Multimedia Tools and Applications (2024) 83:71117–71149
https://doi.org/10.1007/s11042-024-18435-1

A multi convolution pooling group fault diagnosis model with high generalization across data sets and large receptive field characteristics considering industrial environmental noise

Wujiu Pan[1,2] · Shuming Cao[1,2] · Liang Xu[1,2] · YingHao Sun[1,2] · Peng Nie[1,2]

Received: 18 June 2023 / Revised: 9 September 2023 / Accepted: 22 January 2024 /
Published online: 6 February 2024

Abstract
Considering the noise impact in the bearing operating environment and the time-consuming and non-universal design of traditional diagnostic algorithms, this paper proposes a new model for rolling bearing fault diagnosis, which uses convolutional pooling group (CPG) to extract features from data, At the same time, expanding the dual convolutional kernel to obtain a larger receptive field obtained the WCPGCNN (A CPG Convolutional Neural Network with Wide Convolutional Kernel as the First Lay) model based on the CPG network architecture. Firstly, the fault features of the input signal are automatically extracted through four convolutional pooling groups; Next, fault features are further extracted using the fully connected layer, and finally input into the Softmax layer for fault identification. By utilizing algorithms such as Adam, dropout, and batch normalization, the model performs well in terms of accuracy, noise resistance, and timeliness, while also possessing good cross dataset high generalization ability. This article uses the rolling bearing fault standard data from Case Western Reserve University (CWRU) and the American Society for Mechanical Fault Prevention Technology (MFPT), and verifies through multiple controlled experiments that the model established in this article has high accuracy and good generalization characteristics.

Keywords Multi convolutional pooling group · deep learning · Bearing fault diagnosis · Ambient noise · Convolutional neural network

✉ Wujiu Pan
panspace@sina.cn

[1] School of Mechatronics Engineering, Shenyang Aerospace University, Shenyang 110136, People's Republic of China

[2] Advanced Manufacturing Technology Research Center, Shenyang Aerospace University, Shenyang 110136, People's Republic of China

1 Introduction

With the continuous development of modern science and technology, rotating machinery is also constantly moving towards intelligence. Rolling bearings are known as "industrial joints" and are widely used in various fields. Due to the significant impact of the nonlinear characteristics of bearings on the vibration characteristics of rotor systems [1, 2], in industrial sites, in order to identify the source of faults before they occur, real-time monitoring of the vibration generated by bearings during machine operation is usually carried out. Especially in harsh working environments and unstable conditions, it is easy to cause rolling bearings to malfunction, and it is even more necessary to grasp the health status of the bearings.

However, in actual operation, the failure of rolling bearings may occur due to various reasons, such as peeling, burns, crack defects, cage damage, scratches, rust and corrosion, etc. Compared to other mechanical components, rolling bearings have one of the most prominent characteristics, which is their large discrete lifespan. Even if the same processing equipment, process, and materials are used, and the same worker processes the same batch of rolling bearings, their lifespan also varies greatly. Rolling bearings are affected by high temperature, high pressure, high noise, humid air, corrosive gases, and dust during operation, which increases the uncertainty of their lifespan [3–5]. When the failure of rolling bearings causes mechanical equipment to stop working, it can cause economic losses in mild cases, catastrophic accidents in severe cases, and even threaten the safety of citizens' lives [6, 7]. Therefore, the fault diagnosis technology of rolling bearings is of great significance in the transformation and upgrading process of the manufacturing industry.

At present, there are endless methods for fault diagnosis, and bearing fault diagnosis has shifted from the initial "seeing, touching, and listening" to more advanced methods such as machine learning and deep learning. Bearing fault diagnosis can be divided into signal feature extraction and algorithm diagnosis [8], and the current popular signal feature extraction methods can be roughly divided into three stages. The first stage is the traditional signal analysis method, where the signals collected by sensors are often mixed with useless noise signals, which can lead to non-standard measured data. So, it is necessary to find a method to extract useful features inherent in bearings. The classic traditional signal analysis methods include Fourier transform [9], wavelet transform [10], empirical mode decomposition [11], singular value decomposition [12], etc. The shortcomings of these methods are that subjective factors have a significant impact on the diagnostic results and the accuracy of classification is not high. The second stage is that modern methods mainly revolve around machine learning, using methods such as statistical analysis or correlation analysis to achieve accurate fault identification. For example, k-nearest neighbor (KNN) [13], artificial neural network (ANN) [14], variational mode decomposition (VMD) [15], support vector machine (SVM) [16] can be used for classification. In recent years, Han et al. [17] proposed a freely switchable CNN-SVM system, which solves the problem of complex model training with small sample data. At the same time, the system has advantages such as low time consumption, high accuracy, and strong generalization ability. Sinitsin et al. [18] established a new hybrid CNN-MLP model diagnostic method that combines mixed inputs for rolling bearing diagnosis.

The third stage is achieved through deep learning, which uses its algorithm to identify and classify bearings, reads fault features from the original signal or signals, and trains the deep network model to complete end-to-end direct diagnosis. This method has less dependence on human experience knowledge, higher diagnostic efficiency, more accurate results,

and is more conducive to real-time detection. In recent years, deep learning based methods with automatic feature learning capabilities have received increasing attention, such as deep belief networks (DBN) [19], deep residual network (DRL) [20], convolutional neural networks (CNN) [21], and so on. Oh et al. [22] used DBN and vibration imaging to classify different faults in rotor systems. Abid et al. [23] constructed SAE using extracted multi domain features of vibration signals to diagnose bearing faults. Long and Short Term Memory (LSTM) can improve the long-term dependency problem in traditional Recurrent Neural Networks (RNNs), but its training efficiency is reduced and the calculation time is long. The Khoram and Khalooei [24] LSTM models replace the RNN model for fault diagnosis, solving the problem of RNN being unable to obtain information from a long time ago, thereby improving the performance and classification accuracy of the model. In order to address this drawback of the RNN model, An et al. [25] proposed a new RNN model that can ignore the influence of different rotational speeds. In this process, the LSTM model was used to compensate for the shortcomings in the RNN model, thereby improving the performance of the model. In order to solve the problem of imbalanced data not meeting the training requirements of intelligent networks, Peng et al. [26] proposed a new imbalanced fault diagnosis framework based on generative adversarial networks (GAN), and combined Wasserstein loss with hierarchical feature matching loss to achieve higher classification accuracy with fewer data samples. In order to improve the performance of fault diagnosis under imbalanced data, Liu et al. [27] proposed a new data synthesis method based on generative adversarial networks (GAN) and designed a new generator objective function, which has shown great potential in imbalanced fault diagnosis. Due to the difficulty in determining the parameters of the GAN model, which makes it difficult for the model to converge, Liu et al. [28] embedded self-correction in the generator of the GAN, enabling the generator to update parameters simultaneously based on the input and feedback of the discriminator, thus solving the difficulty of the GAN model convergence. The GAN model has advantages in data preprocessing and effectively solves the problem of model training for imbalanced data. However, it is difficult to achieve good synchronization between the generator and discriminator of the GAN model, which makes it difficult for the model to converge. Yang et al. [29] proposed a transformer neural network bearing fault diagnosis method based on attention mechanism. By segmenting the original data, linearly encoding and positional encoding the subsequences, and feeding the encoded subsequences back to the Transformer for feature extraction, fault recognition is achieved. The Transformer neural network has achieved excellent results in feature extraction, and self-attention can generate more explanatory models. However, the computational complexity and efficiency of Transformer neural networks are too high, which also makes it temporarily unable to replace mainstream deep learning algorithms.

For Convolutional Neural Networks (CNN), as one of the representative algorithms of deep learning, it has the characteristic of "end-to-end" and can directly read the original vibration signal. It also has multiple functions such as feature extraction and classifier classification. This algorithm was first used to solve problems such as speech recognition [30], medical imaging [31], and computer vision [32]. Considering the characteristics of CNN models, many scholars have attempted to use CNN to solve bearing failure problems and have achieved certain results. Janssens et al. [33] directly used vibration signals as inputs to the network, maximizing the preservation of signal data features, reducing human intervention, and reducing the difficulty and cumbersome steps in feature extraction. This model only has one convolutional layer, one fully connected layer, and one Softmax layer, and can only distinguish four types. Gültekin et al. [34] established a convolutional neural network bearing fault diagnosis model based on time segmented Fourier synchronous squeezing

transform, and selected the CNN model with the best parameters to evaluate the fault classification ability. Zhang et al. [35, 36] established classic network models such as WDCNN and TIDCNN for time-series vibration signals, and conducted detailed research on the recognition performance in environments such as variable loads and noise. Wang et al. [37] improved the multi-scale convolutional neural network bearing fault diagnosis model by using the reconstructed signal of the optimized VMD decomposition mode component as input to the CNN to obtain the fault diagnosis model. Levent et al. [38] constructed a shallow convolutional neural network model consisting of three convolutional layers and one fully connected layer to identify and classify bearing faults, and verified the effectiveness and feasibility of this model structure. Due to inaccurate classification based on artificial experience, Gao et al. [39] replaced traditional momentum in CNN with Nesterov momentum, improving the accuracy of fault identification. In order to adapt to different signal features, Wang et al. [40] used particle swarm optimization to determine the parameters of convolutional neural networks, and used t-distribution random neighbor embedding (t-SNE) to visualize the hierarchical feature learning process, achieving good results. In order to overcome the limitation of having too many training parameters in CNN, Xu et al. [41] constructed a deep convolutional neural network architecture with fewer training parameters using a proportional exponential linear unit activation function and a global mean pool. Zhang et al. [42] designed a multi-scale full spectrum CNN (MH-CNN) that maps time-domain signals to the time–frequency plane using continuous wavelet transform to fully reflect the complex information contained in the signal. Then, two-dimensional multi-scale feature fusion is introduced to extract features at different scales, which can consider both global and local information. Liu et al. [43] utilized multi-sensor data fusion technology to handle complex conditions in fault diagnosis and proposed an integrated convolutional neural network model for bearing fault diagnosis to reduce information loss during the fusion process. Jin et al. [44] proposed a new intelligent fault diagnosis method based on convolutional neural networks and bidirectional short-term memory networks to address the difficulties of traditional rolling bearing fault diagnosis methods in noisy and variable load environments.

Although the neural network models mentioned above have achieved some good results, they have not taken into account the impact of noise. Once the noise signal is integrated, the bearing vibration signal will become unstable, leading to diagnostic errors. At the same time, they cannot comprehensively consider the characteristics of rolling bearing fault data, and only a single dataset model is used for diagnosis. Therefore, it is necessary to establish a diagnostic model with high accuracy and generalization features. Therefore, this article proposes a new convolutional neural network model WCPGCNN (ACPG Convolutional Neural Network with Wide Convolutional Kernel as the First Lay). The main contributions of this article are: (1)This article proposes a new model for fault diagnosis of rolling bearings—using convolutional pooling group (CPG) for feature extraction of data, while expanding the double-layer convolutional kernel to obtain a larger receptive field. A wide convolutional kernel convolutional neural network WCPGCNN model based on CPG network architecture is obtained, which has multiple layers and fewer parameters, and can effectively extract short-term features, Has excellent ability to suppress overfitting and nonlinear expression. (2) Considering the actual noise environment in industrial scenarios, this model has excellent accuracy and anti-interference ability. (3) On the basis of establishing the model, the impact of different batch sizes, sample numbers, and iteration times on the accuracy of the model under the same dataset was studied, as well as the diagnostic performance of different models under the same dataset and multiple data types. A T-SNE dimensionality reduction flowchart was drawn. (4) The model proposed in this article has

strong generalization and adaptability. In the cross load test, there is still a high classification accuracy.

This article is divided into five parts. After the introduction, the second section introduces the parameter selection of each layer of convolutional neural networks. The third section elaborates on the process of building a diagnostic model in detail. In the fourth section, the model constructed in the third section was used to conduct a control experiment with multiple sets of different variables. The fifth section provides the conclusion.

2 Theoretical background

2.1 Convolutional layer

In convolutional neural networks, the role of convolutional layers is to extract input features and generate corresponding features. The most important feature of convolutional layers is weight sharing. The one-dimensional convolutional layer operation formula is as follows [45].

$$k_1(l) = \sum_{x=1}^{m} w_x^l r_x + b^l,\ l = 1, 2, \cdots, n \tag{1}$$

where, $k_1(l)$ is the feature extracted from the l-th convolutional kernel, w^l and b^l are the weights and deviations of the l-th convolutional kernel r_x represents one-dimensional input, m represents the number of data points in r_x. Similarly, the two-dimensional convolution operation is shown in Eq. 2.

$$k_2(l) = \sum_{x=1}^{m} \sum_{y=1}^{n} w_{x,y}^l r_{x,y} + b^l,\ l = 1, 2, \cdots, n \tag{2}$$

where, $k_2(l)$ is the feature extracted from the l-th convolutional kernel, $r_{x,y}$ represents 2D input, m and n respectively represents the number of data points in channels x and y in $r_{x,y}$.

