point + single damage + Level 1)

Table 2: Fault classification of MFPT dataset.

Category	Description	Category	Description
Health state	270 lb, 25 Hz, 97656 sps,6 s	/	/
Outer ring 1	25 lb, 25 Hz, 48828 sps, 3 s	Inner ring 1	25 lb, 25 Hz, 48828 sps, 3 s
Outer ring 2	50 lb, 25 Hz, 48828 sps, 3 s	Inner ring 2	50 lb, 25 Hz, 48828 sps, 3 s
Outer ring 3	100 lb, 25 Hz, 48828 sps, 3 s	Inner ring 3	100 lb, 25 Hz, 48828 sps, 3 s
Outer ring 4	150 lb, 25 Hz, 48828 sps, 3 s	Inner ring 4	150 lb, 25 Hz, 48828 sps, 3 s
Outer ring 5	200 lb, 25 Hz, 48828 sps, 3 s	Inner ring 5	200 lb, 25 Hz, 48828 sps, 3 s
Outer ring 6	250 lb, 25 Hz, 48828 sps, 3 s	Inner ring 6	250 lb, 25 Hz, 48828 sps, 3 s
Outer ring 7	300 lb, 25 Hz, 48828 sps, 3 s	Inner ring 7	300 lb, 25 Hz, 48828 sps, 3 s

Table 3: Fault classification of SEU dataset.

Sort	Fault mode (RS-LC:20 Hz-0 V)	Sort	Fault mode (RS-LC:3 Hz-2 V)
1	Health gear	11	Health gear
2	Health bearing	12	Health bearing
3	Chipped tooth	13	Chipped tooth
4	Inner ring	14	Inner ring
5	Missing tooth	15	Missing tooth
6	Outer ring	16	Outer ring
7	Root fault	17	Root fault
8	Inner + outer rings	18	Inner + outer rings
9	Surface fault	19	Surface fault
10	Rolling element	20	Rolling element

network. And we let them influence the training process of the network to varying degrees according to the set weights.

Finally, the category of input data is forecasted by the following formula:

$$label(x) = \arg\max\left(\tilde{l}\right). \tag{7}$$

3.3. CNN after Knowledge Distillation. We use the simple CNN model shown in Figure 5. With the input by way of multiple stacked Conv layers and Pool layers, it enters the classifier to obtain the final output.

The model contains only 4 convolutional layers in total, with a very simple structure and very low requirements on the computing power and storage of the deployed equipment, so it is suitable to run in the industrial environment.

3.4. Algorithm Details. In Section 2, we describe the process of knowledge distillation in detail. After knowledge distillation, the CNN model can obtain $loss\_soft$ according to formula (5). (In the experiment, we set T as 10 according to the relevant literature.)

$$loss\_soft = \frac{1}{NT^2}(z_i - v_i). \tag{8}$$

According to the previous formula (6) of MSDC-NET, we can get $loss\_hard$ as follows:

$$loss\_hard = -\sum_{i=1}^{n}\left(l \log\left(\tilde{l}\right) + (1-l)\log\left(1-\tilde{l}\right)\right). \tag{9}$$

Finally, the total loss function of knowledge distillation process is (where ω is the weight of $loss\_hard$)

$$loss = \omega * loss\_hard + loss\_soft. \tag{10}$$

As shown in Figure 6, after the Teacher-Model is divided by the temperature parameter T, the soft label is obtained through the "SoftMax" transformation, and the value in the label is between 0 and 1. The larger the temperature parameter T, the softer the distribution. On the contrary, it is easy to introduce unnecessary noise and amplify the probability of misclassification. We need to make sure that the correct predicted contribution is made in the Teacher-Model, so set T to a larger value. The real label in the sample is converted to a one-hot vector and used as a hard label. The total loss is the weighted sum of the cross entropy of the soft label and the predicted value and the cross entropy of the hard label and the predicted value. If the value of ω is smaller, it indicates that more attention is paid to the contribution of Teacher-Model; on the contrary, let the Student-Model pay more attention to the identification of difficult samples. Therefore, the higher the classification accuracy of the teacher model, the more conducive to student model learning samples. Our specific algorithm is reflected in Algorithm 1.

Algorithm 1: Process of MSDC-NET (or CNN) for fault classificationInput: Sample X, Sample category, Test sample $\tilde{x}$

Step1: The training process of the MSDC- NET model:

1) Processing sample input data:

Unify the sample data X into the same segmentation method, input type and input size, and finally we get the input as (x, y).

