IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 26, 2019, accepted April 16, 2019, date of publication April 18, 2019, date of current version May 13, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2912072

Lightweight Deep Residual CNN for Fault Diagnosis of Rotating Machinery Based on Depthwise Separable Convolutions

SHANGJUN MA[1], WENKAI LIU[2], WEI CAI[1], ZHAOWEI SHANG[2], AND GENG LIU[1]

[1]Shaanxi Engineering Laboratory for Transmissions and Controls, Northwestern Polytechnical University, Xi'an 710072, China
[2]Key Laboratory of Dependable Service Computing in Cyber Physical Society Chongqing University, Chongqing 400044, China

Corresponding author: Shangjun Ma (mashangjun@nwpu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 51875458, and in part by the 111 Project under Grant B13044.

ABSTRACT This paper proposes an efficient and noise-insensitive end-to-end lightweight deep learning method. The method synthesizes the characteristics of a frequency domain transform and a deep convolutional neural network. The former can extract multiscale information in vibration signal processing and the latter has a good classification performance, data-driven, and high transfer-learning ability. A vibration signal is decomposed into a pyramidal wavelet packet, and each sub-band coefficient is used as an input of a channel in the deep network. A deep residual convolutional network based on a separable convolution and concatenated rectified linear unit (CReLU) lightweight convolution technology is used for fault diagnosis. The proposed algorithm is compared with related deep learning algorithms using two bearing datasets produced by Case Western Reserve University (CWRU) and the Center for Intelligent Maintenance Systems (IMS), University of Cincinnati. Compared with the existing algorithms, the experimental results show that the comprehensive performance of the algorithm proposed in this paper is ''small, light, and fast,'' and satisfactory diagnostic results are obtained in the fault diagnosis of rotating machinery.

INDEX TERMS Residual convolutional neural networks, depthwise separable convolutions, deep learning, fault diagnosis, wavelet packet transform.

I. INTRODUCTION

Rotating machinery systems have been widely used in various kinds mechanical equipment and play an increasingly important role. Since the 1970s, fault diagnosis, fault prediction, condition-based maintenance and health management have been gradually applied in engineering. These methods reduce the effect of damage or failure of rotating machinery on the reliability and safety of the entire mechanical system, thereby reducing economic losses. At present, health management, fault diagnosis and prediction of large mechanical systems are well-studied and challenging problems in theoretical research and engineering practice. Only through fault detection, isolation and repair can the normal operation of a mechanical system be ensured. Therefore, fault prediction technology is key for mechanical systems.

The associate editor coordinating the review of this manuscript and approving it for publication was Yan-Jun Liu.

Relying on fault data, research on data-driven artificial intelligence methods is key to realizing fault diagnosis. Traditional methods mainly include artificial neural networks (ANNs), support vector machines (SVMs), logical regression, hidden Markov models (HMMs) and fuzzy decisions [1]–[3]. Traditional artificial intelligence methods are based on feature data, but the feature extraction of fault data must be artificially designed based on the characteristics of different faults rather than automatic extraction. At the same time, the performance of a model or algorithm is directly determined by the quality of the feature. Therefore, the accuracy of fault diagnosis depends on the professional ability of the user. In addition, traditional artificial intelligence methods are shallow leaning models, with which it is difficult to effectively learn the nonlinear relationships of complex systems [4]–[6].

To improve the prediction performance, in recent years, deep learning has been gradually applied to fault diagnosis of

mechanical signals. Jia *et al*. [7] pretrained a three-layer deep neural network by stacking a self-encoder and then fine-tuned the network to obtain the final prediction results for bearing and planetary gearbox fault diagnosis. Li *et al*. [8] proposed a deep random forest fusion (DRFF) structure. A wavelet transform and a deep Boltzmann machine were used to extract signal features, and deep random forest fusion was used for gearbox fault diagnosis. Gan *et al*. [9] employed a characterization based on wavelet packet energy of a raw signal, and a hierarchical diagnosis network based on a deep confidence network was used for bearing fault diagnosis. The first layer was used to diagnose the fault type, and the second layer was developed to further recognize the fault severity ranking from the result of the first layer. Sun *et al*. [10] proposed a sparse deep stacking network and used sparse regularization to optimize the network. The above methods mainly use a self-encoder, deep Boltzmann machine and deep confidence network. These methods are relatively easy to implement and can learn feature representation but exhibit slow convergence and a weak migration learning ability.

Yuan *et al*. [11] investigated three recurrent neural network (RNN) models, including simple RNN, long short-term memory (LSTM) and gated recurrent units (GRU) LSTM, for fault diagnosis and prognostics of an aeroengine. Zhang *et al*. [12] also used similar structures to predict the residual life of lithium-ion batteries. Zhao *et al*. [13] proposed an LSTM-based method for a machine health monitoring system. Park *et al*. [14] developed a fault detection model for an industrial robot manipulator based on LSTM recurrent neural networks. Lu *et al*. [15] proposed a novel deep neural network model with domain adaptation for the fault diagnosis of a bearing. Zhao *et al*. [16] presented a deep network based on convolutional neural network (CNN) and bi-directional LSTM (BiLSTM) to predict the wear of a milling machine cutter. The above research shows that cyclic neural networks and hybrid structures have good performance in terms of time series data detection and can find problems caused over time, but there are difficulties in the training process and implementation, the network structure is relatively complex, and the transfer-learning ability is weak.

At present, the focus of fault diagnosis research is deep CNNs. According to the dimension of the processing object, a CNN can be divided into one-dimensional and two-dimensional signal modes. According to the processing object, the CNN can also be divided into time domain and frequency domain modes. In a one-dimensional time domain, Abdeljaber *et al*. [17] constructed a 1D-LeNet5 network based on LeNet5 for damage detection of mechanical structures. Although local features were extracted effectively, the network structure was complex and required a great deal of time and computational resources in the training and prediction process. Based on the 1D-CNN method, deep convolutional network models with different structures were proposed in references [18]–[21] for fault prediction of different rotating machinery. Among these models, reference [21] adopted a CNN to process a raw signal directly, and with the help of the smoothing effect of convolution, the length of the first-layer convolution filter was set to 64 to improve the noise resistance performance. In a two-dimensional time domain, Wen *et al*. [22] transformed a raw signal into a square matrix through nonoverlapping cutting and normalized the value to 0-255, which was regarded as an image directly using 2D-Lenet-5 for fault prediction of gears and bearings. Reference [23] cut the raw signal into a square matrix by changing the interval K, which was directly used for fault detection in the 2D-CNN structure training. However, the raw time domain signal was used as input, which ignored the frequency domain characteristics of the signal, and the full-connection technology led to a large memory occupation.

To better reveal the characteristics of fault information, many time-frequency analysis methods [24], such as the short-time Fourier transform (STFT), empirical mode decomposition (EMD), continuous wavelet transform (CWT), wavelet packet transform (WPT) and dual tree complex wavelet transform (DTCWT) have been combined with deep learning to detect and diagnose the faults [25]–[29]. These methods transformed 1D vibration signals to 2D representations by time-frequency analysis and utilized deep learning methods to extract discriminative features from the time-frequency representations instead of from the time or frequency domain. Compared with the STFT and EMD, DTCWT, CWT and WPT have the characteristics of multiresolution analysis and a solid theoretical basis. Therefore, to obtain more high-frequency information and facilitate subsequent signal processing, a wavelet packet was selected as the input of the deep learning model for fault diagnosis. For instance, in reference [27], a novel diagnosis method was proposed involving the use of a CNN to directly classify a continuous wavelet transform scalogram (CWTS), which is a time-frequency domain transform of the raw signal and can contain most of the information of vibration signals. Sun *et al*. [28] used multiscale information extracted by DTCWT to form a matrix and combined a CNN for gear fault diagnosis. Zhao *et al*. [29] constituted each subband obtained by a wavelet packet transform into a square matrix and combined it with a residual network for fault prediction of planetary gearboxes.

The above methods perform well for multidimensional fault data and can effectively extract local features, but the network structures are relatively complex, and the computational complexity is high, which requires many computational resources in the training and prediction processes. Compared with a one-dimensional processing method, the two-dimensional structure is relatively complex, which requires more time and computational resources in the training and testing processes. For example, the amount of computation and memory space occupied by 3×1 convolution is only one third of that of a 3×3 convolution. However, the research and practice of deep convolutional networks in fault diagnosis have mostly focused on improving performance. Although the algorithms work well, with the rapid development and maturity of industrial Internet of Things

technology, they cannot meet the requirements of being ''small, light and fast'' for deep learning algorithms in fault diagnosis [2], [3]. The main reasons are as follows:

(1) The existing fault prediction algorithms based on deep learning require more computational complexity and model space, which cannot meet the needs of low cost and high real-time performance and directly restrict the application of related algorithms.

(2) With the diversification and complexity of the working environment, the probability of an algorithm being susceptible to various disturbances is increasing, which puts forward new requirements for the robustness of the algorithm.

(3) Complex and changing working conditions (variable speed and load) make the acquisition data unstable. With the change of working conditions and operating time, the sample distribution no longer meets the same distribution requirements, so a demand for strong migration learning ability emerges. Although existing methods have performed some research on portability, robustness and transfer learning ability, it is difficult for the existing methods to meet the actual application demands. For example, the number of parameters and floating-point computations of reference [22] were 50 MegaByte (MB) and 1.45×10^8, respectively. Similarly, the size of the parameter data and number of floating-point computations of reference [23] were 565.16 KiloByte (KB) and 8.08×10^7, respectively.

According to the advantages of signal analysis and lightweight deep learning, a lightweight deep residual CNN method based on depthwise separable convolutions is proposed. The contributions of this work are the following.