2.2 Activation layer

In the forward propagation process of convolutional neural networks, the activation layer is usually added after the convolutional layer, and the output of the convolutional layer is subjected to nonlinear transformation. The function of the activation layer is to map the linear output of the convolution operation to another space, where the linear separability of features is enhanced. In multilayer neural networks, there is a functional relationship between the input and output of each layer, which is called the activation function, also known as the activation function.

The commonly used activation function in neural networks include Sigmaid function, hyperbolic Tangent function Tanh and modified linear unit ReLU (Rectified Linear Unit), because the output is bounded, it is easy to use as input for the next layer. In this paper, the linear rectification function ReLU is used as the activation function of the convolutional neural network, which can not only avoid the gradient dispersion phenomenon, but also accelerate the convergence of CNN. The mathematical expression of ReLU activation function is shown in Eq. (3), and the curve is shown in Fig. 1 [46, 47].

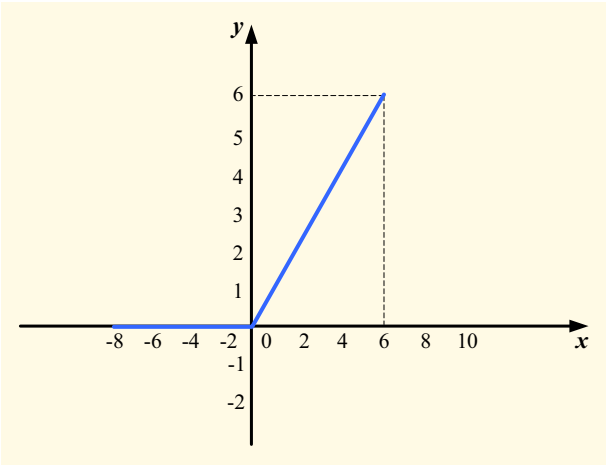


Fig. 1 ReLU activation function

$$a^{l(i,j)} = ReLU\left(x^{l(i,j)}\right) = \max(0, x^{l(i,j)}) \tag{3}$$

where, $x^{l(i,j)}$ represents the j-th feature value in the i-th feature map of the l-th convolutional layer, $d^{l(i,j)}$ is the activation value obtained by the activation function of $x^{l(i,j)}$.

2.3 Pooling layer

The pooling layer is a down sampling operation. The main purpose is to reduce the parameters of the neural network. Pooling layer is usually added after the activation layer. It reduces the amount of calculation by compressing feature dimensions and reducing network parameters, to some extent, it prevents overfitting, and enables the model to extract features in a larger range. The model constructed in this article will adopt Max Pooling operation, which can obtain position independent feature values in the perceptual domain. The mathematical expression for the maximum pooling operation is shown in Eq. (4) [36].

$$y^{l(i,j)} = max\left\{x^{l(i,j)}\right\} \tag{4}$$

where, $x^{l(i,j)}$ represents the activation value of the t-th neuron in the i-th feature map of the l-th layer.

2.4 Fully connected layer

The function of the fully connected layer is to recognize and classify the features extracted at the filtering level. Mainly, the output features obtained from the last pooling layer are spread out into one-dimensional feature vectors, and then the features are extracted and classified, usually combined with Softmax classification networks [48].

$$y^{l+1(j)} = \sum_{i=1}^{n} w_{ij}^{l} x^{l(i)} + b_j^l \tag{5}$$

where, $x^{l(i)}$ represents the output value of layer l; w_{ij}^{l} is the weight of the i-th neuron in layer l and the j-th neuron in layer $l+1$; b_j^l is the bias of all neurons in layer l towards the j-th neuron in layer $l+1$; $y^{l+1(j)}$ is the output of the j-th neuron in the $l+1$ layer.

Softmax classifier is a common multi class classification algorithm widely used in deep learning. It is widely used in fields such as image recognition, natural language processing, and speech recognition. Softmax classifier is a multi class classification algorithm. Its goal is to divide an input vector into multiple different categories. In the field of deep learning, Softmax classifiers are usually used in the output layer to divide the feature vectors of the previous layer into probability distributions corresponding to the target category. The meaning of Softmax is no longer to uniquely determine a certain maximum value, but to assign a probability value to each output classification result, representing the likelihood of belonging to each category. For each element z_i We exponentiate it $\exp(z_i)$, and then sum all elements: sum(exp (z_i)) to obtain the denominator result. Next, we can divide the above two results to obtain the expression of the Softmax Eq. (6):

$$p(x)_i = \frac{e^{Z_i}}{\sum_{k=1}^{C} e^{Z_k}}, i = 1, 2, \ldots, \mathrm{C} \tag{6}$$

where z_i is the inactive value of the i-th neuron in the output layer; C is the number of categories that need to be classified; $p(x)_i$ is the probability output of the i-th neuron in the output layer.

2.5 Loss function

The output value of an input signal on CNN should be consistent with its target value. The function to evaluate this consistency is called the Objective Function, or loss function. This paper uses the cross entropy function as CNN's loss function to measure the difference between the output probability distribution of the Softmax function and the probability distribution of the sample category. The mathematical expression of cross entropy loss function is shown in Eq. (7) [49].

$$L = -\frac{1}{m} \sum_{k=1}^{m} \sum_{j} p_k^j \log q_k^j \tag{7}$$

where m is the size of the input sample batch (mini batch); p_k^j represents the true classification results of the sample; q_k^j represents the Softmax output classification result of the sample.

2.6 Batch normalization layer and dropout layer

The BN layer mainly performs batch normalization processing. If the distribution of training data and test data is consistent, the generalization ability of the network will be greatly reduced. The gradient of each batch of training data is different, and the network will learn to adapt to different distributions in each iteration, which will greatly reduce the training speed of the network [50]. So the introduction of BN layer is to accelerate convergence speed and solve the problem of data distribution during the training process. Currently, the dropout layer is only used for the final fully connected layer.

Temporarily discarding some data can effectively alleviate the occurrence of overfitting, reduce the complex coadaptation relationship between neurons, and achieve the regularization effect to a certain extent.

2.7 Adam Optimization Algorithm

After calculating the derivatives of the loss function with respect to variables and parameters in each layer, optimization algorithms are needed to update the weights and biases of the convolutional layer and fully connected layer, repeatedly updating to reduce the value of the loss function until the value of the loss function no longer changes or the number of iterations reaches the set value. In the training process of shallow models, SGD algorithm is often used as the optimization algorithm for model training due to the small number of network layers, parameters, and hyperparameters. This can converge to the global optimum with fewer iterations.

The CNN model proposed in this article belongs to a deep level model, and if optimized using the SDG algorithm, the result is likely to be a local optimal solution. The Adam algorithm can not only accelerate the convergence of the model, but also adaptively adjust the learning rate of each parameter. Therefore, choosing Adam algorithm as the optimization algorithm for the deep CNN model proposed in this article can effectively solve the problem of parameter optimization in the deep model. The Adam algorithm flowchart is shown in Fig. 2.

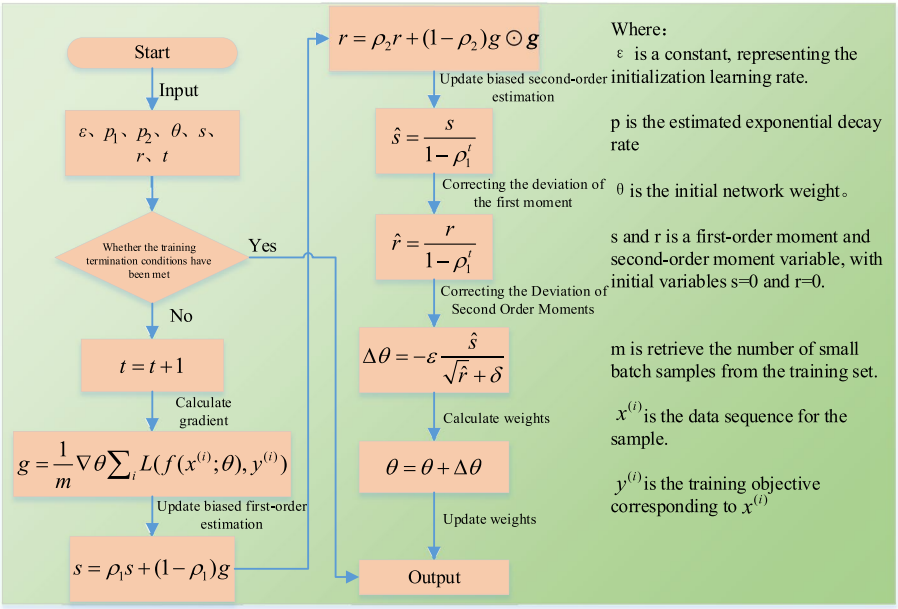


Fig. 2 Adam Algorithm Process

3 The proposed method and the data used for experiment

3.1 WCPGCNN model modeling

Inspired by VGGnet [51], multiple convolutional and pooling layers were used to form a Convolutional Pooling Group (GPG). Each GPG contains two convolutional layers, one pooling layer, and one Dropout layer. In addition, the small core of the first layer is prone to interference from high-frequency noise commonly found in industrial environments. Therefore, in order to capture useful information in the low-frequency frequency range of vibration signals, we first use wide kernels to extract features, and then use continuous small kernels to obtain better feature representations. Therefore, a convolutional neural network with a wide convolutional kernel as the first layer composed of GPG is proposed. The network model of WCPGCNN (A CPG Revolution Neural Network with Wide Revolution Kernel as the First Lay). The proposed WCPGCNN method is shown in Fig. 3 As shown in the figure, it is divided into three parts, namely data preprocessing, model training, and result analysis.

When performing data preprocessing, first locate the file with the corresponding health status; Next, label these data; Finally, these data are proportionally divided into training sets, validation sets, and test sets, where the training set data is enhanced using the dataset.

When conducting model training, first initialize the parameters of the WCPGCNN model; Secondly, enable input data to enter the network for forward propagation; Next, update the model parameters using backpropagation of model errors; Finally, use the validation set to verify the effectiveness of this training session and output the training results of this iteration. Repeat this process until the set number of iterations is reached.

When conducting model testing, input the test set into the trained WCPGCNN model and output the test results.

The WCPGCNN model constructed in this article takes the raw fault signals of CWRU and MFPT as inputs. Firstly, the fault features of the input signal are automatically extracted through four convolutional pooling groups; Next, further extract fault features using the fully connected layer and input them into the Softmax layer; Finally, fault identification is performed in the Softmax classifier. During the training process, the cross entropy function is selected as the loss function of the WCPGCNN model to measure the difference between the output probability distribution of the Softmax function and the probability distribution of the sample category. The Adam algorithm is used as the optimizer of the network, and the learning rate is set to 0.002.

The schematic diagram of the WGPGCNN model structure is shown in Fig. 4, with the following characteristics.

a. WGPGCNN is composed of multiple convolutional pooling groups, which can achieve a larger Receptive field with fewer parameters. This not only increases the depth of the network, improves the expression ability of the network, but also effectively prevents the occurrence of overfitting.
b. The multi convolutional pooling group network model structure consists of four CPGs, each containing two convolutional layers and one maximum pooling layer. Since there are no parameters in the Global Average Pooling (GAP) layer and the data can be flattened, in order to reduce the number of parameters in the network structure, the GAP layer is used instead of the three full connection layers for flattening, which can effectively avoid overfitting of the model.