2) Train the MSDC-NET model:

Train MSDC-NET through the processed training data, and the objective function (6) as:

$$\min loss = -\sum_{i=1}^{n}(l \log(\tilde{l}) + (1-l)\log(1-\tilde{l}))$$

Step2: The training process of the CNN model:

1) Processing sample input data:

2) Train the CNN model:

Train CNN through the processed training data, and the objective function (10) as

$$\min loss = \omega * loss\_hard + loss\_soft$$

Step3: Fault classification

1) Output of MSDC-NET (or CNN):

Input the test data sample $\tilde{x}$ into the pre-trained model (MSDC-NET or CNN) to get the output $\tilde{y}$.

2) Category determination

After obtaining the output $\tilde{y}$ of the model (MSDC-NET or CNN), use Formula (7) to predict the category of the test sample:

$$label(\tilde{x}) = \arg\max(\tilde{y})$$

Output: Predict the category of input x

4. Experiment and Discussion

4.1. Datasets

4.1.1. PU Bearing Dataset. There are 21 categories, including 6 undamaged bearings, 12 damaged bearings, and 14 actual

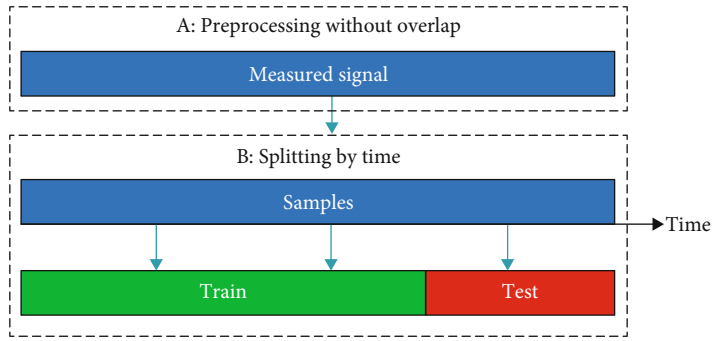


Figure 7: Data split according to time sequences.

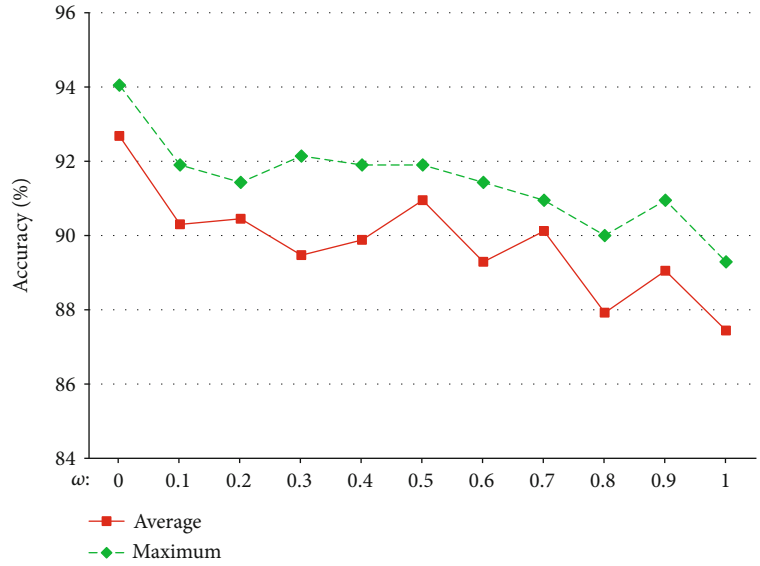
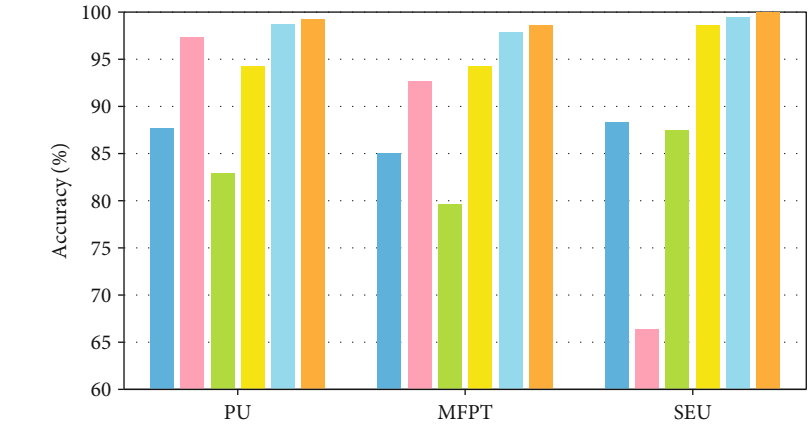
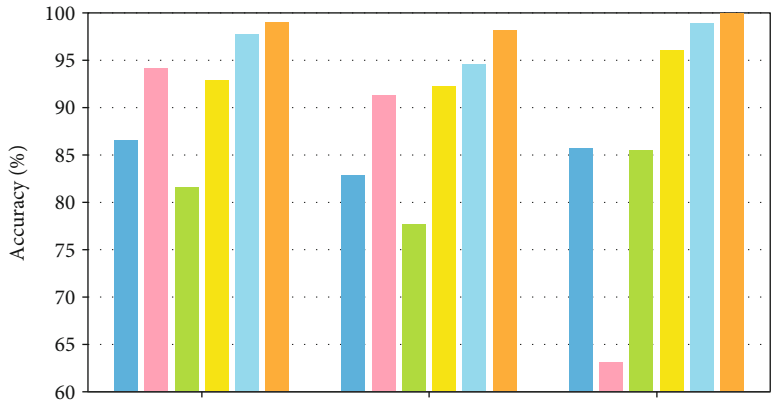