(1) A lightweight one-dimensional deep residual convolutional network structure is proposed, which can effectively improve the recognition accuracy and ensure a fast calculation speed and a small parameter space. (2) The proposed network structure has a strong migration learning ability and noise resistance performance.

The remainder of this paper is organized as follows: Section 2 provides a brief review on a theoretical basis. Then, we propose the lightweight deep CNN network structure based on a wavelet packet in Section 3, and a case study on fault diagnosis of a bearing, an algorithm comparison and a discussion are given in Section 4. Finally, the conclusion is summarized in Section 5.

II. THEORETICAL BASIS

A. WAVELET PACKET TRANSFORM (WPT)

A wavelet packet function is a time-frequency function, which can be described as follows [27]:

$$w_{i,j}^{n} = \sqrt{2} w^{n}(2^{j}t - k) \tag{1}$$

where j and k are integers and indices of scale and translation operations. The index n is an operation modulation parameter or oscillation parameter. The first two wavelet packet functions are the scaling and mother wavelet functions:

$$w_{i,j}^{0}(t) = \varphi(t) \tag{2}$$

$$w_{i,j}^{1}(t) = \psi(t) \tag{3}$$

When $n = 2, 3, \ldots$, the function can be defined by the following recursive relationships.

$$w_{0,1}^{2n}(t) = \sqrt{2} \sum_{k} h(k) w_{1,k}^{n}(2t - k) \tag{4}$$

$$w_{0,1}^{2n+1}(t) = \sqrt{2} \sum_{k} g(k) w_{1,k}^{n}(2t - k) \tag{5}$$

where $h(k)$ and $g(k)$ are quadrature mirror filters (QMFs) associated with the predefined scaling function and mother wavelet function, respectively. The $h(k)$ and $g(k)$ filtered signals are referred to the approximation and the detail, respectively.

The WPT can further obtain the detailed wavelet coefficients of a signal at high frequencies and provide a more detailed and comprehensive time-frequency plane tiling than a discrete wavelet transform (DWT). The advantages of the WPT are used in discrete signal processing, such as fault diagnosis of rotating machinery [31], image processing [32] and video processing [33].

B. DEEP CONVOLUTIONAL NETWORKS (DCNs)

1) LIGHTWEIGHT DEEP CNN NETWORK

Because of the large amount of computation involved in a deep CNN, it is difficult to meet the application requirements of embedded systems. In recent years, research on lightweight deep CNNs has made some achievements [34]. At present, the compression methods of deep learning models are mainly divided into four types: parameter pruning and sharing, low rank decomposition, migration/compression convolution filtering and knowledge refinement. In the above methods, parameter pruning and sharing, low rank decomposition, and knowledge refinement are mainly used to lighten the existing network structure, which will affect the anti-noise performance and migration ability of the algorithm. Therefore, this paper chooses the migration/compression convolution filtering method to compress the deep learning model to meet the requirements of embedded systems.

2) STANDARD CONVOLUTION

The convolutional neural network was first proposed by Le Cun in 1989 [35] and has been well applied in the field of computer vision [36], [37]. After a convolution operation, each channel is summed to realize a joint mapping of channel and spatial correlations. The standard convolution operation is shown in Fig. 1.

After a one-dimensional signal is decomposed into a wavelet packet, multiple subbands of different frequency bands are obtained. Each subband is input into the convolutional neural network as a channel. The standard convolutional layer takes as input a $L_{in} \times C_{in}$ feature map $\boldsymbol{F}$ and produces a $L_{out} \times C_{out}$ feature map $\boldsymbol{G}$, where L_{in} and L_{out} are the lengths of the input and output features, and C_{in} and C_{out} are the number of input and output channels. It is parameterized by convolution kernel $\boldsymbol{K}$ of size $L_k \times C_{in} \times C_{out}$,

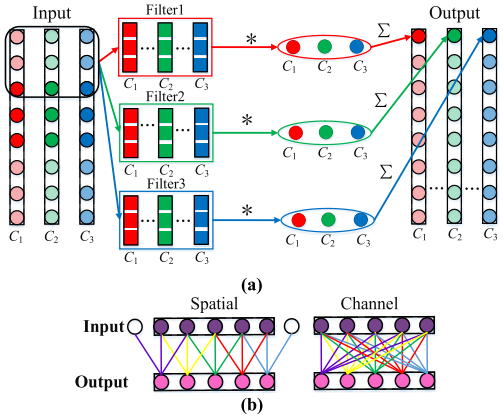

FIGURE 1. The standard convolution operation (a) The operational processes of a standard convolutional layer (b) Spatial and channel correlations of a standard convolutional layer.

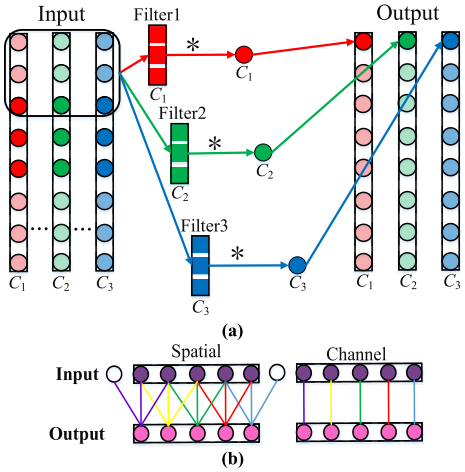

FIGURE 2. Depthwise convolution operation (a) the process of a depthwise convolution (b)spatial and channel correlations of a depthwise convolution.

where L_k is the length of the kernel. The output feature map for standard convolution assuming stride one and padding is computed as:

$$\boldsymbol{G}_{h,n} = \sum_{i,m} \boldsymbol{K}_{i,m,n} \cdot \boldsymbol{F}_{h+i-1,m} \tag{6}$$

The n_{th} filter in $\boldsymbol{K}$ is applied to the all channel in $\boldsymbol{F}$ to produce the n_{th} channel of the filtered output feature map $\boldsymbol{G}$. Output and input are L_k-neighborhood correlations spatially and full correlations in channel, as shown in Fig. 1(b). The parametric and computational cost of standard convolutions can be approximately expressed as:

$$Parameters_{conv} = L_k \cdot C_{in} \cdot C_{out} \tag{7}$$

$$FLOPs_{conv} = 2L_k \cdot C_{in} \cdot C_{out} \cdot L_{out} \tag{8}$$

where $Parameters_{conv}$ denotes the number of parameters of the convolutional layer and $FLOPs_{conv}$ denotes the amount of calculation of the first convolutional layer.

The standard convolutional layer has two main functions: extracting local features of the input data and combining these features linearly to generate new features in one step. The depthwise separable convolution, which factorize a standard convolution into a depthwise convolution and a pointwise convolution [38], splits this into two layers, a separate layer for filtering and a separate layer for combining. This factorization has the effect of drastically reducing computation and model size.

3) DEPTHWISE SEPARABLE CONVOLUTION

The depthwise convolution was first proposed by reference [39], with each convolution filter corresponding to only one input channel, as show in Fig. 2(a). Depthwise convolution with one filter per input channel can be written as:

$$\hat{\boldsymbol{G}}_{h,n} = \sum_{i} \hat{\boldsymbol{K}}_{i,n} \cdot \boldsymbol{F}_{h+i-1,m} \tag{9}$$

where $\hat{\boldsymbol{K}}$ is the depthwise convolutional kernel of size $L_k \times C_{in}$. Similarly, the n_{th} filter in $\hat{\boldsymbol{K}}$ is applied to the n_{th} channel in $\boldsymbol{F}$ to produce the n_{th} channel of the filtered output feature map $\hat{\boldsymbol{G}}$. Output and input are L_k-neighborhood correlations spatially and one-to-one correlations in channel, as shown in Fig. 2(b). Ignoring the effects of bias, the parametric and computational cost of depthwise convolutions can be approximately expressed as:

$$Parameters_{dw} = L_k \cdot C_{in} \tag{10}$$

$$FLOPs_{dw} = 2L_k \cdot C_{in} \cdot L_{out} \tag{11}$$

where $Parameters_{dw}$ and $FLOPs_{dw}$ denote the number of parameters and amount of computation of the depthwise convolution, respectively. Compared with a standard convolution, the number of parameters and amount of computation of the depthwise convolution are $1/C_{out}$ of those of a standard convolution.

Although a depthwise convolution achieves the feature extraction function of the standard convolutional layer, it only extracts the features of each input channel and does not combine to create new features, resulting in information isolation between the channels. To generate new features and increase channel correlation, it is necessary to add a pointwise convolutional layer [38].

A pointwise convolution uses a convolution filter of length 1 to linearly combine the output features of a deep convolution to form new features. As shown in Fig. 3(a), the essence of the pointwise convolution is a special form of the standard convolution filter length of 1. The points in each channel space are weighted and summed to increase the correlation between channels, as shown in Fig. 3(b). Ignoring the effects of bias, the number of parameters and amount of computation can be written as:

$$Parameters_{pw} = 1 \cdot C_{in} \cdot C_{out} \tag{12}$$

$$FLOPs_{pw} = 2 \cdot C_{in} \cdot C_{out} \cdot L_{out} \tag{13}$$

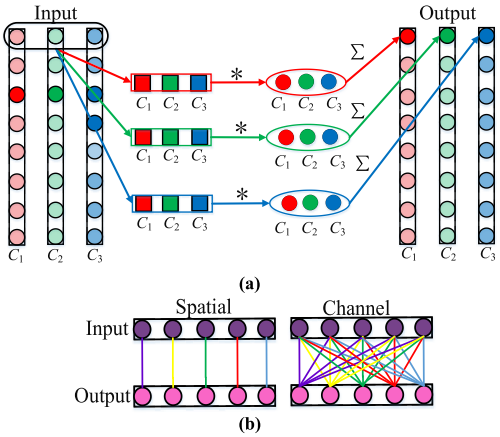

FIGURE 3. Pointwise convolution operation (a) the process of a pointwise convolution (b) spatial and channel correlations of a pointwise convolution.

where $Parameters_{pw}$ and $FLOPs_{pw}$ denote the number of parameters and amount of computation of the pointwise convolution, respectively.