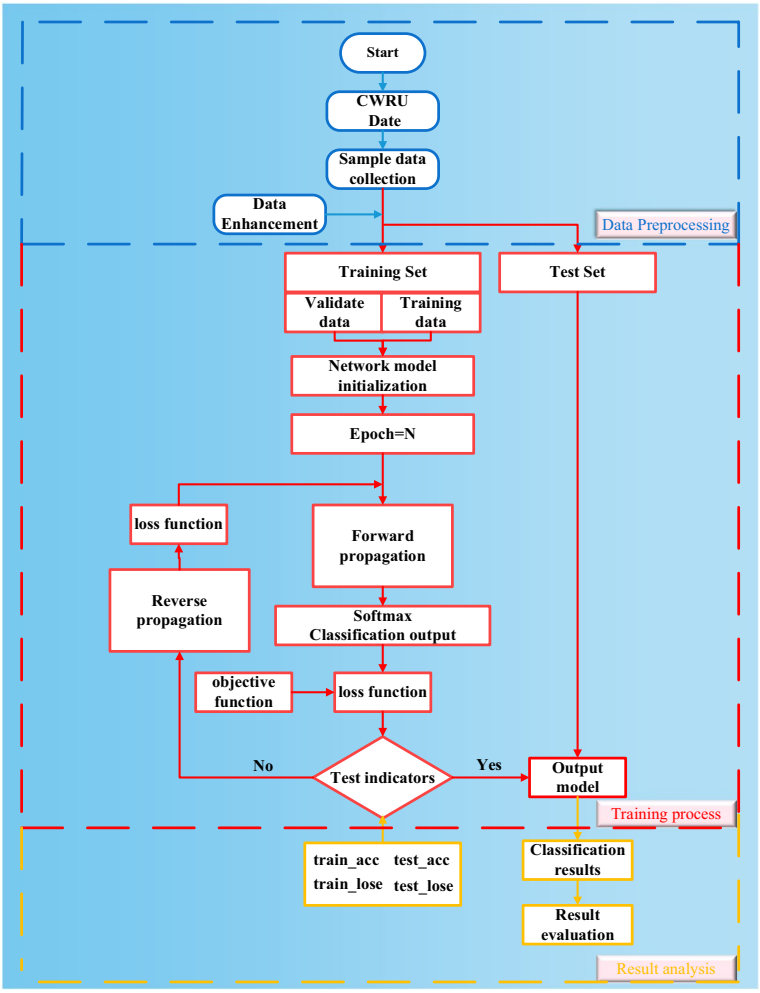


Fig. 3 WCPGCNN overall model method

c. The first convolutional layer uses a 64×1 wide convolutional kernel, which can more effectively extract short-term features and suppress high-frequency noise, helping to obtain a good representation of input signals and improving network performance.
d. In addition to the first and second convolution layers of the entire network, the rest of the convolution layers use 2×1 small convolution cores. With smaller convolution cores and deeper network layer structure, larger Receptive field can be obtained, and at the same time, the nonlinear expression ability and recognition ability of the model can be increased.

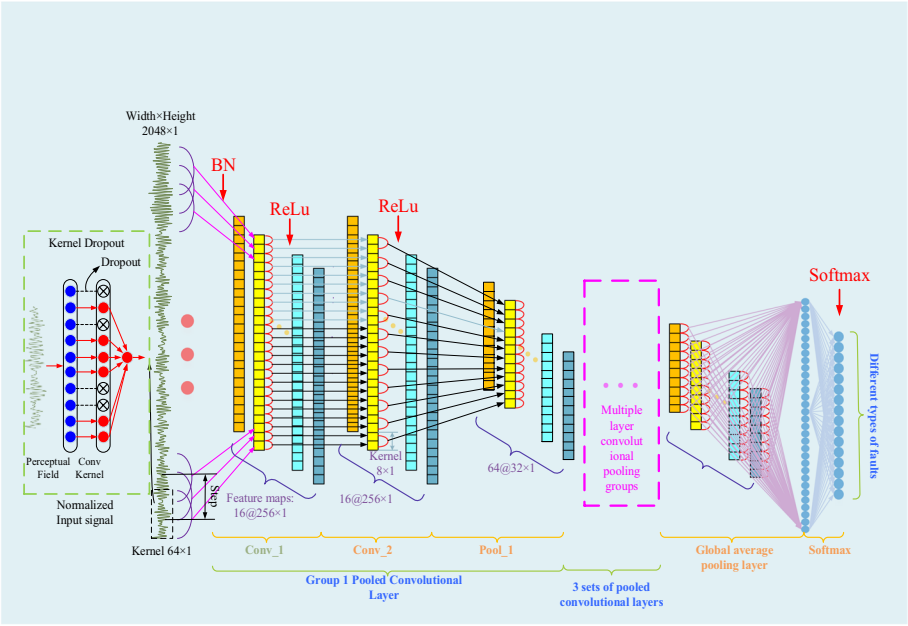


Fig. 4 WGPGCNN model structure diagram

3.2 Configuration

In Section 3.1, the main structure of WCPGCNN is introduced, and this section will design reasonable parameters for this model. The design core of convolutional neural networks is the receptive field, which is the perceptual range of a neuron in the lower layer of the network. In order to enable the network to learn more features, Using $T \leq R^0 \leq L$ as the design criterion, where, R^0 represents the receptive field of the last pooling layer neuron in the input signal; T represents the number of data points recorded by the accelerometer after one revolution of the bearing, and L represents the length of the input signal.

The neurons in the last pooling layer, R^l in the receptive field of layer l, and R^{l-1} in the receptive field of layer $l-1$, are shown in Eq. (8).

$$R^{(l-1)} = S^{(l)}\left(P^{(l)}R^{(l)} - 1\right) + W^{(l)} \tag{8}$$

where, $S^{(l)}$*and* $W^{(l)}$ represent the step size and kernel width of the l-th convolutional layer; $P^{(l)}$ represents the size of the pooling operation for the l th pooling layer.

Due to the particularity of the WCPGCNN model, when $l>1$, given $S^{(l)}=1$, $W^{(l)}=1$, and $P^{(l)}=2$. So, Eq. (8) can be written as Eq. (9):

$$R^{(l-1)} = 2R^{(l)} \tag{9}$$

When $l=n$, $R^{(n)}=1$.Among them, n is the number of convolutional layers in the network, and there are 8 convolutional layers in the WCPGCNN network model. So, $R^{(8)}=1$ and taken into Eq. (9), we obtain $R^{(1)}=128$. By combining Eq. (8), the receptive field of the neurons in the last pooling layer on the input signal can be obtained, as shown in Eq. (10).

$$R^{(0)} = 255 \times S^{(l)} + 128 \tag{10}$$

Due to $T \leq R^{0} \leq L$, and considering that $S^{(1)}$ must be divisible by l, the relationship used to constrain $S^{(1)}$ can be obtained, as shown in Eqs. 11 and 12.

$$T \leq 255 \times S^{(l)} + 128 \leq L \tag{11}$$

$$S^{(1)} | L \tag{12}$$

The input signal length of the WCPGCNN model constructed in this article is $L = 2048$, so the period $T \approx 400$. So only when the step size of the first convolutional layer is 2 or 4, can the model parameters of WLGCNN meet the design requirements. Therefore, in this article, the step size of the first convolutional layer is set to 4, as this can shorten the diagnostic time and increase efficiency. The specific model parameters are shown in Table 1.

The parameter settings of the WGPGCNN model are shown in Table 1. The input of the model is the frequency domain signal of 2048×1. The relevant hyperparameter of the model are set as follows through the test:

(1) Using the Adam optimizer, the learning rate is 0.0001.
(2) The cross entropy function is used as the loss function of the network.
(3) The pooling layer selects the maximum pooling operation.
(4) All activation function used by hidden layers are ReLU functions.
(5) The dropout rate of all dropout layers is set to 0.3.

The model proposed in this article has strong feature extraction and classification capabilities, achieving adaptive data-driven fault diagnosis. In the diagnosis process, it not only greatly reduces the impact of intermediate human intervention on data, but also greatly reduces the dependence on expert experience. The workflow of this article is shown in the Fig. 5:

Table 1 WGPGCNN model parameter configuration

Network layer	Nuclear size	Number of nuclei	step	Output depth	Padding
Conv_1-1	64	16		16	SAME
Conv _1-2	64	16	4	16	SAME
Pooling_1	2			16	SAME
Conv_2-1	8	64	2	64	SAME
Conv_2-2	8	64	2	64	SAME
Pooling_2	2			64	SAME
Conv_3-1	2	256	2	256	SAME
Conv _3-2	2	256	2	256	SAME
Pooling_3	2			256	SAME
Conv_4-1	2	512	1	512	SAME
Conv_4-2	2	512	1	512	SAME
Pooling_4	2			512	SAME
GAP				1	SAME
Softmax	16	1	1	1	SAME

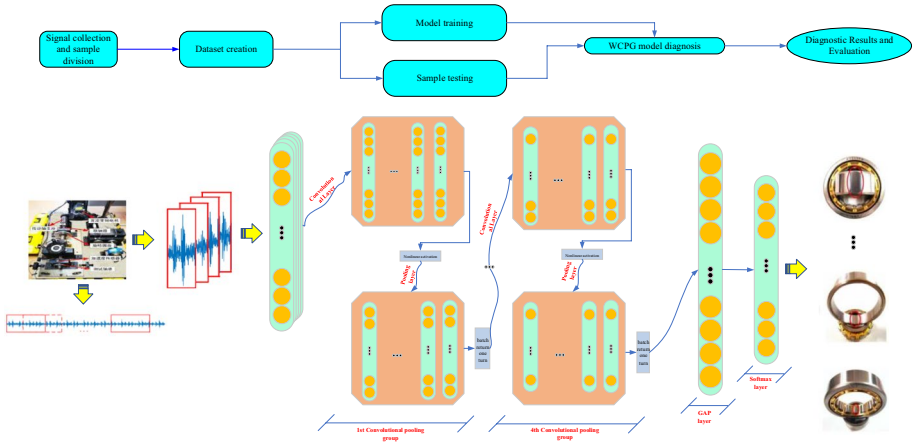


Fig. 5 Method flowchart

(1) By using dataset augmentation technology, samples are collected from the original fault data to form the original dataset;
(2) Normalize the data, and then classify the samples into test sets and training sets. The training set data is further divided into training data and validation data. The above three sets of data do not overlap, and the samples within the group are arranged in random order;
(3) Build a WCPGCNN model and use training set data for training. Obtain sample features through forward propagation, calculate the accuracy of the learned features compared to real features through loss function, and optimize network parameters through reverse propagation; Until the detection indicators of the set parameters meet the requirements, output and save the model that meets the conditions.
(4) Input the test set data into the model obtained in step (3) for diagnostic testing, and evaluate the diagnostic effectiveness of the model using accuracy, loss function, and confusion matrix.

3.3 Data source and classification

This article selects the open bearing failure standard dataset from Case Western Reserve University (CWRU) and the American Society for Mechanical Fault Prevention Technology (MFPT) in the United States. For CWRU, taking rolling bearing SKF6205 as the research object, the sampling frequency is 1.2 kHz. In the experiment, faulty bearings with diameter damage of 0.007 mm, 0.014 mm, and 0.021 mm were selected. Each fault diameter contains three types of faults: ball, inner ring, and outer ring.

The experimental dataset consists of 9 fault datasets and 1 normal dataset. Since the accelerometer is used to pick up vibration signals, it will be installed on the driver, fan and base of the motor housing, so that the measurement results of three different sensors can be obtained. This article selects 10 types of faults, with each diameter damage corresponding to three fault states. Please refer to Table 2 for details.

The MFPT bearing dataset is also used to validate the proposed model. It includes 17 sets of 4 fault situations and 3 sets of normal data. The load under normal bearing conditions is 270 pounds, and the input shaft speed is 25 Hz; The fault load of the three outer

Table 2 CWRU fault type parameters

Fault type	Fault diameter/inch	Label
Ball	0.007	1
	0.014	2
	0.021	3
Outer race	0.007	4
	0.014	5
	0.021	6
Inner race	0.007	7
	0.014	8
	0.021	9
Normal	0	10

rings is 270 pounds, and the input shaft speed is 25 Hz; The load of 7 outer ring faults is 25, 50, 100, 150, 200, 250, and 300 lbs respectively, and the input shaft speed is 25 Hz; The loads of 7 inner ring faults are 0, 50, 100, 150, 200, 250, and 300 lbs respectively, and the input shaft speed is 25 Hz. This article selects 10 types of faults. The parameters are shown in Table 3 below.

3.4 Data enhancement

Before conducting deep learning, it is necessary to label and enhance the collected data. In this respect, it is actually necessary to adjust the hyper-parameter, and correspond the input and output one by one. The data augmentation method used in this article is overlapping sampling. As shown in Fig. 6, there is a high possibility of overlap between two consecutive training samples when dividing training data from the original signal. When overlap occurs, sampling is performed as shown in the figure. The overlap ratio refers to the proportion of time overlap between adjacent time windows. For the overlap ratio, the larger the overlap ratio, the more time series samples generated and the better the quality, but at the same time, the computational complexity is also increasing.