Figure 8: The influence of different ω on the model.



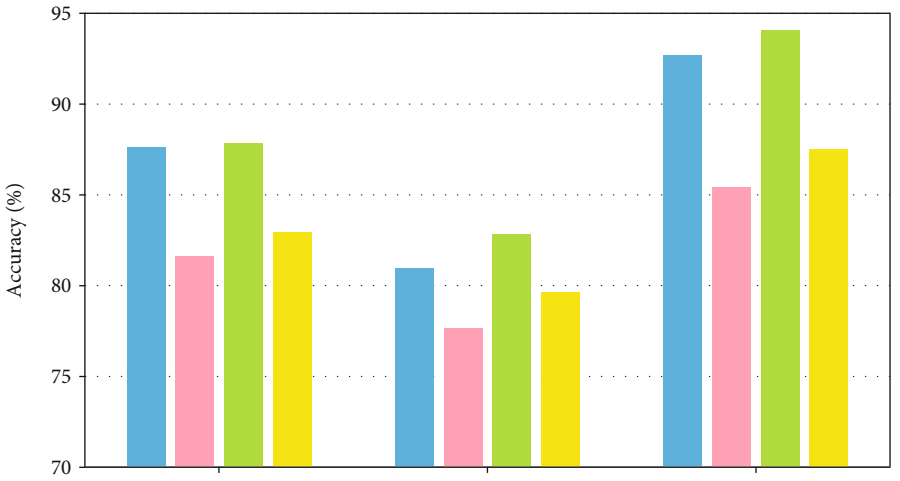
	PU	MFPT	SEU
LSTM	87.71	85.03	88.33
AE	97.38	92.67	66.35
CNN	82.95	79.63	87.5
AlexNet	94.33	94.24	98.58
ResNet18	98.69	97.89	99.52
MSDC	99.23	98.66	100

Figure 9: Maximum prediction accuracy graph of MSDC-NET and five contrast algorithms on three datasets.



	PU	MFPT	SEU
LSTM	86.55	82.87	85.66
AE	94.15	91.33	63.12
CNN	81.62	77.68	85.44
AlexNet	92.91	92.22	96.07
ResNet18	97.73	94.58	98.88
MSDC	99	98.13	100

Figure 10: Average prediction accuracy graph of MSDC-NET and five contrast algorithms on three datasets.



	PU	MFPT	SEU
CNN_D_Avg	87.6	80.97	92.68
CNN_Avg	81.62	77.68	85.44
CNN_D_Max	87.86	82.82	94.05
CNN_Max	82.95	79.63	87.5

Figure 11: Map of average and maximum classification accuracy of original CNN and knowledge distilled CNN_D on three datasets.

damaged bearings caused by an accelerated life test. The details are shown in Table 1.

4.1.2. MFPT Bearing Dataset. There are 15 categories, according to different loads, one healthy bearing and fourteen faulty bearings. The fault classification is displayed in Table 2. The four values in the Description are load, input shaft speed, sampling rate, and duration.