A depthwise separable convolution achieves the effect of a standard convolution of extracting and combining features. It reduces the complexity of the model and almost does not lose accuracy. Compared with a standard convolution, the number of parameters and amount of computation of a deep separable convolution are significantly reduced, and the computation time and storage space can be reduced by half.

$$\frac{L_k \cdot C_{in} + 1 \cdot C_{in}C_{out}}{L_k \cdot C_{in} \cdot C_{out}} = \frac{1}{C_{out}} + \frac{1}{L_k} \tag{14}$$

4) RESIDUAL NETWORK

Convolutional networks can extract and combine data features hierarchically. The extracted data features are more advanced and richer with increasing network layers. The features of deeper network layers are more abstract and represent semantic information. However, the training of deep neural networks does not involve simple stacking. The deeper the network, the easier the problem of gradient explosion and gradient disappearance occurs [40]. Moreover, as the network depth increases, the accuracy of the model decreases [41]. He *et al*. [42] proposed the concept of a residual network that solved the problem of gradient disappearance and successfully applied it to image classification.

As shown in Fig. 4, residual blocks can be implemented by using a shortcut, that is, adding the raw low-level features directly across the multilayer network to the high-level features, which will neither increase the number of parameters nor increase the amount of calculation of the model. Therefore, using a residual structure combined with batch normalization can train a very deep network structure without the problem of gradient disappearance and gradient explosion.

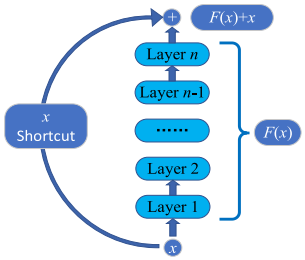

FIGURE 4. The structural diagram of a residual network.

5) BATCH NORMALIZATION

Batch normalization (BN) [43] is a technology widely used in neural networks, which is mainly added after the convolutional layer and the full-connection layer. Before the activation function, the distribution of eigenvalues is normalized (mean value of 0, variance of 1). This not only accelerates the convergence speed of the model but also alleviates the "gradient dispersion" problem of the deep network and makes the training of the deep network model easier and more stable. The detailed calculation process is as follows.

$$\hat{y}_i(j) = \frac{y_i(j) - \mu}{\sigma} \tag{15}$$

$$\hat{z}_i(j) = \gamma \cdot \hat{y}_i(j) + \beta \tag{16}$$

where $\hat{z}_i(j)$ represents an output element after BN, and $\mu = E[y_i(j)]$ and $\sigma^2 = D[y_i(j)]$ represent the mean and variance of the eigenvalues of a layer, respectively. γ and β are the parameters that need to be trained in the networks.

6) CReLU FUNCTION

CReLU is an improved activation function of ReLU. By analyzing the internal structure of a CNN, Shang *et al*. [44] found the following phenomena from a statistical point of view: 1) the parameters of a shallow convolution filter in a convolutional network have a strong negative correlation, and the negative correlation gradually weakens with a deepening of the network layer; 2) a shallow network tends to extract positive and negative phase information; however, the activation function ReLU will erase a negative response resulting in the redundancy of the convolution filter. Therefore, the CReLU function is developed and can be expressed as:

$$\mathrm{ReLU}(x) = x \; if \; x > 0; 0 \; if \; x \leq 0 \tag{17}$$

$$\mathrm{CReLU}(x) = concat[\mathrm{ReLU}(x), \mathrm{ReLU}(-x)] \tag{18}$$

where *concat* operation denotes the connection of two matrices on the channel axis.

CReLU retains the positive and negative outputs of the convolutional layer to form new features and realizes delinearization and increases the number of channels of the feature graph. Compared with the ReLU, the activative channel quantity of CReLU is twice that of the ReLU.

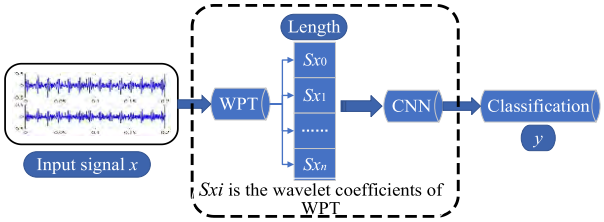

FIGURE 5. Network structure presented in this paper.

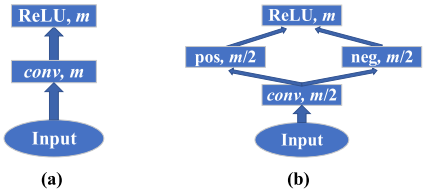

FIGURE 6. The first convolutional layer structure proposed in this paper (a) the general first layer convolution (b) the improved first layer.

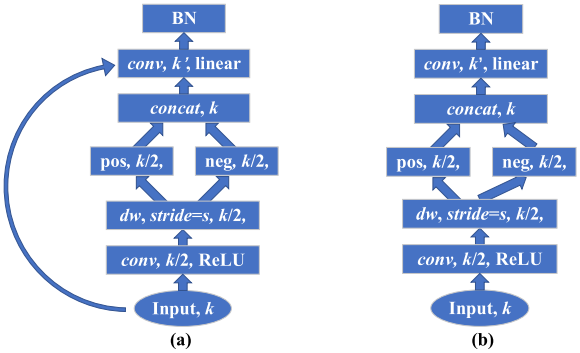

FIGURE 7. Structural design of the basic unit in this paper (a) with shortcut structure and stride = 1 (b) without shortcut structure and stride = 2.

TABLE 1. The specific parameters of the basic unit.

Input	Operator	Output
$h\times k$	*conv*, *ksize*=1, *stride*=1, ReLU	$h\times\frac{k}{2}$
$h\times\frac{k}{2}$	*dw*, *ksize*=3, *stride*=*s*, CReLU	$\frac{h}{s}\times k$
$\frac{h}{s}\times k$	*conv*, *ksize*=1, *stride*=1, Linear	$\frac{h}{s}\times k'$

7) CROSS-ENTROPY LOSS

Cross-entropy loss is usually used as a loss function in classification tasks. The equation is as follows:

$$Loss(p, q) = -\sum_x p(x) \cdot \log q(x) \tag{19}$$

where $p(x)$ is a label for the training set and $q(x)$ is the label value predicted by the network.

In classification problems, the cross-entropy function is often used as a loss function, because the gradient of the cross-entropy loss is only related to correct classification prediction results in the model optimization process. In this paper, the cross-entropy function is chosen as the loss function of the training model.

III. LIGHTWEIGHT CNN STRUCTURE BASED ON A WAVELET PACKET

A. NETWORK STRUCTURE

In this paper, an end-to-end network structure is proposed, which is mainly for one-dimensional fault signal classification. The flow chart is shown in Fig. 5. This network can be divided into two steps according to its function. The first step performs the wavelet packet transform (WPT), with the aim of extracting finer information from a frequency-domain perspective, and the second step is a one-dimensional lightweight CNN. The network has the following characteristics:

(1) The activation function of the first standard convolutional layer uses the CReLU function, which can reduce the number of parameters and the amount of calculation by half compared to the ReLU function, as shown in Fig. 6.

(2) A lightweight basic unit module is designed based on the CReLU function, as shown in Fig. 7.

(3) The complexity of the network structure can be adjusted according to the data characteristics by using super parameters m and i, as shown in Table 2.

B. DEEP CONVOLUTIONAL NEURAL NETWORK AND ITS IMPROVEMENT

1) IMPROVEMENT OF A STANDARD CONVOLUTIONAL LAYER

To extract the feature information of raw data effectively, a standard convolution is still used in the first layer of the lightweight network based on a separable convolutional design [45], [46]. Fig. 6 (a) shows the general first layer of the convolution, and Fig. 6 (b) is an improved structure using the CReLU function in this paper. As shown in Fig. 6, The activation feature maps of m channels can be obtained by ReLU activation function. But for CRelu activation function, a total of $2m$ feature maps can be obtained, which include the positive feature maps of m channels and the negative feature maps of m channels. In other words, in order to obtain the activation feature map of m channels, only input feature maps of $m/2$ channels into the CReLu layer. Then, the upper convolution layer only needs to extract the feature map of $m/2$ channels. Compared with extracting the feature maps of m channels, the number of convolution cores can be reduced by half, so CReLU function can reduce the number of parameters and the amount of calculation by half.

2) BASIC UNIT

The network structure presented in this paper is formed by stacking several basic units. The basic unit is a residual module, which includes a pointwise convolutional layer, a depthwise convolutional layer, a CReLU activation layer, a BN layer and an identity shortcut. The specific parameters and structural design of the basic unit are shown in Table 1 and Fig. 7.

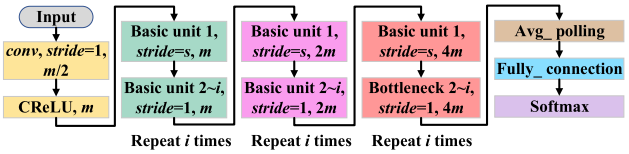

FIGURE 8. LDR-CNN network structure design.

In Table 1, h is the length of the input signal, k is the number of channels of the input, k' is the number of channels of the output, *conv* denotes a standard convolution operation, *dw* denotes a depthwise convolution operation, *ksize* denotes the length of the convolution kernel, *stride* denotes the sliding step, and s is a parameter of the sliding step which represents the separation distance between two convolution fields and and the size of the output feature map is $1/s$ of the original.