The reason why the number of samples increases is because each time series sample contains multiple time windows, and there is also overlap between adjacent time windows.

Table 3 MFPT fault type parameters

Fault type	Load/ lbs	Sampling frequency	Label
Outer race	270	97,656	1
	25	48,828	2
	50	48,828	3
	100	48,828	4
	150	48,828	5
	200	48,828	6
Inner race	50	48,828	7
	100	48,828	8
	150	48,828	9
Normal	270	97,656	10

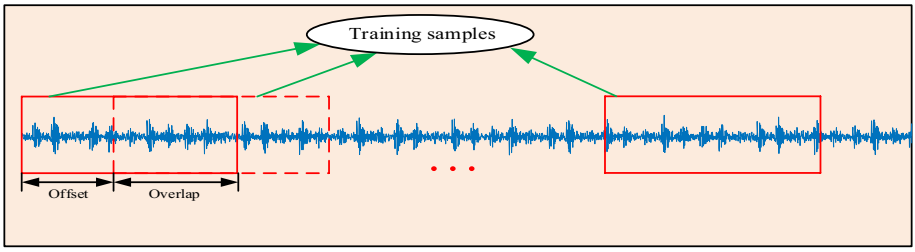


Fig. 6 Overlapping data enhancement

At the same time, there is inevitable overlap when sliding windows slide between samples. Therefore, the number of samples and the step size of the sliding window are both factors that affect the overlap ratio.

3.5 Data feature extraction

Before extracting and processing the features, all 10 types of fault data were mixed together in a disorderly manner with high dimensionality. When the dimensionality is high, the computational efficiency of the algorithm slows down because the complexity and computational complexity of the model are proportional to the dimensionality, so dimensionality reduction operations are required. Dimension reduction refers to the use of feature mapping methods to reduce data from high dimensions to low dimensions. Commonly used dimensionality reduction methods include PCA [52] and T_ SNE [53] et al. This article uses T_ SNE performs dimensionality reduction operation. The full name of T_ SNE is (T-distributed Stochastic Neighbor Embedding). T-distribution random nearest neighbor embedding is a technology that combines dimensionality reduction and rendering. It is based on SNE visualization enhancement and solves the characteristics of crowded sample distribution and blurred SNE boundaries after imaging. T_ SNE models the similarity of the original space as probability density, and the distribution of similarity is given by Gaussian distribution. The initial data graph without classification operation after dimensionality reduction is shown in Fig. 7. In short, in the original space, the similarity between a point and other points can be expressed by a probability density distribution:

$$p_{j|i} = \frac{\exp(-\frac{||x_i - x_j||^2}{2\sigma_i^2})}{\sum_{k \neq i} \exp(-\frac{||x_i - x_k||^2}{2\sigma_i^2})} \tag{13}$$

where, $p_{j|i}$ is the conditional probability, x_i is a random data point, x_j is the nearest neighbor point of x_i, $||x_i - x_j||^2$ is the distance between the two points, σ_i is the data point x_i is the Gaussian distribution standard deviation of the mean, x_k is the nearest neighbor point of x_i, $||x_k - x_j||^2$ is the distance between the two points. σ_i for each x_i are all different and have a predetermined confusing impact, σ_i is automatically set.

In the dimensionality reduced space, we use the T distribution instead of the Gaussian distribution because the T distribution can retain more similarity over longer distances. So in the target space after dimensionality reduction, the joint probability distribution is:

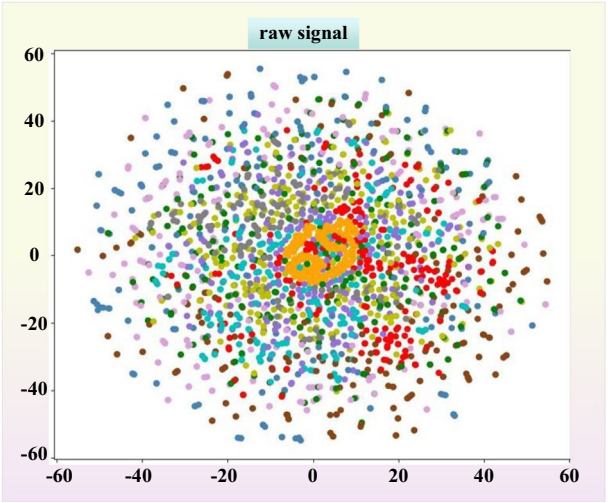


Fig. 7 Initial data visualization

$$q_{j|i} = \frac{(1 + ||y_i - y_j||^2)^{-1}}{\sum_{k \neq l}(1 + ||y_k - y_l||^2)^{-1}} \tag{14}$$

where, $q_{j|i}$ is the joint conditional probability density, y_i is the mapping point of high-dimensional data x_i in low latitude space, y_j is the mapping point of high-dimensional data x_j in low latitude space, $||y_i - y_j||^2$ is the distance between two points. y_k is the mapping point of high-dimensional data x_k in low latitude space, y_l is the mapping point of high-dimensional data x_l in low latitude space, $||y_k - y_l||^2$ is the distance between two points.

From the graph, it can be seen that the original input signal has the maximum entropy and the highest degree of confusion. Various faults are mixed together and cannot be separated, and the feature interval is relatively fuzzy. Each color in the figure represents a type of fault data, and the edge areas are very scattered. The more concentrated the center area is, the harder it is to distinguish. Therefore, deep learning should be used to identify different types of faults for better classification.

4 Validation of the WGPGCNN model

A controlled experiment refers to an experiment conducted to investigate the impact of a certain condition on a research object, in which all other conditions are the same except for different conditions. This chapter adopts a controlled experiment method, which only changes one variable at a time and keeps the other unrelated variables unchanged, to verify the accuracy of the model.

This chapter uses rolling bearing fault standard data from Case Western Reserve University (CWRU) and the American Society for Mechanical Fault Prevention Technology (MFPT). Through comparative experiments, the impact of different batch sizes, sample sizes, and iteration times on alignment accuracy of the model under the same dataset is analyzed, as well as the diagnostic performance of different models under the same dataset

and multiple data sets. At the same time, the impact of environmental noise was also considered, and the accuracy of the model under different noise levels was simulated. The detailed analysis flowchart is shown in Fig. 8.

4.1 Test results of different batch sizes

Select a set of samples in the training set to update the weight, with a value of generally 17. Choose a value that can be evenly divided by the test set. If the final training count cannot be segmented, the number of epochs will increase by 1. Because the dataset is divided in

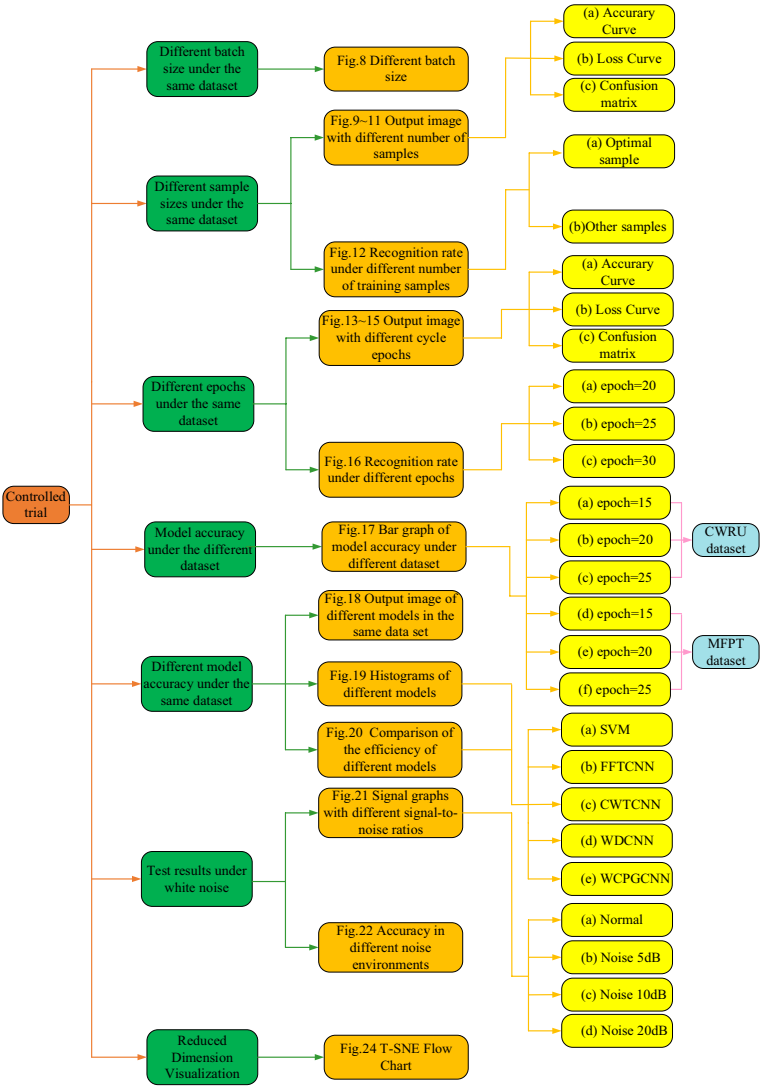


Fig. 8 Flow chart of experimental analysis

a 6:2:2 ratio, the batch size values are set to 16, 32, 35, 64, 70, and 128. In deep learning, SGD training is typically used, which involves obtaining batch size data from each training set. Its advantage is that it can accelerate training speed and occupy relatively less memory during the training process. After the weight update, the next batch of data can be used to improve the training speed, but it may cause significant gradient fluctuations. To prevent unexpected situations, multiple tests were conducted, and the average accuracy was calculated.

From Fig. 9, it can be seen that the accuracy is highest when the batch size value is 32. When the batch size value is large, the image does not converge, and the accuracy decreases sharply. At appropriate values, as the number of iterations increases, the overall accuracy shows an upward trend, but it also takes more time. To divide the batch size by the training set, set the batch size to 35 during this process.

4.2 Experimental results of different numbers of training samples

In order to train a large number of parameters, sufficient sample data is a prerequisite for WGPGCNN, in order to study how much training data is sufficient, and the performance of WGPGCNN under different sample data. So, in the experiment, WGPGCNN input training data of different sizes on groups of 60, 300, 600, 1200, 3000, 4800, 6000, 9000, and 12,000 training samples for training. During the training process, with a load of 0hp, using the Adam optimizer, the learning rate is 0.0001, and the number of cycles is 20.

It is important to find the appropriate number of training samples, as the larger the number of training samples, the longer the time required. However, it is also difficult to continue improving accuracy when reaching the bottleneck. Figure 10 represents the accuracy curves for training samples of 4800, 6000, and 9000, respectively, Fig. 11 represents the Loss curves for training samples of 4800, 6000, and 9000, From Figs. 10a and 11a, it can be seen that when the number of training samples is small, the fit is not good enough, resulting in violent fluctuations, making it difficult to distinguish fault types clearly. From Figs. 10b and 11b, it can be seen that when the number of training samples is 6000, the accuracy curves of the training and validation sets fit well, and the overall trend is close to 1.0. From the Loss curve, it can be seen that the validation setting steadily decreases and approaches 0.0. In this case, various types of faults are obvious. When it exceeds 6000, as