4.1.3. SEU Gearbox Dataset. There are 20 categories, including inner ring, outer ring, and rolling element in different states. In each file, we use the second line of eight vibration signals. Details are shown in Table 3.

4.2. Data Split. There are two common data segmentation methods in mechanical intelligent diagnosis. One is the stochastic segmentation strategy, and the other is the time sort segmentation of datasets. The random segmentation strategy may cause data overlap between two sets, resulting in the risk of test leakage, and the final test accuracy cannot be used as a basis for fair evaluation of model performance. Industrial data are generally continuous, which may contain

time-dependent knowledge information. Therefore, in order to evaluate the scientific and retain the useful information in the dataset as much as possible, we adopt the strategy of dividing the dataset according to the time sort (Figure 7). The last 20% of the time series data is used for testing, and the rest is used for training.

4.3. Input Types. In mechanical intelligent diagnosis, signal processing methods are often used to map sample data to other domains. These methods include converted into time domain (TD), converted into frequency domain (FD), converted into time-frequency domain (TFD), converted into slice image (SI), and converted into wavelet domain (WD). The benchmark study of Zhao et al. [33] showed that FD and TFD can achieve higher accuracy, so we finally used TFD to complete the experiment.

$$x_i^{\mathrm{STFT}} = \mathrm{STFT}(x_i), \quad i = 1, 2, \cdots, N. \tag{11}$$

Inside $STFT(\cdot)$ is to convert x_i into TFD.

TFD is to perform short-time Fourier transform (STFT) on each specimen, and the Hanning window length is 64. Since the CNN model requires a larger 2D size to extract features, the final signal size is adjusted to 330×330.

4.4. Training Details. During model training, we used the Adam optimizer. Some parameters of model training are as follows: learning rate = 0.001 and batch size = 8 (due to the device limitation). Each model is trained and tested alternately during training, and a total of 100 epochs are experienced. All comparison models and MSDC-NET are trained and fairly compared under the open source code framework proposed by Zhao et al. [33], and the data enhancement method provided in the code framework is adopted. All work is performed on the device equipped with E5-2640 V4 @ CPU and 12G GeForce RTX 2080 Ti GPU. The device system is Ubuntu 16.04, the running environment is Python 3.6, and the Pytorch1.4 library is used.

We have done the optimization experiment on the SEU gearbox dataset for the value of ω between 0 and 1. It is able to deduce from Figure 8 that as ω increases, the performance of the CNN model will tend to the performance without knowledge distillation; classification accuracy showed a downward trend, which also shows the effectiveness of knowledge distillation technology from the side, so we set ω to 0 in the subsequent comparative experiments.

4.5. Evaluating Indexes. In our research process, the comprehensive accuracy (Acc) is used to fairly assess the capability of the algorithm. With the purpose of avoiding the impact of the fluctuation of DNN model in training, we repeat each experiment 10 times. Finally, we take the average Acc and the maximum Acc as the assessment indexes, the average Acc is the average of 10 times of Acc, and the maximum Acc is the maximum of 10 times of Acc.

4.6. MSDC-NET Experimental Results. For each algorithm, we completed 10 tests on the corresponding dataset in the same environment. The achievement of the highest predicted Acc is recorded in Figure 9, and then, the achievement of the average predicted Acc of 10 times is recorded in Figure 10. From the overall view of these two figures, our algorithm reached the first-rate capability. The LSTM algorithm does not perform well on classification problems, and the average test Acc is between 85% and 90%.

Compared with those on other datasets, the capability of the AE network on the SEU dataset is significantly not the same, which shows that the AE network has higher requirements for datasets. The classification ability of the simple CNN network is relatively poor without knowledge distillation, ranging from 75 to 85. The test Acc of DNN is relatively high, the test Acc of AlexNet is between 95 and 100, the test Acc of ResNet18 is between 98 and 100, and the test Acc of our MSDC-NET is between 98 and 100. From the comparison chart of average test Acc, our method is 1.27%, 3.55%, and 1.12% higher than ResNet18, which is the best algorithm in comparison. In the maximum test Acc diagram, our method is 0.54%, 0.77%, and 0.48% higher than ResNet18. Meanwhile, it is a surprise that our MSDC-NET has achieved 100% results on SEU gearbox dataset for 10 times, which reflects the powerful classification ability and robustness of our algorithm and will not produce serious over fitting phenomenon due to the deep network layers.