To reduce the loss of information in the process of delinearization, reference [46] first used a 1×1 convolution kernel to enhance the dimension and then ReLU activation in the basic unit. As shown in Eq. (18), CReLU keeps the negative response information of the convolutional layer in the negative part without losing information. Even in low dimensions, CReLU can achieve a good delinearization effect. Therefore, only a 1×1 convolution kernel is needed to reconstruct new features.

3) CNN STRUCTURE

Zeriler and Fergus [47] used deconvolution techniques to visualize the characteristics of convolutional neural networks and gained insight into the many characteristics of convolutional networks. It was found that the output from channels of each convolutional layer represent multiple features of the input. Theory and experiments have shown that the width and depth of neural networks are two core factors that characterize the network complexity, and neural networks are often overparameterized and overcomplicated [48].

The numbers of channels and network layers determine the width and depth of a network, respectively. The width determines the number of features extracted from the input data. The depth determines the abstraction degree of the extracted features. Compared to the image, the one-dimensional signal does not have direction information, so the number of features is relatively small and does not require too many channels.

To make the complexity of the network conform to the complexity of the data and reduce the network redundancy, two hyperparameters i and m are set to represent the depth and width of the network, respectively, to control the complexity of the network. The impact of these two parameters on network performance was verified in subsequent experiments.

Based on the above theory, a lightweight deep residual CNN (LDR-CNN) is proposed in this section. The network structure is shown in Fig. 8, and the design parameters of the network structure are shown in Table 2.

In Fig. 8, *conv* denotes the structure of Fig. 6 (b), and basic unit denotes the structure of Fig. 7. Only the sliding step of the basic unit in the first layer is equal to s, and the sliding steps of the remaining n-1 layers are equal to 1. In the network, two parameters m and i are designed to represent the number of channels and repetitions of the basic unit at a certain scale. The effects of these two parameters on the performance will be verified in subsequent experiments.

TABLE 2. The design parameters of the network structure.

Input	Operator	Output channels	repetitions	*stride*
64×16	*conv*	m	1	1
64×m	Basic Unit	m	i	2
32×2m	Basic Unit	2m	i	2
16×4m	Basic Unit	4m	i	2
8×4m	Average pooling			
1×4m	Fully connection	k		

TABLE 3. Comparison of networks in terms of parameters and computational load.

Method	Number of parameters	Floating-point computations
LDR-CNN	16.602KB	1.18×10^5
Reference [29]	72.35KB	1.40×10^6
Reference [20]	203.80KB	6.81×10^7
Reference [21]	258.52KB	4.69×10^6
Reference [23]	565.16KB	8.08×10^7
Reference [27]	367.814KB	2.00×10^8
Reference [7]	2696.54KB	1.37×10^6
Reference [22]	50371.07KB	1.45×10^8
Reference [28]	206.32KB	1.01×10^7
Reference [13]	1028.00KB	4.20×10^6
Resnet-50	80849.75KB	2.04×10^9
Resnet-18	14325.75KB	3.35×10^8
VGG-16	78544.75KB	9.24×10^8
1D-LeNet5	1349Kb	1.27×10^6

C. PERFORMANCE ANALYSIS OF THE NETWORK PARAMETERS

Various network structures, including LDR-CNN, DNN, 1D-LeNet5, Resnet-18, Resnet-50 and VGG-16, are compared in terms of the number of parameters and the floating-point computations, as shown in Table 3.

DNN is the deep neural network proposed in reference [7] for machinery fault diagnosis (contains three hidden layers), and each hidden layer contains 600, 200, and 100 neurons. 1D LeNet5 is a 1D CNN that is obtained by improving LeNet-5 and used for motor fault detection. Resnet-18, Resnet-50 and VGG-16 are commonly used CNN frameworks. To accommodate 1D input data, 2D convolutions in these three networks were altered to 1D convolutions for the calculation.

The LDR-CNN proposed in this paper has no parameters to be trained in the wavelet packet network layer, and the floating-point computation can be neglected, so only the data of the other layer can be calculated. In the calculation process, the input signals of all networks are unified as mechanical vibration signals with a length of 1024×1.

TABLE 4. Classification of the bearing fault datasets.

Datasets	Load (HP)	Training samples	Test samples	Fault types	Flaw size (inches)	Models
A/B/C/D	0/1/2/3	800/800/800/800	100/100/100/100	normal	0	1
		800/800/800/800	100/100/100/100	ball	0.007	2
		800/800/800/800	100/100/100/100	ball	0.014	3
		800/800/800/800	100/100/100/100	ball	0.021	4
		800/800/800/800	100/100/100/100	inner_race	0.007	5
		800/800/800/800	100/100/100/100	inner_race	0.014	6
		800/800/800/800	100/100/100/100	inner_race	0.021	7
		800/800/800/800	100/100/100/100	outer_race	0.007	8
		800/800/800/800	100/100/100/100	outer_race	0.014	9
		800/800/800/800	100/100/100/100	outer_race	0.021	10

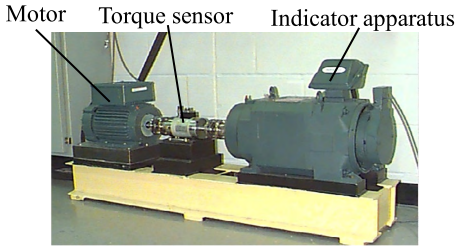

FIGURE 9. Fault simulation test rig.

The results show that the LDR-CNN structure is at least three orders of magnitude less than Resnet-18, Resnet-50 and VGG-16 in terms of floating-point computations, and the number of parameters of LDR-CNN is approximately one thousandth of that of Resnet-18, Resnet-50 and VGG-16. The computational loads of 1D-LeNet5, DNN, reference [29] and reference [20] are 10.7, 11.6, 11.8 and 577 times that of LDR-CNN, respectively. The number of parameters of LDR-CNN are only 11% of that of 1D-LeNet5, 0.6% of that of DNN, 23% of that of reference [29] and 8% of that of reference [20]. Moreover, the proposed hybrid network has at least 18 convolutional layers, which are much larger than 1D-LeNet5 and DNN in network depth, and has a stronger ability to extract features, which can be seen in subsequent experiments.

IV. EXPERIMENT AND ANALYSIS

A. INTRODUCTION OF THE DATASET

To validate the effectiveness of the proposed LDR-CNN structure, 12-kHz drive-end data collected by Case Western Reserve University (CWRU)'s Bearing Data Center were used. Fig. 9 shows the test rig used for data collection [49]. The dataset contains four different categories, namely, normal bearings, bearings with a faulty ball (ball), bearings with a faulty inner race (inner_race) and bearings with a faulty outer race (outer_race). For each type of fault, there are three fault diameters, namely, 0.007 in, 0.014 in and 0.021 in. Thus, there are 10 classifications in this dataset.

Due to the limited experimental data, the overlapping sampling method is used to enhance the data according to references [7] and [26], as shown in Fig. 10.

FIGURE 10. Schematic diagram of overlapping sampling.

In this study, the overlap length was determined based on the data length. If a raw signal had a length of 240,000, then the shift was set to 200, and the signal was segmented into $(240,000-1,024)/200 = 1,194$ samples, of which 800 were randomly selected as training samples and 100 were used as test samples. If a raw signal had a length of 120,000, it was segmented into $(120,000-1,024)/100z = 1,189$ samples. In the experiments, the dataset was divided into eight training sets, one validation set, and one test set. In addition, 800 training samples, 100 test samples and 100 validation samples were randomly selected. Then, in the training process, the validation set was used to examine the identification accuracy after every 10 epochs. Finally, the model with the highest accuracy was preserved. This method is more accurate than directly fixing the number of iterations. Thus, all the data required for the experiments were generated with various datasets. The details are given in the following.

The dataset is divided into four subdatasets, namely, A, B, C and D, corresponding to the data collected under 0, 1, 2 and 3 loads, respectively. As shown in Table 4, each category of datasets A, B, C and D contains 800 training samples, 100 test samples and 100 validation samples, for a total of 8,000 training samples, 1,000 test samples and 1,000 validation samples.

Under normal circumstances, bearing vibration signals are affected by the surrounding ambient noise. The CWRU dataset selected in this study was collected in an environment with a relatively low level of ambient noise and therefore cannot reflect the performance of the fault diagnosis algorithm in an actual environment. In addition, there are a number of noise sources in an actual environment, and it is impossible to obtain training samples under all the conditions in various noise environments. Therefore, noise was added to the

TABLE 5. Experimental results for the effects of the different number of WPT decompositions on the algorithm performance.

	Validate data	0%	10%	30%	50%	70%	90%	100%
Time domain	100	100	99.92±0.09	98.28±0.26	94.83±0.56	92.21±0.55	89.50±0.83	88.58±0.77
WPT3	100	100	99.99±0.03	99.88±0.11	99.27±0.32	98.38±0.23	96.50±0.61	95.55±0.62
WPT4	100	100	100.00±0.00	99.83±0.08	99.63±0.18	98.91±0.43	97.70±0.41	**97.43±0.43**
WPT5	100	100	99.99±0.03	99.76±0.13	99.41±0.20	98.50±0.20	97.85±0.36	97.17±0.37
WPT6	100	100	99.92±0.09	99.52±0.29	98.90±0.30	98.14±0.41	97.10±0.54	96.80±0.40

TABLE 6. Experimental results for the effects of the different network depths on the noise resistance performance.

i	0%	10%	30%	50%	70%	90%	100%
1	100	99.86±0.12	99.14±0.18	97.96±0.39	96.94±0.33	95.22±0.67	94.42±0.71
2	100	99.99±0.03	99.82±0.10	99.45±0.15	98.88±0.30	97.88±0.41	97.36±0.32
3	100	100.00±0.00	99.83±0.13	99.63±0.21	98.91±0.24	97.70±0.26	97.43±0.36
4	100	100.00±0.00	99.88±0.10	99.52±0.28	99.18±0.29	98.06±0.27	97.60±0.44
5	100	100.00±0.00	99.92±0.06	99.65±0.16	99.43±0.22	98.54±0.44	98.13±0.27
6	100	100.00±0.00	99.82±0.11	99.49±0.22	98.88±0.25	97.79±0.27	97.02±0.30
7	100	99.99±0.03	99.78±0.07	99.43±0.30	98.86±0.40	97.57±0.51	97.28±0.54
8	100	100.00±0.00	99.82±0.12	99.36±0.23	98.58±0.44	97.37±0.41	97.09±0.46
9	100	99.95±0.07	99.57±0.19	99.27±0.16	98.53±0.35	97.35±0.33	97.05±0.39

samples in the raw test set to simulate data from actual conditions. Using the resultant data for testing can produce results closer to those obtained under actual industrial production conditions.