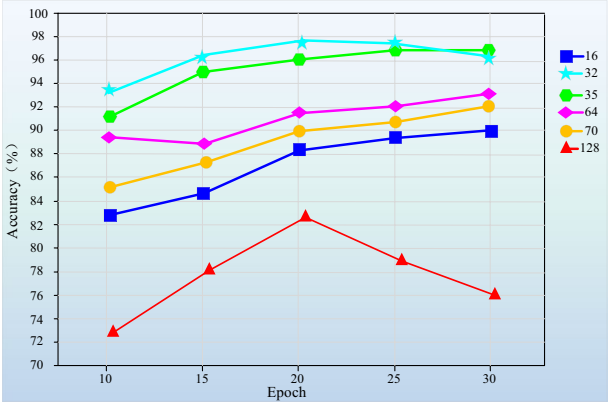


Fig. 9 Impact curve of different batch size values on accuracy

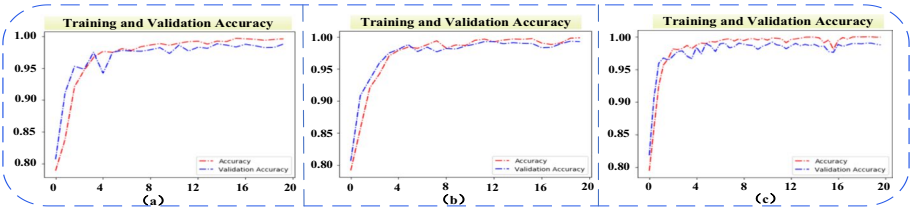


Fig. 10 Accuracy curve of WGPGCNN under different sample data

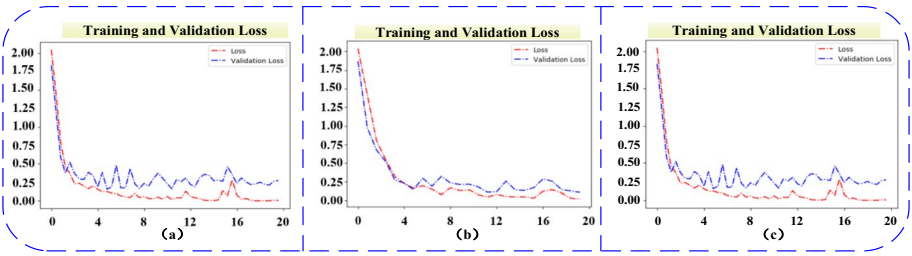


Fig. 11 Loss curve of WGPGCNN under different sample data

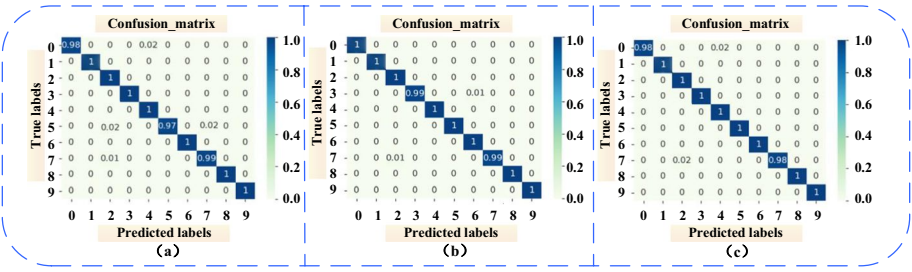


Fig. 12 Confusion matrix of WGPGCNN under different sample data

shown in Fisg. 10 c and 11c, there will be a slight overfitting phenomenon, and the curve will have a small major fluctuation, but it has little impact on the overall effect.

Figure 12 shows the confusion matrix for different sample sizes. Confusion matrix is the most basic and intuitive method to measure the accuracy of classification model. By observing the confusion matrix, the closer to 1, the higher the accuracy. By comparing the confusion matrix under different sample numbers, when the number of samples is 6000, the higher the accuracy is, the better the fault classification is.

Figures 13 and Table 4 show the recognition rates of different training sample sizes tested in the experiment. When the training sample is 12,000 times, the recognition accuracy is as high as 98.4%, while when the training sample is 60 times, the recognition accuracy is only 38.6%. The experimental results demonstrate the impact of training sample size on diagnostic accuracy. When the number of training samples exceeds 3000, the accuracy can reach over 93.8%, and it is not difficult to find that as the sample size increases, the accuracy initially improves significantly. However, after reaching a certain threshold, the speed of accuracy improvement begins to slow down, and after reaching the peak, the

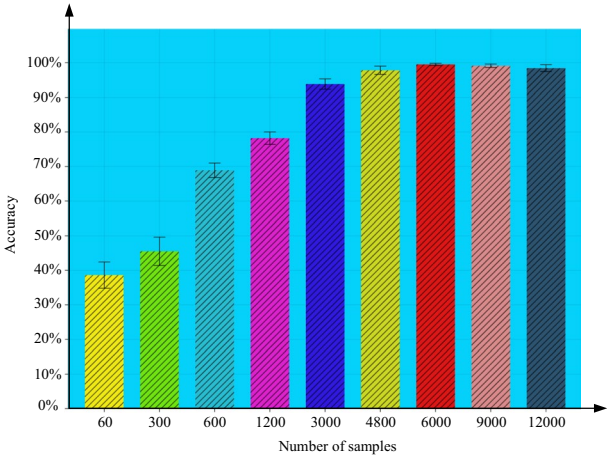


Fig. 13 Recognition rate of WGPGCNN under different number of training samples

Table 4 WCPGCNN recognition rate under different training samples

Number	60	300	600	1200	3000	4800	6000	9000	12,000
Accuracy(%)	38.6 ± 3.8	45.5 ± 4.1	68.9 ± 2.1	78.2 ± 1.8	93.8 ± 1.5	97.8 ± 1.2	99.5 ± 0.3	99.1 ± 0.5	98.4 ± 1.0

speed of accuracy improvement begins to slowly decrease. When the sample size is 6000, the accuracy is 99.50%, which is the vertex of the curve. When the sample size exceeds 6000, the accuracy slightly decreases and remains around 99%. Through this experiment, it is not difficult to conclude that 6000 sets of training data are an ideal value. Not only does it have the highest accuracy, but it can also meet the requirements of WGPGCNN in terms of quantity and scale. When the training sample data exceeds 6000, the accuracy rate of the training sample will increase, and the overfitting problem will cause the accuracy rate of the test set sample to decline. In other words, when the training data set is too small, there will be under fitting, and when the training data set is too large, there will be overfitting. Therefore, in the following experiment, the WGPGCNN model was trained with 6000 samples.

4.3 Experimental results under different training cycles

One epoch means that one loop represents the entire dataset being passed forward and backward only once in the neural network structure. For a convolutional neural network with a large training set, only one transmission is not sufficient to obtain accurate experimental results. In the same neural network, a complete dataset needs to undergo repeated transmission cycles to obtain results. As the number of epochs increases, the number of times the weights in the neural network are changed also increases. The test dataset goes from under fitting to optimal, and then to Overfitting.

It is important to find the correct number of iterations, as higher iterations require longer time. However, when the bottleneck is reached, it is difficult to continue improving accuracy. In this section, the number of data in the training set is 6000, the test dataset is 2000, the load is 0hp, and the learning rate of the Adam algorithm is 0.001. Figure 14

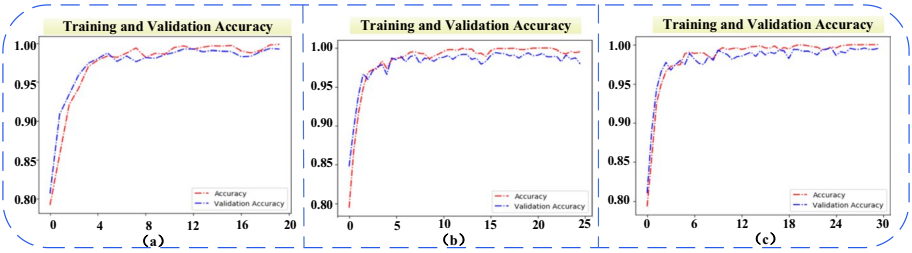


Fig. 14 Accuracy curves of WGPGCNN under different epochs

represents the accuracy curves with iterations of 20, 25, and 30, while Fig. 15 represents the loss curves with iterations of 20, 25, and 30, respectively, From Figs. 14a and 15a, it can be seen that when the number of iterations is 20, the accuracy curves of the training and validation sets are well fitted, and the overall trend is close to 1.0, with a high accuracy rate. From the loss curve, it can be seen that the validation setting steadily decreases and approaches 0.0. In this case, various types of faults are obvious. When the number of iterations exceeds 20, as shown in Figs. 14b and 15b, there will be a slight overfitting phenomenon, and the curve will have a small major fluctuation, but it has little impact on the overall effect. When the epoch increases again, the loss curve will experience severe fluctuations, as shown in Fig. 15c.

Figure 16 shows the confusion matrix under different iterations. By comparing the confusion matrix under different iterations, it can be seen that when epoch = 20, the higher the accuracy, the better the fault classification. With the increase of epoch, its precision decreases, indicating that a slight overfitting phenomenon has occurred.

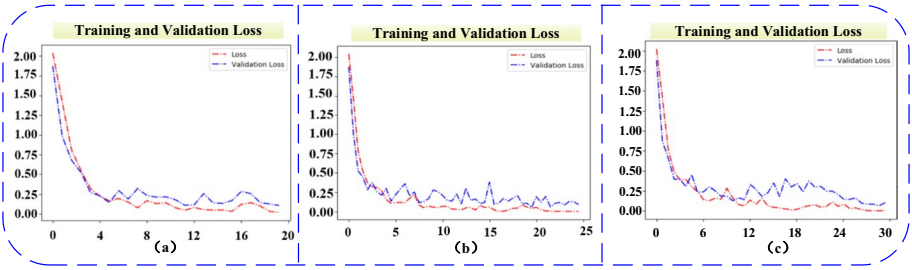


Fig. 15 Loss curves of WGPGCNN under different epochs

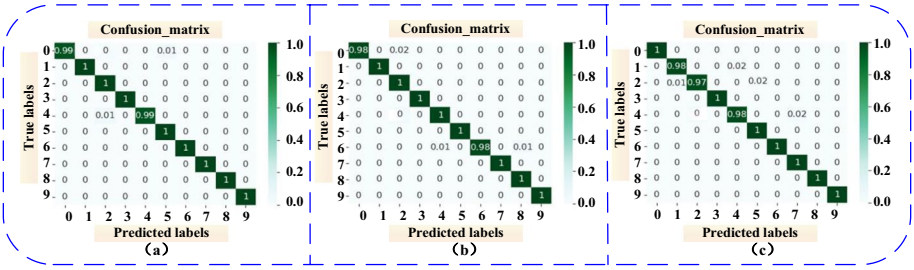


Fig. 16 Confusion matrix of WGPGCNN under Different Epoch

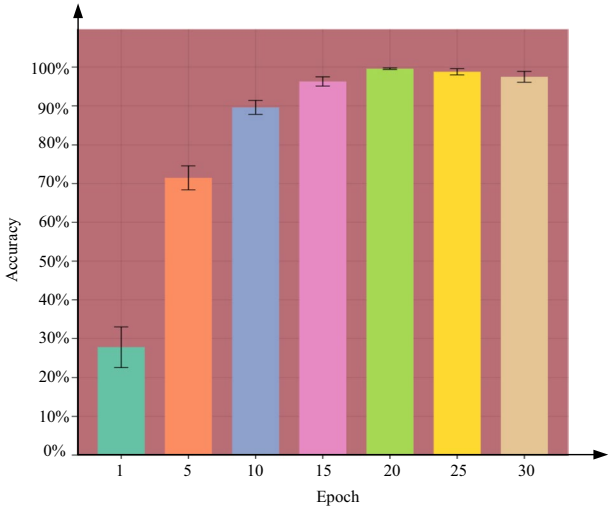


Fig. 17 Recognition rate of WGPGCNN under different epochs

Table 5 WCPGCNN recognition rate under different training times

Epoch	1	5	10	15	20	25	30
Accuracy(%)	27.8 ± 5.2	71.5 ± 3.1	89.6 ± 1.8	96.3 ± 1.2	99.6 ± 0.2	98.2 ± 0.8	97.5 ± 1.4

In this experiment, the epoch values in the WGPGCNN model were adjusted to 1,5,10,15,20,25,30. As shown in Fig. 17 and Table 5, the accuracy increased by 68.5% when epoch increased from 1 to 15. 20 epochs, with a maximum accuracy of 99.6%. When epoch increases to 25, the recognition rate is 98.2%, and when epoch increases to 30, the accuracy rate is only 97.5%. Obviously, the experimental results obtained strongly support our previous conclusions. When the epoch is between 1 and 20 times, the neural network model belongs to an under fitting state. When more than 20 times, the error of the training data set decreases, while the error of the test data set increases, and the neural network model belongs to the overfitting state. So we determined the optimal epoch number to be 20. If the number of epochs is too small, it will lead to under fitting, but if the number of epochs is too high, it will lead to overfitting. Compared to the past (1000 epochs), WGPGCNN can achieve higher recognition rates with fewer epochs.