4.7. CNN Experimental Results after Knowledge Distillation. We distill the knowledge of MSDC-NET according to the algorithm in Figure 6, so that the classification Acc of the CNN model is significantly improved without any change in the internal structure. As can be seen in Figure 11, contrasted with the original CNN, the average test Acc of CNN_D after knowledge distillation is 5.98%, 3.29%, and 7.24% higher on the three datasets. The maximum test Acc is 4.91%, 3.19%, and 6.55% higher, respectively. The average test Acc of CNN_D even exceeds the maximum test Acc of the original CNN. CNN_D has the most obvious improvement on the SEU gearbox dataset, which is comparable to the deep neural classification network AlexNet shown in Figure 10. Due to the problems of MFPT dataset itself, the effect of knowledge distillation is not obvious, but it has also been improved.

5. Conclusion

In this paper, we created a new MSDC-NET, which is derived from several typical deep convolutional neural network structures, combined with the most advanced multiscale and jump connection technologies. Experimental results on several datasets show that our MSDC-NET has a more prominent classification level than other latest research. At the same time, we found that knowledge of distillation technology in the diagnosis of intelligent mechanical issue is effective, and the classification accuracy of small CNN model after knowledge distillation is significantly improved. Next, we will further study the knowledge distillation technology, improve the diagnostic accuracy of the CNN model after knowledge distillation, and verify the portability and reliability of the related technology in the direction of mechanical intelligence prediction.

Data Availability

The labeled dataset used to support the findings of this study is available from the corresponding author upon request.

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 61772241 and Grant U20A20228.

References

[1] X. L. Zhang, B. J. Wang, and X. F. Chen, "Intelligent fault diagnosis of roller bearings with multivariable ensemble-based incremental support vector machine," *Knowledge-Based Systems*, vol. 89, pp. 56–85, 2015.

[2] T. P. Banerjee and S. Das, "Multi-sensor data fusion using support vector machine for motor fault detection," *Information Sciences*, vol. 217, pp. 96–107, 2012.

[3] H. al-Bugharbee and I. Trendafilova, "A fault diagnosis methodology for rolling element bearings based on advanced signal pretreatment and autoregressive modelling," *Journal of Sound and Vibration*, vol. 369, pp. 246–265, 2016.

[4] H. Keskes, A. Braham, and Z. Lachiri, "Broken rotor bar diagnosis in induction machines through stationary wavelet packet transform and multiclass wavelet SVM," *Electric Power Systems Research*, vol. 97, pp. 151–157, 2013.

[5] M. O. Mustafa, D. Varagnolo, G. Nikolakopoulos, and T. Gustafsson, "Detecting broken rotor bars in induction motors with model-based support vector classifiers," *Control Engineering Practice*, vol. 52, pp. 15–23, 2016.

[6] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, *Learning Internal Representations by Error Propagation*, California Univ San Diego La Jolla Inst for Cognitive Science, 1985.

[7] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[8] Y. LeCun and Y. Bengio, "Convolutional networks for images, speech, and time series," *The handbook of brain theory and neural networks*, vol. 3361, p. 1995, 1995.

[9] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," 2014, http://arxiv.org/abs/1312.6114.

[10] W. Zaremba, I. Sutskever, and O. Vinyals, "Recurrent neural network regularization," 2014, http://arxiv.org/abs/1409.2329.

[11] H. D. Shao, H. K. Jiang, X. Zhang, and M. G. Niu, "Rolling bearing fault diagnosis using an optimization deep belief network," *Measurement Science and Technology*, vol. 26, no. 11, p. 115002, 2015.

[12] P. Tamilselvan and P. F. Wang, "Failure diagnosis using deep belief learning based health state classification," *Reliability Engineering and System Safety*, vol. 115, pp. 124–135, 2013.

[13] J. Zhang, Y. Sun, L. Guo, H. Gao, X. Hong, and H. Song, "A new bearing fault diagnosis method based on modified convolutional neural networks," *Chinese Journal of Aeronautics*, vol. 33, no. 2, pp. 439–447, 2020.

[14] Y. Li, X. Du, F. Wan, X. Wang, and H. Yu, "Rotating machinery fault diagnosis based on convolutional neural network and infrared thermal imaging," *Chinese Journal of Aeronautics*, vol. 33, no. 2, pp. 427–438, 2020.