Accordingly, 10%, 30%, 50%, 70%, 90% and 100% white Gaussian noise was added to the data to simulate actual conditions. The signal-to-noise ratio (SNR) used when adding noise is defined as follows.

$$\mathrm{SNR} = 10\log_{10}(\frac{P_{signal}}{P_{noise}}) \tag{20}$$

where P_{signal} and P_{noise} are the intensities of the raw and noise signals, respectively.

To examine the noise resistance of the proposed algorithm, the learning model with the highest accuracy for the validation set in 1,000 iterations was selected. Noise-containing data were randomly generated 10 times for each sample in the training set of dataset A and used for testing and statistically analyzing the experimental results, which were represented in the form of the mean and standard deviation. Similarly, dataset D was used as a training set to train the model, and datasets A, B and C were used to test the accuracy of the algorithm (denoted by DA, DB and DC, respectively) to determine the transfer-learning ability for various loads.

B. DISCUSSION

1) THE EFFECT OF THE DIFFERENT NUMBER OF WPT DECOMPOSITIONS ON THE ALGORITHM PERFORMANCE

The network was trained and tested for a raw time-domain signal and different numbers of WPT decompositions using the parameters $m = 8$ and $i = 3$. The experimental results are shown in Table 5.

As shown in Table 5, under different noise conditions, the mean in the time domain is less than the different number of WPT decompositions, and the variance values are greater. Therefore, the proposed first step wavelet packet decomposition network structure in this paper can improve the noise resistance performance of the model. In the WPT, when the white Gaussian noise is up to 100%, WPT 4 shows a relatively high performance. At layer 4, the length of each subband is 64, and there are 16 subbands. The experimental results show that for a 1D input, a subband length of 64 results in a satisfactory division of the frequency domain, a moderate length and a large amount of information.

2) THE EFFECT OF THE MODEL PARAMETERS ON THE NETWORK PERFORMANCE

Table 5 shows that the number of decompositions of the WPT layer is 4, which has the highest noise resistance. Therefore, to verify the effect of network depth and channel number on the noise resistance performance, WPT 4 and the network structure parameter $m = 8$ are selected in this section. Table 6 and Table 7 summarize the experimental results.

The effects of different network depths on the noise resistance performance are shown in Table 6. As shown in Table 6, the noise resistance performance of the model first increases slightly with increasing depth and then decreases. At the same time, as shown in Fig. 11, the number of parameters and the amount of calculation of the model increase linearly. In this paper, the contribution of the model parameters to the model performance is evaluated by the performance parameter earnings ratio to select the super parameters i and m, which are defined as follows.

$$Ratio = \frac{\Delta acc}{\Delta param} \tag{21}$$

where $Ratio$ denotes the performance parameter earnings ratio, $\Delta\ acc$ denotes the improvement in test accuracy in a noisy environment with 0db SNR, and $\Delta\ param$ denotes the increase in the parameters of the model.

TABLE 7. Experimental results for the effects of different channel numbers on the noise resistance performance.

m	0%	10%	30%	50%	70%	90%	100%
2	100	90.62±0.47	88.36±0.81	86.64±0.77	84.78±0.93	82.28±1.04	81.56±0.46
4	100	98.67±0.24	97.62±0.25	96.47±0.48	95.37±0.78	93.66±0.55	92.71±0.66
6	100	100.00±0.00	99.47±0.27	98.20±0.52	96.55±0.62	94.91±0.32	93.84±0.66
8	100	100.00±0.00	99.83±0.13	99.63±0.21	98.91±0.24	97.70±0.26	97.43±0.36
10	100	100.00±0.00	99.96±0.05	99.74±0.22	99.12±0.22	98.38±0.33	97.56±0.42
12	100	100.00±0.00	99.73±0.16	99.28±0.17	98.52±0.27	97.51±0.42	97.25±0.43
14	100	99.98±0.04	99.85±0.12	99.42±0.26	99.01±0.18	98.57±0.32	98.20±0.30
16	100	100.00±0.00	99.77±0.09	99.31±0.31	98.61±0.24	97.84±0.44	97.18±0.21
18	100	100.00±0.00	99.96±0.07	99.65±0.18	99.56±0.17	98.86±0.24	98.48±0.43
20	100	99.89±0.10	99.63±0.16	99.45±0.22	98.70±0.25	97.91±0.24	97.57±0.27
22	100	99.99±0.03	99.73±0.16	99.28±0.24	98.67±0.31	97.95±0.30	97.71±0.59
24	100	99.99±0.03	99.64±0.15	99.32±0.22	98.60±0.20	98.14±0.32	98.08±0.32
26	100	99.96±0.05	99.65±0.13	99.19±0.21	98.82±0.26	97.64±0.35	97.18±0.36
28	100	100.00±0.00	99.98±0.04	99.81±0.14	99.34±0.22	98.45±0.28	98.22±0.60
30	100	100.00±0.00	99.96±0.09	99.69±0.18	99.33±0.18	98.62±0.33	98.22±0.41
32	100	99.92±0.10	99.79±0.05	99.53±0.08	99.04±0.22	97.80±0.20	97.02±0.46

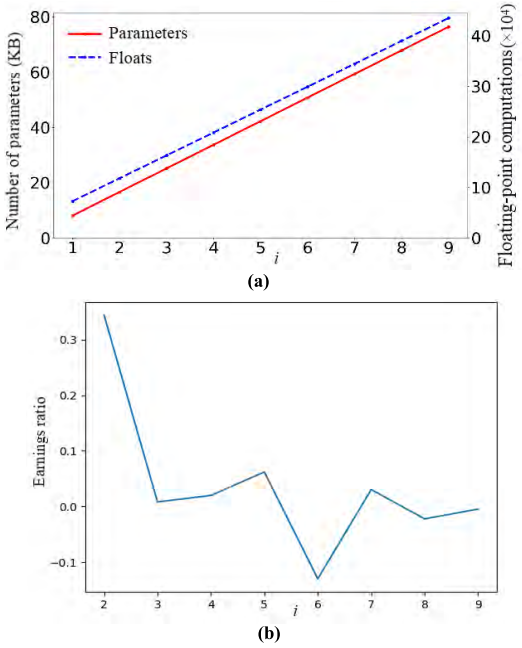

FIGURE 11. The experimental results of different network depths (a) the number of parameters for different network depths (b) the variation trend of the earnings ratio with network depth.

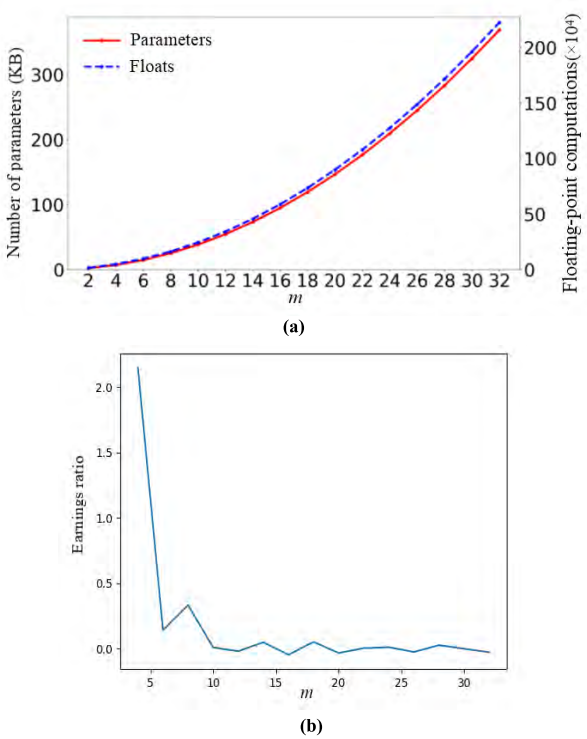

FIGURE 12. The experimental results of different channel numbers (a) the number of parameters under different channel numbers (b) the variation trend of the earnings ratio with channel number.

When the parameter i is increased from 1 to 2, the test accuracy can be increased by 0.34% for every 1 KB increase in the number of parameters. When parameter i is greater than 3, the earnings ratio does not exceed 0.06, and when the network depth continues to increase, the earnings ratio growth is very small. Therefore, to balance performance and resource consumption, $i = 2$ is chosen as the network depth of the model.

The effects of the different channel numbers on the noise resistance performance are shown in Table 7.

As shown in Table 7, with an increase in the proportion of white Gaussian noise, the network performance decreases. Additionally, with an increase in the channel number m, the network performance improves. This is because the number of convolution kernels and the number of feature extractions increases. However, after $m = 8$, the earnings ratio is not higher than 0.05, and the change in the earnings ratio is very small with increasing channel number, as shown in Fig. 12.

3) COMPARISON OF THE NOISE RESISTANCE AND TRANSFER-LEARNING ABILITIES OF VARIOUS ALGORITHMS

To verify the noise resistance and transfer-learning abilities of the proposed algorithm, the proposed algorithm is compared with an SVM and the available deep learning

TABLE 8. Experimental results of various algorithms for various percentages of added white Gaussian noise.