4.4 Experimental results across datasets

Due to different operating conditions such as load and speed, the features extracted by signal extraction techniques are also different. In order to verify the generalization characteristics of the model and whether the model can operate stably under different working conditions, this article selects other fault datasets as input to the model for diagnosis, as shown in Table 6. Due to different operating conditions such as load and speed, the features extracted by signal extraction techniques are also different. To ensure that only one variable is changed, all data with a total of 10,000 were selected. Except for the original dataset 0HP, the other four types of datasets used different datasets of 1HP,

Table 6 Parameter tables for different datasets

	Train	Valid	Test	Motor load
DT I	7000	2000	1000	0HP
DT II	6000	2000	2000	0HP
DT III	7000	2000	1000	1HP
DT IV	6000	2000	2000	1HP
DT V	7000	2000	1000	2HP
DT VI	6000	2000	2000	2HP
DT VII	7000	2000	1000	3HP
DT VIII	6000	2000	2000	3HP
DT IX	7000	2000	1000	MFPT
DT X	6000	2000	2000	MFPT

2HP, 3HP, and MFPT, respectively. For each type of dataset, in addition to 6:2:2, the dataset is also divided according to 7:2:1. The differences in the observed results are shown in the histogram of Fig. 18. It can be seen that the model has good data processing ability under different working conditions. As the number of iterations increases, the overall trend is upward.

From Table 7, it can be seen that when the original dataset is divided by 6:2:2, the processing effect is slightly better than other datasets. Under low iteration, the accuracy of dataset V under 2HP operating conditions is 91.64%, and the processing ability is slightly poor. Overall, CWRU has weak adaptability to 2HP. When epoch = 20, the average accuracy is 99.05%. From the table, it can be seen that the fault diagnosis model established in this article has good adaptability and generalization characteristics. Due to different types of datasets, the model proposed in this article has slightly poor processing ability for MFPT data.

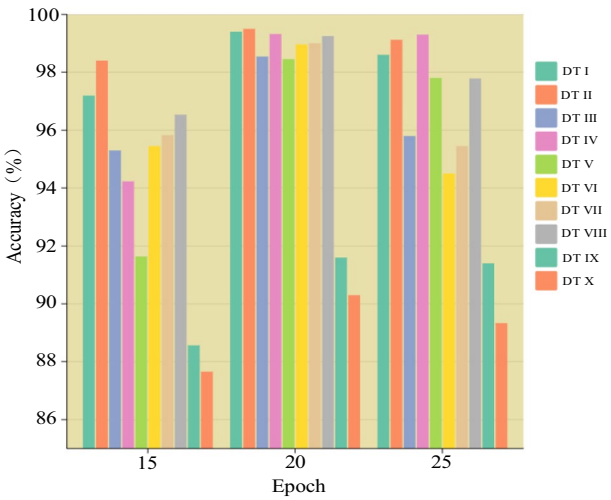


Fig. 18 Bar chart of model precision under different datasets

Table 7 Model Precision under Different Datasets

	DTI	DTII	DTIII	DTIV	DTV	DTVI	DTVII	DTVIII	DTIX	DTX
epoch = 15	97.20%	98.40%	95.30%	94.23%	91.64%	95.45%	95.83%	96.54%	88.56%	87.65%
epoch = 20	99.40%	99.50%	99.32%	98.95%	98.46%	99.00%	99.25%	99.25%	91.60%	90.30%
epoch = 25	98.60%	99.12%	95.80%	99.30%	95.81%	96.75%	98.25%	99.02%	91.40%	89.33%

4.5 Experimental results of different models

In order to verify the accuracy and effectiveness of the convolutional neural network model established in this article, the results of this method were compared with traditional support vector machines (SVM) [54], convolutional neural networks based on fast Fourier transform spectra (FFTCNN) [55], convolutional neural networks based on continuous wavelet transform time–frequency maps (CWTCNN) [56], and one-dimensional convolutional neural networks based on wide convolutional kernels (WDCNN) [35]. Using the same dataset and epoch = 20, in order to prevent unexpected phenomena, each model needs to be trained 5 times and the most stable value selected.

As can be seen from Fig. 19, the accuracy of the adaptive feature extraction method of the convolutional neural network is higher than that of the three traditional intelligent diagnostic methods under the conditions of different load domains. This is mainly because the adaptability of the manually designed feature extraction is poor, and the recognition rate of SVM under different load conditions is limited by the nonlinear expression ability. Although FFTCNN and CWTCNN have strong fitting ability, their generalization ability is low, and their accuracy under different loads still needs further improvement. The automatic extraction and classification of features by WGPGCNN is directly end-to-end,

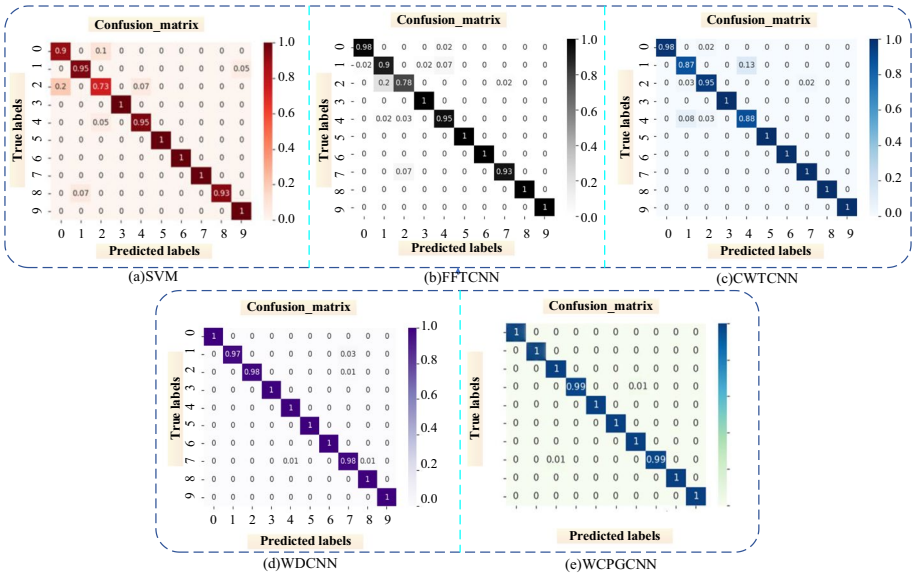
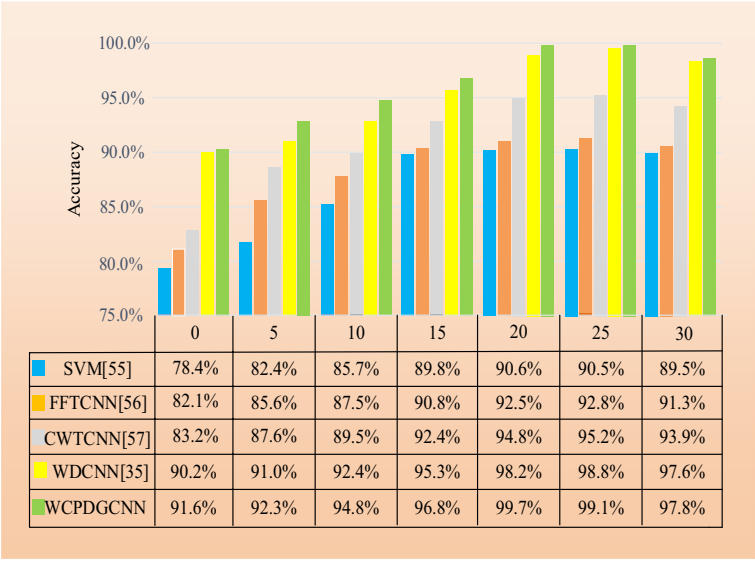


Fig. 19 Confusion matrix of different models

Table 8 Accuracy of different model test sets

Model	SVM[54]	FFTCNN[55]	CWTCNN[56]	WDCNN[35]	WCPGCNNN[the present model]
Accuracy	90.55%	92.47%	94.83%	98.20%	99.70%



	0	5	10	15	20	25	30
SVM[55]	78.4%	82.4%	85.7%	89.8%	90.6%	90.5%	89.5%
FFTCNN[56]	82.1%	85.6%	87.5%	90.8%	92.5%	92.8%	91.3%
CWTCNN[57]	83.2%	87.6%	89.5%	92.4%	94.8%	95.2%	93.9%
WDCNN[35]	90.2%	91.0%	92.4%	95.3%	98.2%	98.8%	97.6%
WCPDGCNN	91.6%	92.3%	94.8%	96.8%	99.7%	99.1%	97.8%

Fig. 20 Comparison of different models under different cycle times

without the need for manual intervention. It relies entirely on one-dimensional convolutional neural networks. Compared with the three traditional methods mentioned above, WGPGCNN eliminates the process of feature extraction without the need for FFT transformation, and better preserves hidden features in the samples. Compared with WDCNN, each convolution kernel of WGPGCNN is wider and has double convolution layers, which can get a larger acceptance region, make the features it gets more global, and have more effect on restraining overfitting. The accuracy of the above models is shown in Table 8.

From Fig. 20, it can be clearly seen that the accuracy of WCPGCNN is higher at the initial epoch. This reflects the excellent classification performance of the WCPG network. With the increase of epoch, the accuracy of the WCPGCNN model established in this article exceeds that of the other five models. The above results indicate that the model established in this article has good convergence and the recognition ability of the entire model has been improved.

Although the accuracy of the model can be improved by stacking parameters and designing more complex structure, it will lead to more computation and slower training speed. Therefore, how well the model predicts is not the only consideration, training time will lead to higher costs. The convolutional neural network is split into CPU and GPU, and GPU computing is about 40 percent faster than CPU. Considering that not every computer can be calculated using the GPU method, all subsequent experiments in this article are carried out using the CPU. The network architecture also has a big impact on the speed of the models, and the

lightweight convolutional neural network models are quick to diagnose because of their simple structure.

Figure 21 shows the speed of the training using the length of time the program ran. In order to prove that the convolution iteration times of this model are fast, a comparative experiment is carried out. A self-coding fault diagnosis model based on one-dimensional residual convolution 1DRCAE [57], a wide convolution kernel neural network WDCNN [35] fault diagnosis model and a Support vector machine SVM [54] are added to the comparison test.

As can be seen from the figure, in the overall comparison, 1DRCAE has the longest diagnostic time because of the presence of self-encoders, with the largest computational load among all models. The WCPDGCNN model established in this paper can be seen in the overall comparison with a very fast diagnostic speed.

4.6 Test results under white noise

In the actual operation of bearings, there is usually external interference noise. The noise of the diagnostic signal is generally additive Gaussian white noise. This scene is more consistent with the situation in real industrial production. Because the noise changes greatly, we cannot get all marked training samples in different noise environments. First, the composite signals with different signal to noise ratios are processed with additive Gaussian white noise. The signal to noise ratio (SNR) is the standard to evaluate the noise intensity. The definition of signal-to-noise ratio is as follows.

$$SNR_{dB} = \log_{10}(\frac{P_{signal}}{P_{noise}}) \tag{15}$$

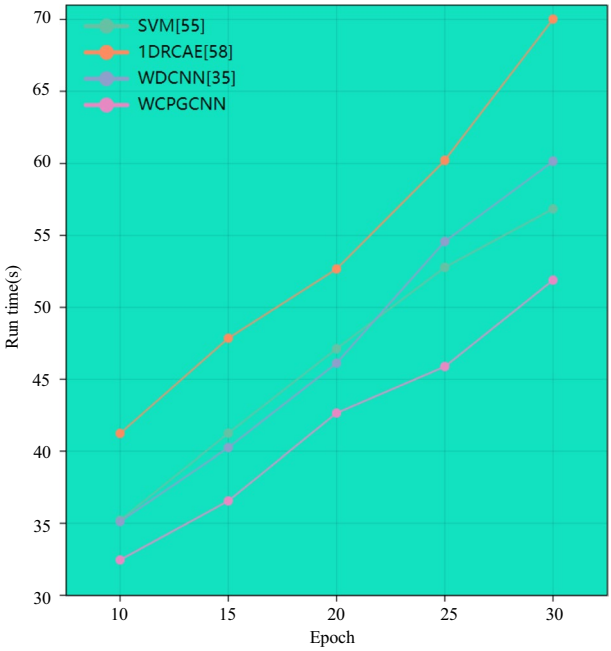


Fig. 21 Comparison of program running time under different training times

where, SNR_{dB} is the signal-to-noise ratio, usually expressed in decibels (dB), P_{signal} represents the power of the normal signal, P_{noise} represents the power of noise.

According to Eq. (11), the larger the noise, the lower the signal-to-noise ratio. The signal-to-noise ratio of a composite noise signal is 0 dB, and when the signal-to-noise energy is the same, the signal-to-noise ratio is 0. Therefore, in this experiment, Gaussian additive white noise with signal to noise ratio of 0-10db is added to the training set to detect the noise immunity of WGPGCNN. In Fig. 22, Gaussian white noise with different signal-to-noise ratios is added to the original signal of the normal bearing.