[15] A. Youcef Khodja, N. Guersi, M. N. Saadi, and N. Boutasseta, "Rolling element bearing fault diagnosis for rotating machinery using vibration spectrum imaging and convolutional neural networks," *The International Journal of Advanced Manufacturing Technology*, vol. 106, no. 5-6, pp. 1737–1751, 2020.

[16] Y. Zhang, K. Xing, R. Bai, D. Sun, and Z. Meng, "An enhanced convolutional neural network for bearing fault diagnosis based on time-frequency image," *Measurement*, vol. 157, p. 107667, 2020.

[17] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza et al., "Generative adversarial networks," *Advances in Neural Information Processing Systems*, vol. 3, pp. 2672–2680, 2014.

[18] D. B. Verstraete, E. L. Droguett, V. Meruane, M. Modarres, and A. Ferrada, "Deep semi-supervised generative adversarial fault diagnostics of rolling element bearings," *Structural Health Monitoring*, vol. 19, no. 2, pp. 390–411, 2020.

[19] W. Zhang, X. Li, X.-D. Jia, H. Ma, Z. Luo, and X. Li, "Machinery fault diagnosis with imbalanced data using deep generative adversarial networks," *Measurement*, vol. 152, article 107377, 2020.

[20] O. Janssens, V. Slavkovikj, B. Vervisch et al., "Convolutional neural network based fault detection for rotating machinery," *Journal of Sound and Vibration*, vol. 377, pp. 331–345, 2016.

[21] Y. Jiang, X. Gu, D. Wu et al., "A novel negative-transfer-resistant fuzzy clustering model with a shared cross-domain transfer latent space and its application to brain CT image segmentation," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 18, no. 1, pp. 1–52, 2020.

[22] Y. Jiang, K. Zhao, K. Xia et al., "A novel distributed multitask fuzzy clustering algorithm for automatic MR brain image segmentation," *Journal of Medical Systems*, vol. 43, no. 5, 2019.

[23] G. Xu, M. Liu, Z. Jiang, W. Shen, and C. Huang, "Online fault diagnosis method based on transfer convolutional neural networks," *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 2, pp. 509–520, 2020.

[24] W. Mao, L. Ding, S. Tian, and X. Liang, "Online detection for bearing incipient fault based on deep transfer learning," *Measurement*, vol. 152, article 107278, 2020.

[25] Z. Chen, K. Gryllias, and W. Li, "Intelligent fault diagnosis for rotary machinery using transferable convolutional neural network," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 1, pp. 339–349, 2020.

[26] Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proceedings of the 25th international conference on Machine learning - ICML '08*, pp. 1096–1103, Helsinki, Finland, 2008.

[27] M. Ranzato, C. Poultney, S. Chopra, and Y. L. Cun, "Efficient learning of sparse representations with an energy-based model," *Advances in Neural Information Processing Systems*, vol. 19, pp. 1137–1144, 2007.

[28] X. Xiong, J. Hongkai, X. Li, and M. Niu, "A Wasserstein gradient-penalty generative adversarial network with deep auto-encoder for bearing intelligent fault diagnosis," *Measurement Science and Technology*, vol. 31, no. 4, p. 045006, 2020.

[29] 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, p. 104837, 2020.

[30] 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, no. 3, pp. 2044–2053, 2020.

[31] N. Jiang, X. Hu, and N. Li, "Graphical temporal semi-supervised deep learning-based principal fault localization in wind turbine systems," *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, vol. 234, no. 9, pp. 985–999, 2020.

[32] X. Li, J. Li, Y. Qu, and D. He, "Semi-supervised gear fault diagnosis using raw vibration signal based on deep learning," *Chinese Journal of Aeronautics*, vol. 33, no. 2, pp. 418–426, 2020.

[33] Z. Zhao, T. Li, J. Wu et al., "Deep learning algorithms for rotating machinery intelligent diagnosis: an open source benchmark study," *ISA Transactions*, vol. 107, pp. 224–255, 2020.

[34] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, pp. 1097–1105, 2012.

[35] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, Las Vegas, NV, USA, 2016.

[36] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[37] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *Computer Science*, vol. 14, no. 7, pp. 38-39, 2015.

[38] G. Huang, Z. Liu, V. Laurens, and K. Q. Weinberger, "Densely connected convolutional networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–9, Honolulu, HI, USA, 2017.

[39] C. Szegedy, W. Liu, Y. Jia et al., "Going deeper with convolutions," in *2015 IEEE conference on computer vision and pattern recognition (CVPR)*, Boston, MA, USA, 2015.