	GPU train Time (s)	CPU test time (s)	0%	10%	30%	50%	70%	90%	100%
LDR-CNN	397	0.123	100	99.99±0.03	99.82±0.10	99.45±0.15	98.88±0.30	97.88±0.41	**97.36±0.32**
Reference [22]	876	0.865	100	32.96±0.60	33.36±0.74	33.36±0.51	33.96±0.58	33.40±0.65	33.36±0.65
Reference [23]	586	0.584	100	99.99±0.03	99.45±0.10	95.58±0.26	84.11±0.84	65.33±1.23	56.01±0.73
Reference [28]	873	0.515	100	96.27±0.62	86.44±0.53	81.35±0.52	73.97±0.87	64.91±0.65	61.20±0.93
Reference [27]	2,992	5.752	100	77.15±0.51	54.88±0.77	45.68±0.93	41.69±0.73	37.99±0.58	36.96±0.85
Reference [29]	848	0.387	100	99.88±0.09	99.44±0.25	98.52±0.31	97.02±0.60	95.33±0.46	94.70±0.48
Reference [20]	674	0.305	100	99.75±0.18	95.80±0.49	91.47±0.44	87.92±0.65	84.44±0.33	83.10±0.43
Reference [31]	37	7.288	82.3	82.51±0.43	77.40±0.48	64.67±1.05	53.73±0.60	47.08±0.88	43.82±0.52
Reference [13]	398	0.066	100	100.00±0.00	99.97±0.05	99.73±0.16	99.37±0.23	98.64±0.41	**98.32±0.26**
Reference [7]	108	0.027	100	99.76±0.13	99.39±0.16	98.77±0.27	97.86±0.35	96.68±0.29	96.14±0.44

TABLE 9. Experimental results of various algorithms for transfer-learning ability.

	AB	AC	AD	BA	BC	BD	CA	CB	CD	DA	DB	DC	AVG
LDR-CNN	**100.0**	**100.0**	**98.8**	**100.0**	**100.0**	**99.2**	97.6	99.0	99.5	**97.5**	**98.7**	**100**	**99.19**
Reference [29]	98.4	99.9	98.2	**99.8**	100	98.9	96.0	98.3	99.3	90.4	93.8	100	97.75
Reference [20]	99.9	98.9	89.8	99.9	99.9	97.9	**98.6**	**99.4**	**99.8**	81.3	85.0	94.2	95.40
Reference [27]	89.3	84.5	73.0	83.6	98.0	91.5	85.1	96.9	95.9	78.0	88.0	95.3	88.26
Reference [23]	98.7	98.3	81.7	98.7	99.9	98.5	93.2	97.7	97.4	89.4	92.0	94.9	95.03
Reference [13]	59.7	64.2	60.2	82.6	88.3	82.2	74.3	89.2	80.7	72.6	83.0	85.7	71.00
Reference [7]	39.3	46.8	45.1	46.0	47.6	57.2	52.4	53.5	55.4	49.8	52.3	52.6	49.83
Reference [31]	40.4	40.3	39.3	40.5	55.3	59.8	41.2	56.3	57.1	36.9	56.1	52.6	47.98
Reference [28]	99.8	99.2	80.0	95.1	100.0	96.5	89.1	95.9	93.7	77.7	79.7	85.6	91.02
Reference [22]	67.3	63.5	61.8	73.0	64.5	64.2	78.3	54.8	66.0	66.0	59.2	69.5	65.67

TABLE 10. Experimental results of the transfer-learning ability of various algorithms for various percentages of added white Gaussian noise.

	DA			DB			DC		
	0%	50%	100%	0%	50%	100%	0%	50%	100%
LDR-CNN	**97.5**	**93.49±0.23**	**87.30±0.92**	**97.9**	**97.07±0.24**	**93.52±0.55**	**100**	**97.70±0.32**	**93.41±0.54**
Reference [29]	90.4	85.29±0.88	79.05±0.74	93.8	92.56±0.50	89.47±0.76	100	93.64±0.47	87.32±0.87
Reference [20]	81.3	71.90±0.79	58.14±0.59	85.0	87.79±0.65	78.10±0.71	94.2	75.52±0.70	67.11±1.10
Reference [27]	78.0	39.30±0.68	31.85±0.68	88.0	45.12±0.94	35.49±0.71	95.3	45.56±0.79	37.54±1.38
Reference [23]	89.4	72.48±0.82	54.54±0.64	92.0	87.42±0.46	67.67±0.40	94.9	88.44±0.69	68.15±0.97
Reference [13]	**77.7**	71.72±0.59	70.25±1.20	78.9	79.61±0.51	78.47±0.85	86.7	84.20±0.73	81.87±0.70
Reference [7]	49.8	47.31±0.53	45.28±0.98	52.3	50.69±0.50	49.34±0.48	52.6	49.89±0.71	47.11±0.97
Reference [31]	36.9	43.88±0.59	26.60±0.86	56.1	50.16±0.61	35.21±0.60	52.6	44.11±0.95	29.35±0.56
Reference [28]	77.7	44.41±0.45	43.34±0.58	79.7	48.57±0.48	48.90±0.88	85.6	59.35±0.67	49.14±0.64
Reference [22]	66.0	32.60±0.70	32.20±0.65	59.2	33.56±1.00	33.44±0.97	69.5	29.92±0.24	29.80±0.27

algorithms [7], [13], [20], [22], [23], [27]–[29] and [31]. The parameters $m = 8$ and $i = 2$ are selected for the experimental model according to the above experimental results. Tables 8, 9 and 10 summarize the experimental results.

As shown in Table 8, the experimental data show that the proposed algorithm exhibits the second highest performance under various percentages of added white Gaussian noise, and the noise resistance performance is 3-58% higher than that of the available algorithms. Although reference [13] outperforms the proposed algorithm with 0.96% of the test accuracy in a 0 dB noise environment, the number of parameters is more than 60 times than the proposed algorithm, as shown in Table 3.

As shown in Table 9, the experimental data indicate that the proposed algorithm exhibits the highest performance for various datasets, and its transfer-learning ability is 1.45-52% higher than that of the available algorithms. The anti-noise ability of reference [13] is the best, but its transfer-learning accuracy is 28% lower than the proposed algorithm.

As shown in Table 10, the experimental data indicate that the proposed algorithm exhibits the highest performance for various percentages of added white Gaussian noise, and its transfer-learning ability is 3.82-64.06% higher than that of the available algorithms.

TABLE 11. Classification of IMS bearing fault datasets.

Training samples	validate samples	Test samples	Fault types	Model
1600	200	200	normal	0
1600	200	200	roller	1
1600	200	200	outer_race	2
1600	200	200	inner_race	3

In order to further verify the performance of the proposed algorithm under complex operating conditions, an additional comparative experiment using test-to-failure data is carried out. This dataset was provided by the Center for Intelligent Maintenance Systems (IMS), University of Cincinnati. The bearing test rig and data description can be obtained from reference [50].

The bearings experienced "increase-decrease-increase" degradation trends. This behavior is due to the "self-healing" nature of the damage [51], [52]. First, the amplitude of

TABLE 12. Experimental results of noise resistance comparison of IMS datasets.

	Train time	Test time	0%	10%	30%	50%	70%	90%	100%
LDR-CNN	**309**	**0.117**	100	100±0.0	99.70 ±0.13	98.87±0.50	97.09±0.66	94.71±0.72	93.89±1.05
Reference [22]	1,006	1.344	100	98.51±0.52	97.04±0.28	94.60±0.09	91.52±0.73	88.66±0.22	87.19±1.47
Reference [23]	615	0.473	100	99.97±0.06	98.43±0.22	95.03±0.65	91.67±0.40	88.48±0.72	87.15±1.07
Reference [28]	723	0.436	100	89.16±1.09	60.66±1.75	38.62±3.06	30.04±1.72	26.85±1.52	26.17±3.46
Reference [27]	2,441	4.657	100	96.96± 0.89	82.49±3.14	73.25± 3.73	66.88±3.59	62.78±3.98	61.12±3.73
Reference [29]	634	0.334	100	99.86±0.16	98.43±0.26	95.56±0.58	92.48±0.43	89.29±0.52	87.29±1.02
Reference [20]	472	0.243	100	99.99±0.02	99.46±0.17	97.19±0.51	94.12±0.69	90.53±0.45	89.12±1.01
Reference [31]	**32**	3.497	76	72.28±1.08	57.76±1.37	52.50±0.61	50.73±0.34	50.51±0.43	50.24±0.21
Reference [13]	367	0.067	99.67	99.34±0.33	97.99±0.66	95.73±1.08	92.47±1.43	88.93±1.86	82.27±1.99
Reference [7]	**84**	**0.030**	91.3	89.42±2.18	85.44±1.90	82.38±2.21	79.22±2.04	76.92±2.05	75.80±2.00

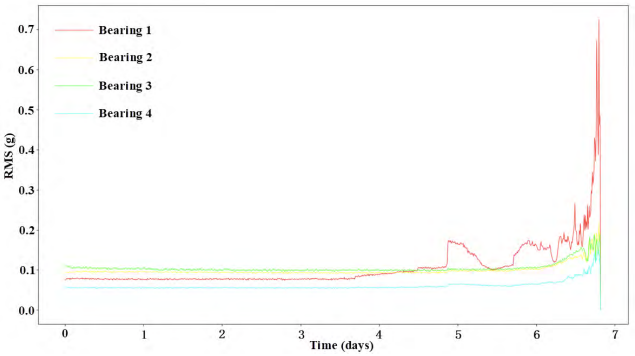

FIGURE 13. The values of root mean square (RMS) in dataset 2 of IMS.