The results of the WGPGCNN model proposed in this article under different noise environments are shown in Table 9. It can be seen that the wider the first layer of kernel, the higher the accuracy. For example, when the kernel size is 8, the average accuracy is only 58.38%, while when the kernel size increases to 64, the accuracy surges to 90.18%. In addition, the extraction results are best when the kernel size is 64 and 72, rather than when the maximum size is 80, which also proves that larger kernel sizes are not suitable for extracting local features. As shown in Fig. 22, the accuracy comparison under different noise environments is shown.

From Fig. 23, it can be seen that both the size of the kernel and the size of the noise have an impact on the accuracy of the model. When the kernel size is too small, the larger the noise, the lower the accuracy. As the kernel size continues to increase, the impact of the noise on it gradually decreases. When $S=64$ is reached, the impact is minimal, and the diagnostic effect of the model decreases when the kernel size continues to increase.

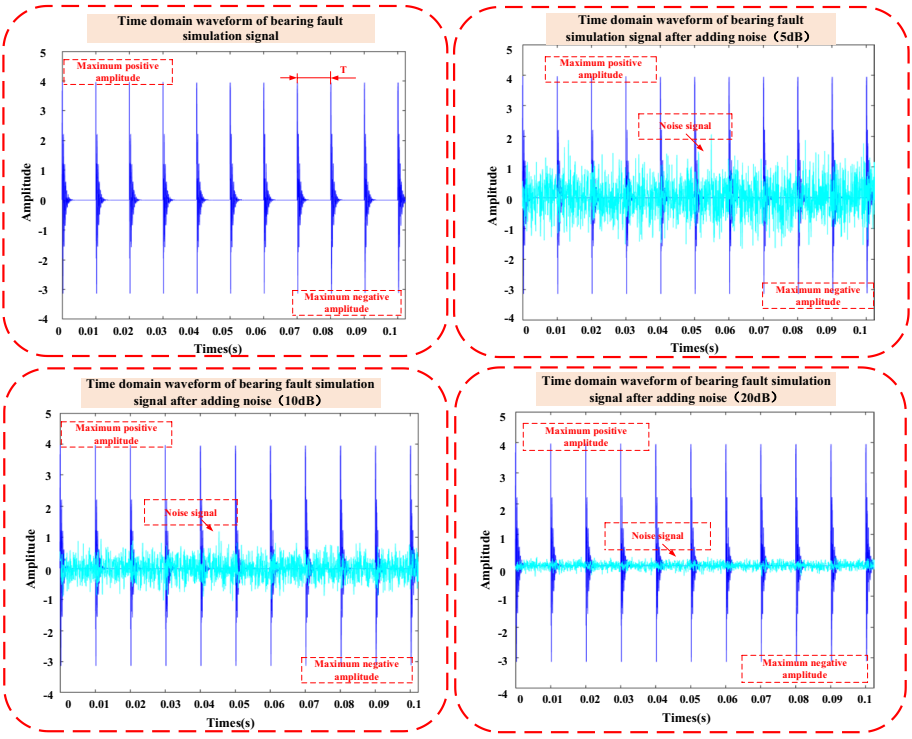


Fig. 22 Composite noise signal diagram of normal bearings with different signal-to-noise ratios

Table 9 Results of WGPGCNN under different noise environments

Kernel size S	SNR/dB					
	0	2	4	6	8	10
8	58.38%	67.01%	78.65%	86.65%	93.44%	97.31%
16	68.25%	80.45%	88.37%	93.20%	96.82%	98.62%
24	80.21%	86.53%	93.47%	96.04%	97.52%	98.89%
32	82.85%	90.20%	94.06%	97.21%	98.58%	99.28%
40	84.01%	92.04%	95.69%	98.25%	99.31%	99.55%
48	85.91%	91.34%	96.84%	99.22%	99.55%	99.68%
56	88.04%	93.05%	98.07%	99.28%	99.75%	99.71%
64	90.18%	95.08%	98.65%	99.34%	99.79%	99.83%
72	88.91%	94.56%	98.44%	99.25%	99.73%	99.83%
80	86.31%	93.08%	98.63%	99.08%	99.52%	99.72%
MAX	90.18%	95.08%	98.65%	99.34%	99.79%	99.83%

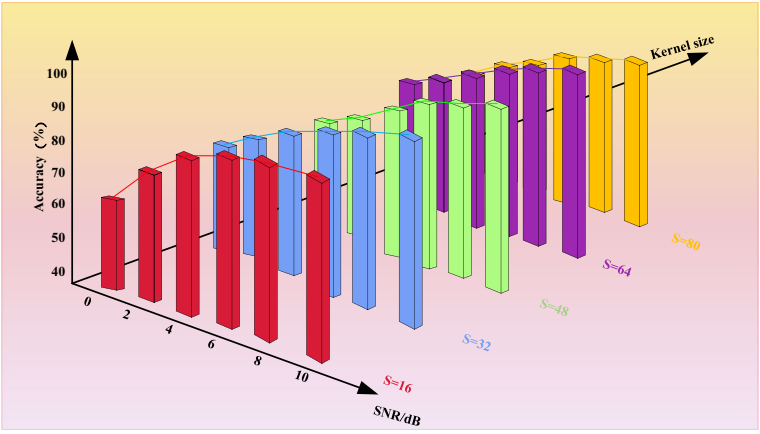


Fig. 23 Accuracy under different noise environments

Based on the above observations, it can be inferred that when the core size is small, it is susceptible to high-frequency noise interference, and low-frequency features are difficult to capture; When the kernel size is large, the resolution in the time domain decreases, which is prone to missing some detailed features, resulting in a decrease in accuracy instead of an increase. Therefore, selecting the appropriate size of the first layer kernel has a significant impact on the noise resistance performance of the model.

4.7 Network visualization

Usually, CNN is considered a blind box because some of its internal operating mechanisms cannot be clearly captured and are difficult to understand. In this article, we use the

activation function of visual neural networks to explore the internal operating process of the WGPGCNN model.

Firstly, in order to better understand which types of features were extracted by deconvolution kernels, we drew the filter kernels learned by WGPGCNN and the frequency domain features of FFT transform. Figure 24a shows the time-domain waveform of the first layer wide convolutional kernel, from which we can also see that there are significant differences in the data features learned by different convolutional kernels in the first layer; Some convolutional kernels have shapes similar to sine functions, such as the 2nd, 4th, 7th, and 8th convolutional kernels (marked in the figure), and exhibit large periods. Therefore, these convolutional kernels extract low-frequency features of the input data, while high-frequency features are filtered out by them.

Figure 24b shows the frequency domain representation of the convolutional kernel for this layer (the convolutional kernel data is obtained through fast Fourier transform). From its frequency domain expression, we can see that, the labeled convolutional kernels learn features that are medium to low frequency, and the frequency bands learned by different convolutional kernels are different. For a single convolutional kernel, it can adaptively extract sensitive frequency bands between faults and filter out data features from other frequency bands besides its own learning, making it more targeted. Compared with traditional filtering methods, using wide convolutional kernels not only reduces human intervention in the filtering process, but also has a better adaptive filtering process. It can also extract input frequency band features more targeted, providing higher quality data for subsequent feature extraction.

In order to see clearly the capability of WCPDGCNN model for bearing fault diagnosis, the dimension reduction operation is performed and the image is output, as shown in Fig. 25, which shows the visualization process at different times in CWRU and MFPT datasets. From Fig. 25a, we can see that the original input signal has the biggest entropy and the highest degree of confusion. In the process of diagnosis, the dimension of the model was cut down at a certain moment, as shown in Fig. 25b. It can be seen that the model has shown some classification ability after a period of convolution operation. The same category of fault data began to converge, and some fault data has even been completely separated. Figure 25c shows a reduced dimension map of the model final classification, with fault data separated into its own groups as shown in the figure. Many of the same types of fault data have been overlapped, and the ten types are clustered and dispersed. At the same

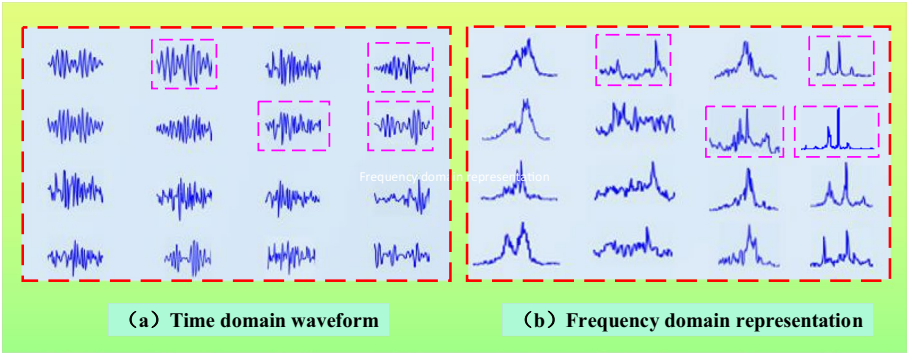


Fig. 24 First layer wide convolution kernel visualization

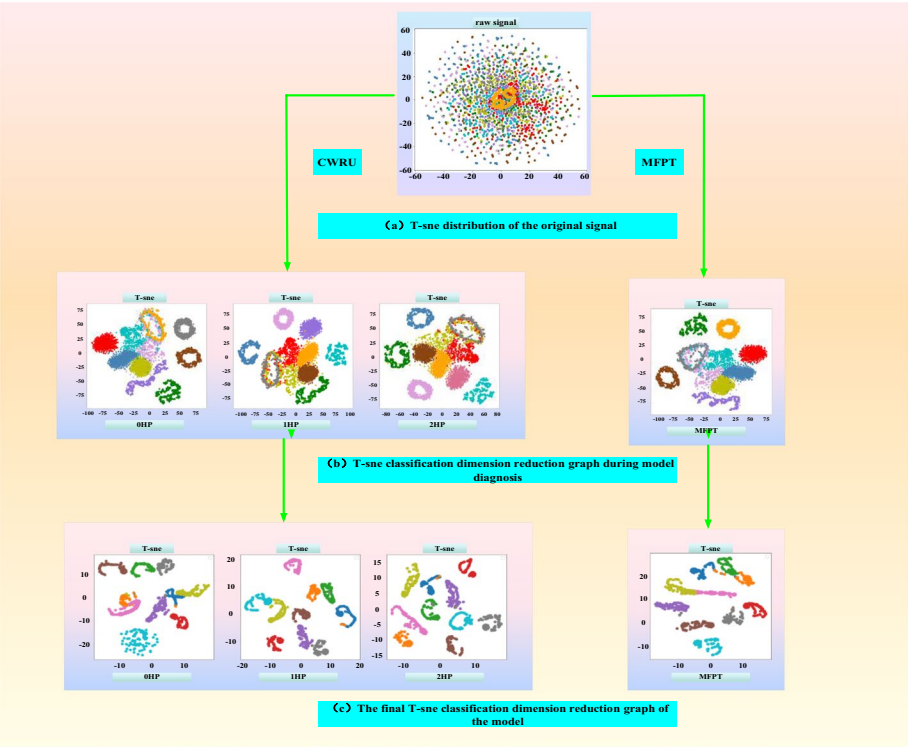


Fig. 25 Visualization of Model Recognition Process

time, the classification effect of MFPT data set is worse than that of CWRU data set. The reason for this may be due to different fault classification scenarios.

5 Conclusions

In this paper, a new deep convolution neural Network tomography diagnosis method is proposed to solve the problems of traditional fault diagnosis, such as inefficient and accurate identification, greater dependence on human prior knowledge, and single fault diagnosis method model. A convolutional neural network model that can directly act on the original time-domain signal was constructed and validated on the standard bearing fault dataset, achieving excellent results. The main achievements of this article are as follows.

(1) This article proposes a new model for fault diagnosis of rolling bearings—using convolutional pooling group (CPG) for feature extraction of data, while expanding the double-layer convolutional kernel to obtain a larger receptive field. A wide convolutional kernel convolutional neural network WCPGCNN model based on CPG network architecture is obtained, which has good performance in accuracy, noise resistance, and timeliness. At the same time, it also has good cross dataset high generalization ability.

(2) We conducted some control experiments. By studying the impact of different batch sizes, sample sizes, and iteration times on the accuracy of the model under the same dataset, as well as the diagnostic performance of different models under the same dataset and multiple data types. Through comparative experiments and analysis, it can be seen that the accuracy of the model established in this paper is relatively high during the initial iteration. As the number of iterations increases, the accuracy of the model established in this paper exceeds that of the other four models. The model established in this article has a reduced diagnostic ability in extracting fault datasets from the 2HP terminal. However, it achieved the highest accuracy on the 0HP base, increasing by 3% compared to conventional diagnostic models, reaching 99.50%. SVM, FFTCNN, CWTCNN, and WDCNN, as more traditional machine learning methods, have good self-diagnostic performance, but their accuracy is slightly lower than that of the models built in this article. At the same time, the speed of this model is also faster than the other four models in terms of running speed. In summary, the model performs well in terms of accuracy and timeliness.

(3) Considering the impact of environmental noise, this article also simulated the accuracy of the model under different noise conditions. The model added noise during training, so that the trained model can maintain high recognition rate when the test signal changes. In summary, the model also exhibits good noise resistance under different operating conditions. The experimental results verify that the multi pooling group feature processing with large receptive field characteristics has good performance, and deep learning has better diagnostic performance than machine learning.

Acknowledgements The authors would like to acknowledge the financial support of the National Natural Science Foundation of China (NSFC) (Grant No. 52375113); Natural Science Foundation of Liaoning Province of China (Grant No. 2022-MS-298); Shenyang Youth Science and Technology Innovation Talent Fund (Grant No. RC230309); Scientific Research Fund of Liaoning Education Department (Grant No. LJKMZ20220531).

Data Availability Data available on request from the authors.

Declarations

Conflict of Interest The authors declare that they have no conflict of interest.

References

1. Pan WJ, Ling LY, Qu HY, Wang MH (2022) Coupling dynamic behavior of aero-engine rotor system caused by rolling, pitching and yawing maneuver loads. Appl Math Model 102:726–747
2. Pan WJ, Ling LY, Qu HY (2023) Dynamic modeling and response analysis of rub-impact rotor system with squeeze film damper under maneuvering load. Appl Math Model 114:544–582
3. Pan WJ, Qu HY, Sun YH (2023) A deep convolutional neural network model with two-stream feature fusion and cross-load adaptive characteristics for fault diagnosis. Meas Sci Tecgnol 34:1–23
4. Pan WJ, Sun YH, Cheng RR, Cao SM (2023) A SENet-TSCNN model developed for fault diagnosis considering squeeze-excitation networks and two-stream feature fusion. Meas Sci Technol 34:1–21
5. Cerrada M, Sanchez RV, Li C, Pacheco F, Cabrera D, Oliverira JVD, Vasquez RE (2018) A review on data-driven fault severity assessment in rolling bearings. Mech Syst Signal 99:169–196
6. Hoang DT, Kang HJ (2019) A survey on deep learning based bearing fault diagnosis. Neurocomputing 335:327–335

7. Rauber TW, Loca A, Boldt F, Rodrgues A, Varejo F (2021) An experimental methodology to evaluate machine learning methods for fault diagnosis based on vibration signals. Expert Syst Appl 167:1–21
8. Aggarwal K, Mijwil MM, Sonia SH, Abdulrhman (2022) Has the future started? The current growth of artificial intelligence, machine learning, and deep learning. Iraqi Journal for Computer Science and Mathematics 3:115–123
9. Rikam LE, Bitjoka L, Nketsa A (2022) Quaternion fourier transform spectral analysis of electrical currents for bearing faults detection and diagnosis. Mech Syst Signal Process 68:1–18
10. Wang D, Zhao Y, Yi C (2018) Sparsity guided empirical wavelet transform for fault diagnosis of rolling element bearings. Mech Syst Signal Process 101:292–308
11. Cao X, Wang Y, Chen B (2021) Domain-adaptive intelligence for fault diagnosis based on deep transfer learning from scientific test rigs to industrial applications. Neural Comput Appl 33:83–4499
12. Li H, Tao L, Wu X (2020) A bearing fault diagnosis method based on enhanced singular value decomposition. IEEE T Ind Inform 17:3220–3230
13. Kumar HS, Manjunath SH (2022) Use of empirical mode decomposition and K- nearest neighbour classifier for rolling element bearing fault diagnosis. Materials Today: Proceedings 52:796–801
14. Yassine T, Billel B, Sidahmed L, Mohamed O (2022) FPGA implementation of a bearing fault Classification System Based on an Envelope Analysis and Artificial Neural Network. Arab J Sci Eng 47:3955–13977
15. Xing TT, Zeng Y, Meng Z, Guo XL (2020) A fault diagnosis method of rolling bearing based on VMD Tsallis entropy and FCM clustering. Multimed Tools Appl 79:30069–30085
16. Wang B, Zhang X, Xing S, Sun C, Chen X (2021) Sparse representation theory for support vector machine kernel function selection and its application in high-speed bearing fault diagnosis. ISA Trans 118:207–218
17. Han T, Zhang L, Yin Z, Tan A (2021) Rolling bearing fault diagnosis with combined convolutional neural networks and support vector machine. Measurement 177:1–13
18. Sinitsin V, Ibryaeva O, Sakovskaya V (2022) Intelligent bearing fault diagnosis method combining mixed input and hybrid CNN-MLP model. Mech Syst Signal Process 109:1–24
19. Ma M, Sun C, Chen X (2018) Deep coupling autoencoder for dault diagnosis with multimodal sensory data. IEEE T Ind Inform 14:1137–1145
20. Ayas S, Ayas MS (2022) A novel bearing fault diagnosis method using deep residual learning network. Multimed Tools Appl 81:22407–22423
21. Sun HB, and Fan YG (2023) Fault diagnosis of rolling bearings based on CNN and LSTM networks under mixed load and noise. Multimed Tools Appl https://doi.org/10.1007/s11042-023-15325-w
22. Oh H, Jung JH, Jeon BC (2017) Scalable and unsupervised feature engineering using vibration-imaging and deep learning for rotor system diagnosis. IEEE T Ind Electron 65:3539–3549
23. Abid A, Khan MT, Khan MS (2020) Multidomain features-based ga optimized artificial immune system for bearing fault detection. IEEE Trans Syst Man Cybern-Syst 50:348–359
24. Khorram A, Khalooei M (2019) Intelligent bearing fault diagnosis with convolutional long-short-term-memory recurrent neural network. Cornell Univ Electric Eng Syst Sci 7:1–13
25. An ZH, Li SM, Wang JX, Jiang XX (2019) A novel bearing intelligent fault diagnosis framework under time-varying working conditions using recurrent neural network ISA T 100:155–170
26. Peng YZ, Wang Y, Shao YM (2022) A novel bearing imbalance fault-diagnosis method based on a Wasserstein conditional generative adversarial network. Measurement 192:1–9
27. Liu SW, Jiang HK, Wu ZH, Li XQ (2022) Data synthesis using deep feature enhanced generative adversarial networks for rolling bearing imbalanced fault diagnosis. Mech Syst Signal Pr 163:1–20
28. Liu Y, Jiang H, Wang Y, Wu Z, Liu S (2022) A conditional variational autoencoding generative adversarial networks with self-modulation for rolling bearing fault diagnosis. Measurement 192:1–14
29. Yang ZH, Cen J, Liu X, Xiong JB, Chen HH (2022) Research on bearing fault diagnosis method based on transformer neural network Meas. Sci Technol 33:1–12
30. Waibel A, Hanazawa T, Hinton G, Shikano K, Lang KJ (1989) Phoneme recognition using time-delay neural networks IEEE T Signal Proces 37:328–339
31. Zhang W (1988) Shift-invariant pattern recognition neural network and its optical architecture. In Proceedings of annual conference of the Japan Society of Applied Physics 33:2147–2151
32. LeCun Y, Boser B, Denker JS, Henderson D, Howard RE, Hubbard W, Jackel LD (1989) Backpropagation applied to handwritten zip code recognition. Neural Comput 1:541–551
33. Janssens O, Slavkovikj V, Vervisch B, Stockman K (2016) Convolutional neural network based fault detection for rotating machinery. J Sound Vib 77:331–345
34. Özgür G, Eyüp Ç (2022) A novel deep learning approach for intelligent fault diagnosis applications based on time-frequency images. Neural Comput Appl 34:4803–4812

35. Zhang W, Li C, Peng G (2018) A deep convolutional neural network with new training methods for bearing fault diagnosis under noisy environment and different working load. Mech Syst Signal Process 100:439–453
36. Zhang JQ, Sun Y, Guo L, Gao HL, Hong X, Song HL (2019) A new bearing fault diagnosis method based on modified convolutional neural networks. Chinese J Aeronaut 33:439–447
37. Wang Q, Yang C, Wan H (2021) Bearing fault diagnosis based on optimized variational mode decomposition and 1D convolutional neural networks. Meas Sci Tecgnol 32:1–17
38. Levent E, Turker I, Serkan K (2019) A generic intelligent bearing fault diagnosis system using compact adaptive 1D CNN classifier. J Signal Process Sys 91:179–189
39. Gao S, Pei Z, Zhang Y, Li T (2021) Bearing fault diagnosis based on adaptive convolutional neural network with Nesterov momentum. IEEE Sens J 99:1–9
40. Wang FA, Jiang HK, Shao H, Duan WJ, Wu SP (2017) An adaptive deep convolutional neural network for rolling bearing fault diagnosis. Meas Sci Tecgnol 28:1–17
41. Xu K, Li SM, Wang JR, An ZH, Xin Y (2020) A novel adaptive and fast deep convolutional neural network for bearing fault diagnosis under different working conditions P. I Mech Eng D-J Aut 234:1167–1182
42. Zhang X, Liu S, Li L (2021) Multiscale holospectrum convolutional neural network-based fault diagnosis of rolling bearings with variable operating conditions. Meas Sci Tecgnol 32:1–12
43. Liu Y, Yan XS, Zhang CA, Liu W (2019) An ensemble convolutional neural networks for bearing fault diagnosis using. Multi-Sensor data Sensors-Basel 19:1–20
44. Jin J, Xu Z, Li C (2022) Rolling bearing fault diagnosis based on convolutional bidirectional short term memory network and chaos theory. J Sound Vib 41:160–169
45. Li YY, Hou LY, Tang M, Sun QC, Chen JH et al (2022) Prediction of wind turbine blades icing based on feature Selection and 1D-CNN-SBiGRU. Multimed Tools Appl 81:4365–4385
46. Hu Z, Bian J (2022) A deep feature extraction approach for bearing fault diagnosis based on multiscale convolutional autoencoder and generative adversarial networks. Meas Sci Tecgnol 33:1–12
47. Wang H, Liu Z, Peng D (2021) Feature-level attention-guided multitask CNN for fault diagnosis and working conditions identification of rolling bearing. IEEE T Neu Ne Lear 99:1–13
48. Huang DT, Kang HJ (2018) Rolling element bearing fault diagnosis using convolutional neural network and vibration image. Cogn Sys Res 53:42–50
49. Jia F, Lei YG, Lu N, Xing SB (2018) Deep normalized convolutional neural network for imbalanced fault classification of machinery and its understanding via visualization. Mech Syst Signal Process 110:349–367
50. Cooley JW, Tukey (1965) An algorithm for the machine calculation of complex fourier series. Math Comput 19:297–301
51. Simonyan K, and Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. Comput Sci 10:1–13
52. Breloy A, Kumar S, Sun Y (2021) Majorization-Minimization on the stiefel manifold with application to robust sparse PCA. IEEE T Signal Process 69:1507–1520
53. Henrique MB, Ramirez P, Nunan Z (2021) Improving barnes-hut t-SNE algorithmin modern GPU architectures with random forest KNN and simulated wide-warp. ACM J Emerg TechCom 53:1–26
54. Li R, Zhuang L, Li Y (2021) Intelligent bearing fault diagnosis based on scaled ramanujan filter banks in noisy environments. IEEE T Instrum Meas 70:1–13
55. Shan Y, Zhou JZ, Jiang W (2019) A fault diagnosis method for rotating machinery based on improved variational mode decomposition and hybrid artificial sheep algorithm. Meas Sci Tecgnol 30:1–16
56. Chen R, Zhou J, Hu X, Han X, Zhu S (2021) Fault diagnosis method of rotating machinery based on deep Q learning and continuous wavelet transform. J Vib Eng Technol 34:1092–1100
57. Yu J, Zhou X (2020) One-Dimensional Residual Convolutional Autoencoder Based Feature Learning for Gearbox Fault Diagnosis. IEEE T Ind Inform 16:6348–6358