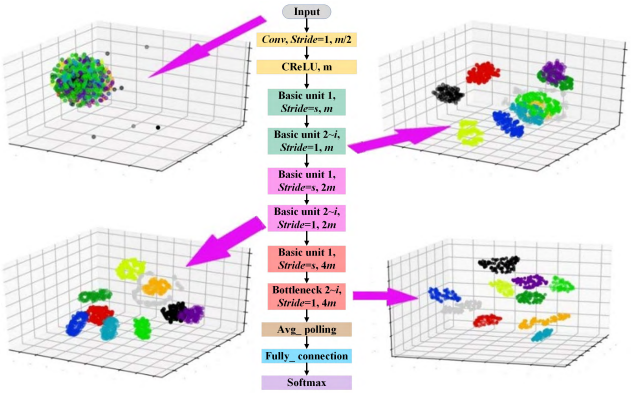

FIGURE 14. The distribution of the output characteristics with the number of network layers.

vibration increases because of the impact caused by the initial surface defect, such as spalling or cracks. Then the initial defect is smoothed by continuous rolling contact leading to the decrease of the impact amplitude. When the damage spreads over a broader area, the vibration amplitude increases again.

During the ''self-healing'' period, the amplitude of the fault bearing is similar to that of the normal bearing, which makes it difficult to detect the fault during the ''self-healing'' period, as shown in Fig. 13. At the end of the experiment, the inner race defect, outer race defect and roller element defect were detected manually [50].

The fault data category is shown in Table 11.

As shown in Fig. 13, the red curve indicates the wear process of the outer race defect in bearing 1. The ''self-healing'' appeared after the failure on the fifth day, and its amplitude was basically the same as that of the normal bearing (green curve).

In order to increase the difficulty of classification, we choose bearings in ''self-healing'' period as fault data, and normal bearings with similar amplitude as normal data. Also, a length of 1024 is directly sampled as a sample instead of overlapping sampling due to relatively enough IMS datasets.

The noise resistance comparison of IMS datasets are shown in Table 12.

As shown in Table 12, compared with references [31] and [7], although the experimental data show that the proposed algorithm exhibits the third highest performance under train time and the second highest performance under test time, the test accuracy of the proposed algorithm is the highest than that of the available algorithms under various percentages of added white Gaussian noise. On the other hand, as shown in Table 3, the amount of calculation of reference [7] is 10 times that of the proposed algorithm. The network structure of reference [7] is only 4 layers and the proposed algorithm is 28 layers, which lead to the waiting time caused by data dependence is 7 times of reference [7].

Besides, because there is more than 30 times acceleration gap between GPU and CPU, which indicates that the parallel computing is one of the main reasons that the actual acceleration effect of experiment is far lower than the theoretical acceleration effect. At the same time, the multi-core CPU used in the experiment has only a small amount of parallel computing power, which also slightly reduces the acceleration effect of the algorithm in the test process. Even so, the computational time and performance of the proposed algorithm are better than the available algorithms.

C. VISUALIZATION OF THE NETWORK LEARNING PROCESS

Because the principle of a convolutional neural network is similar to a black box, the internal working mechanism cannot be explained. Therefore, the output characteristics of the network training process and model testing process can be visualized using the T-SNE method after dimensionality reduction so that the entire network operation process can be easily understood. The visualization of the network learning process is shown in Fig. 14.

From Fig. 14, we can see that the characteristics of all the raw data samples are mixed together and cannot be distinguished. As the number of convolutional layers increases,

the distance between the features of different types becomes larger, while the distance between similar tags decreases, presenting the phenomenon of clustering. In the last layer of the convolutional layer, we can see that the features of the same label are clustered into one group, which shows that the CNN can effectively extract information related to category mapping. With a deepening of the network, the learning ability of the network for features becomes stronger, and the classification accuracy is also higher.

V. CONCLUSION

In this paper, fault prediction of rotating machinery is studied. Based on the premise that a wavelet packet transform can effectively extract frequency domain information and the amount of calculation of a one-dimensional convolution is small, a lightweight deep learning fault prediction method based on a deep residual convolutional network is proposed. The algorithm can effectively improve the recognition accuracy and ensure a fast calculation speed and a small parameter space. The proposed network structure has no parameters to be trained in the wavelet packet transform, and the floating-point computation can be neglected, so only the data of the other layer can be calculated. Moreover, the proposed hybrid network has at least 18 convolutional layers, which are much larger than 1D-LeNet5 and DNN in network depth and has a stronger ability to extract features.

A variety of comparative experiments are carried out by using open CWRU and IMS datasets. The results show that the number of decompositions of the WPT layer is 4, the proposed algorithm has better noise resistance, which results in a satisfactory division of the frequency domain, a moderate length and a large amount of information. To evaluate the contribution of the model parameters to the model performance, the performance parameter earnings ratio is introduced in this paper to select the super parameters. When the network depth and the channel number of the model are 2 and 8, the algorithm proposed in this paper achieves a small amount of computation and a large earnings ratio. Additionally, the experimental data show that the noise resistance performance is 3-58% higher than that of the available algorithms under various percentages of added white Gaussian noise. The transfer-learning ability is 1.45-52% for various datasets and 3.82-64.06% for various percentages of added white Gaussian noise, which has a better transfer-learning ability than the available algorithms.

REFERENCES

[1] Y. Lei, J. Lin, M. J. Zuo, and Z. He, "Condition monitoring and fault diagnosis of planetary gearboxes: A review," *Measurement*, vol. 48, pp. 292–305, Feb. 2014.

[2] W. G. Hatcher and W. Yu, "A survey of deep learning: Platforms, applications and emerging research trends," *IEEE Access*, vol. 6, pp. 24411–24432, 2018.

[3] R. F. Molanes, K. Amarasinghe, J. Rodriguez-Andina, and M. Manic, "Deep learning and reconfigurable platforms in the Internet of Things: Challenges and opportunities in algorithms and hardware," *IEEE Ind. Electron. Mag.*, vol. 12, no. 2, pp. 36–49, Jun. 2018.

[4] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1798–1828, Aug. 2013.

[5] H. Shao, H. Jiang, F. Wang, and H. Zhao, "An enhancement deep feature fusion method for rotating machinery fault diagnosis," *Knowl.-Based Syst.*, vol. 119, pp. 200–220, Mar. 2017.

[6] E. de la Rosa and W. Yu, "Randomized algorithms for nonlinear system identification with deep learning modification," *Inf. Sci.*, vols. 364–365, pp. 197–212, Oct. 2016.

[7] F. Jia, Y. Lei, J. Lin, X. Zhou, and N. Lu, "Deep neural networks: A promising tool for fault characteristic mining and intelligent diagnosis of rotating machinery with massive data," *Mech. Syst. Signal Process.*, vols. 72–73, pp. 303–315, May 2016.

[8] C. Li, R.-V. Sanchez, G. Zurita, M. Cerrada, D. Cabrera, and R. E. Vásquez, "Gearbox fault diagnosis based on deep random forest fusion of acoustic and vibratory signals," *Mech. Syst. Signal Process.*, vols. 76–77, pp. 283–293, Aug. 2016.

[9] M. Gan, C. Wang, and C. Zhu, "Construction of hierarchical diagnosis network based on deep learning and its application in the fault pattern recognition of rolling element bearings," *Mech. Syst. Signal Process.*, vols. 72–73, pp. 92–104, May 2016.

[10] C. Sun, M. Ma, Z. Zhao, and X. Chen, "Sparse deep stacking network for fault diagnosis of motor," *IEEE Trans. Ind. Informat.*, vol. 14, no. 7, pp. 3261–3270, Jul. 2018.

[11] M. Yuan, Y. Wu, and L. Lin, "Fault diagnosis and remaining useful life estimation of aero engine using LSTM neural network," in *Proc. IEEE Int. Conf. Aircr. Utility Syst.*, Beijing, China, Oct. 2016, pp. 135–140.

[12] Y. Zhang, R. Xiong, H. He, and M. G. Pecht, "Long short-term memory recurrent neural network for remaining useful life prediction of lithium-ion batteries," *IEEE Trans. Veh. Technol.*, vol. 67, no. 7, pp. 5695–5705, Jul. 2018.

[13] R. Zhao, J. Wang, R. Yan, and K. Mao, "Machine health monitoring with LSTM networks," in *Proc. 10th Int. Conf. Sens. Technol.*, Nanjing, China, Nov. 2016, pp. 1–6.

[14] D. Park, S. Kim, Y. An, and J.-Y. Jung, "LiReD: A light-weight real-time fault detection system for edge computing using LSTM recurrent neural networks," *Sensors*, vol. 18, no. 7, p. 2110, Jun. 2018.

[15] W. Lu, B. Liang, Y. Cheng, D. Meng, J. Yang, and T. Zhang, "Deep model based domain adaptation for fault diagnosis," *IEEE Trans. Ind. Electron.*, vol. 64, no. 3, pp. 2296–2305, Mar. 2017.

[16] R. Zhao, R. Yan, J. Wang, and K. Mao, "Learning to monitor machine health with convolutional bi-directional LSTM networks," *Sensors*, vol. 17, no. 2, pp. 1–18, Jan. 2017.

[17] O. Abdeljaber, O. Avci, S. Kiranyaz, M. Gabbouj, and D. J. Inman, "Real-time vibration-based structural damage detection using one-dimensional convolutional neural networks," *J. Sound Vib.*, vol. 388, pp. 154–170, Feb. 2017.

[18] O. Janssens *et al.*, "Convolutional neural network based fault detection for rotating machinery," *J. Sound Vib.*, vol. 377, pp. 331–345, Sep. 2016.

[19] T. Ince, S. Kiranyaz, L. Eren, M. Askar, and M. Gabbouj, "Real-time motor fault detection by 1-D convolutional neural networks," *IEEE Trans. Ind. Electron.*, vol. 63, no. 11, pp. 7067–7075, Nov. 2016.

[20] Z. Zhuang and Q. Wei, "Intelligent fault diagnosis of rolling bearing using one-dimensional multi-scale deep convolutional neural network based health state classification," in *Proc. IEEE 15th Int. Conf. Netw., Sens. Control*, Zhuhai, China, Mar. 2018, pp. 1–6.

[21] W. Zhang, C. Li, G. Peng, Y. Chen, and Z. Zhang, "A deep convolutional neural network with new training methods for bearing fault diagnosis under noisy environment and different working load," *Mech. Syst. Signal Process.*, vol. 100, pp. 439–453, Feb. 2018.

[22] L. Wen, X. Li, L. Gao, and Y. Zhang, "A new convolutional neural network-based data-driven fault diagnosis method," *IEEE Trans. Ind. Electron.*, vol. 65, no. 7, pp. 5990–5998, Jul. 2018.

[23] R. Liu, G. Meng, B. Yang, C. Sun, and X. Chen, "Dislocated time series convolutional neural architecture: An intelligent fault diagnosis approach for electric machine," *IEEE Trans. Ind. Informat.*, vol. 13, no. 3, pp. 1310–1320, Jun. 2017.

[24] J. Liu and Y. Shao, "Overview of dynamic modelling and analysis of rolling element bearings with localized and distributed faults," *Nonlinear Dyn.*, vol. 93, no. 4, pp. 1765–1798, Sep. 2018.

[25] L.-H. Wang, X.-P. Zhao, J.-X. Wu, Y.-Y. Xie, and Y.-H. Zhang, "Motor fault diagnosis based on short-time Fourier transform and convolutional neural network," *Chin. J. Mech. Eng.*, vol. 30, pp. 1357–1368, Nov. 2017.

[26] K. Chen, X.-C. Zhou, J.-Q. Fang, P.-F. Zheng, and J. Wang, "Fault feature extraction and diagnosis of gearbox based on EEMD and deep briefs network," *Int. J. Rotating Mach.*, vol. 2017, Jun. 2017, Art. no. 9602650.

[27] S. Guo, T. Yang, W. Gao, and C. Zhang, "A novel fault diagnosis method for rotating machinery based on a convolutional neural network," *Sensors*, vol. 18, no. 5, p. 1429, May 2018.
[28] W. F. Sun *et al.*, "An intelligent gear fault diagnosis methodology using a complex wavelet enhanced convolutional neural network," *Materials*, vol. 10, no. 7, p. 790, Jul. 2017.
[29] M. Zhao, M. Kang, B. Tang, and M. Pecht, "Deep residual networks with dynamically weighted wavelet coefficients for fault diagnosis of planetary gearboxes," *IEEE Trans. Ind. Electron.*, vol. 65, no. 5, pp. 4290–4300, May 2018.
[30] R. Yan, R. X. Gao, and X. Chen, "Wavelets for fault diagnosis of rotary machines: A review with applications," *Signal Process.*, vol. 96, pp. 1–15, Mar. 2014.
[31] P. Gangsar and R. Tiwari, "Multifault Diagnosis of induction motor at intermediate operating conditions using wavelet packet transform and support vector machine," *J. Dyn. Syst., Meas., Control*, vol. 140, Mar. 2018, Art. no. 081014.
[32] P. Y. Dibal, E. N. Onwuka, J. Agajo, and C. O. Alenoghena, "Enhanced discrete wavelet packet sub-band frequency edge detection using Hilbert transform," *Int. J. Wavelets, Multiresolution Inf. Process.*, vol. 16, no. 1, Jan. 2018, Art. no. 1850009.
[33] X. Zong, A. Men, and B. Yang, "Rate-distortion optimal wavelet packet transform for low bit rate video coding," in *Proc. IEEE Int. Workshop Imag. Syst. Techniques.*, Shenzhen, China, May 2009, pp. 364–368.
[34] Y. Cheng, D. Wang, P. Zhou, and T. Zhang, "A survey of model compression and acceleration for deep neural networks," *IEEE Signal Process. Mag.*, pp. 1–10, Oct. 2017. [Online]. Available: https://arxiv.org/abs/1710.09282
[35] Y. L. Cun *et al.*, "Handwritten digit recognition with a back-propagation network," in *Proc. Adv. Neural Inf. Process. Syst.*, 1990, pp. 396–404.
[36] B. Yan, Q. Chen, R. Ye, and X. Zhou, "Insulator detection and recognition of explosion based on convolutional neural networks," *Int. J. Wavelets Multiresolution Inf. Process.*, vol. 17, no. 2, Mar. 2019, Art. no. 1940008. doi: 10.1142/S0219691319400083.
[37] W. Kai, A. Jun, X. Zhao, and J. Zou, "Accurate landmarking from 3D facial scans by CNN and cascade regression," *Int. J. Wavelets Multiresolut. Inf.*, vol. 16, no. 2, Feb. 2018.
[38] A. G. Howard *et al.* (2017). "MobileNets: efficient convolutional neural networks for mobile vision applications." [Online]. Available: https://arxiv.org/abs/1704.04861
[39] L. Sifre and S. Mallat, "Rigid-motion scattering for texture classification," *Comput. Sci.*, 2014. [Online]. Available: https://arxiv.org/abs/1403.1687v1. doi: 10.1007/11503415_34.
[40] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. 13th Int. Conf. Artif. Intell. statistics*, Mar. 2010, pp. 249–256.
[41] K. He and J. Sun, "Convolutional neural networks at constrained time cost," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Boston, MA, USA, Jun. 2015, pp. 5353–5360.
[42] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Las Vegas, NV, USA, Jun. 2016, pp. 770–778.
[43] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. 32nd Int. Conf. Mach. Learning.*, Mountain View, CA, USA, vol. 37, 2015, pp. 448–456.
[44] W. L. Shang, K. Sohn, D. Almeida, and H. Lee, "Understanding and improving convolutional neural networks via concatenated rectified linear units," in *Proc. 33rd Int. Conf. Mach. Learn.*, Ann Arbor, MI, USA, vol. 48, Jun. 2016, pp. 2217–2225.
[45] X. Zhang, X. Zhou, M. Lin, and J. Sun. (Jul. 2017). "ShuffleNet: An extremely efficient convolutional neural network for mobile devices." [Online]. Available: https://arxiv.org/abs/1707.01083
[46] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen. (2018). "MobileNetV2: Inverted residuals and linear bottlenecks." [Online]. Available: https://arxiv.org/abs/1801.04381
[47] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proc. Eur. Conf. Comput. Vis.*, vol. 8689, Sep. 2014, pp. 818–833.
[48] J. Yosinski, J. Clune, A. Nguyen, T. Fuchs, and H. Lipson, "Understanding neural networks through deep visualization," in *Proc. ICML Workshop Deep Learn.*, Lille, France, Jun. 2015, pp. 1–9.
[49] (Dec. 2018). Case Western Reserve University (CWRU) Bearing Data Center. [Online]. Available: http://www.eees.ease.edu/laboratory/bearing
[50] H. Qiu, J. Lee, J. Lin, and G. Yu, "Wavelet filter-based weak signature detection method and its application on rolling element bearing prognostics," *J. Sound Vibrat.*, vol. 289, nos. 4–5, pp. 1066–1090, Feb. 2006.
[51] T. Williams, X. Ribadeneira, S. Billington, and T. Kurfess, "Rolling element bearing diagnostics in run-to-failure lifetime testing," *Mech. Syst. Signal Process.*, vol. 15, no. 5, pp. 979–993, Sep. 2001.
[52] Y. Lei, N. Li, L. Guo, N. Li, T. Yan, and J. Lin, "Machinery health prognostics: A systematic review from data acquisition to RUL prediction," *Mech. Syst. Signal Process.*, vol. 104, pp. 799–834, May 2018.

SHANGJUN MA was born in Qingyang, Gansu, China, in 1980. He received the B.S. degree in traffic engineering from the Taiyuan University of Science and Technology, Taiyuan, Shanxi, China, and the M.S. and Ph.D. degrees in mechanical engineering from Northwestern Polytechnical University, Xi'an, Shaanxi, in 2009 and 2013, respectively. He is currently an Associate Researcher with the School of Mechanical Engineering, Northwestern Polytechnical University. His research interests include design and analysis of mechanical transmission and the fault diagnosis of rotating machinery, and electromechanical actuator and planetary roller screw mechanism.

WENKAI LIU received the B.S. degree from the College of Computer Science, Chongqing University, Chongqing, China, in 2016, where he is currently pursuing the M.S. degree. His research interests include signal processing and machine learning.

WEI CAI received the B.S. degree in mechanical and electrical engineering from the Henan University of Science and Technology, Luoyang, Henan, China, in 2017. He is currently pursuing the M.S. degree in mechanical engineering with Northwestern Polytechnical University, Xi'an, Shaanxi, China. He is currently working on fault diagnosis of rotating machinery and planetary roller screw mechanism.

ZHAOWEI SHANG received the Ph.D. degree in computer science from Xi'an Jiaotong University, Xi'an, China, in 2005. He is currently a Professor with the College of Computer Science, Chongqing University, Chongqing, China. His research interests include pattern recognition, machinery health prognostics, wavelet theory, hyperspectral images, and signal processing. He is an Associate Editor of the *International Journal on Wavelets, Multiresolution, and Information Processing*.

GENG LIU received the Ph.D. degree in mechanical engineering from Xi'an Jiaotong University, Xi'an, in 1994. He is currently a Professor, a Supervisor of Ph.D. students, and the Director of the Shaanxi Engineering Laboratory for Transmissions and Controls, Northwestern Polytechnical University, Xi'an, Shaanxi, China. His research interests include mechanical dynamic design, mechanical system dynamics, simulation and virtual prototype design, tribology, contact mechanics, and numerical methods.