

## Research Article

# Fault Diagnosis of Rotating Machinery Based on Evolutionary Convolutional Neural Network

Yihao Bai , Weidong Cheng, Weigang Wen , and Yang Liu

*School of Mechanical, Electronic and Control Engineering, Beijing Jiaotong University, Beijing 100044, China*

Correspondence should be addressed to Weigang Wen; [wgwen@bjtu.edu.cn](mailto:wgwen@bjtu.edu.cn)

Received 28 September 2022; Revised 17 October 2022; Accepted 19 October 2022; Published 29 October 2022

Academic Editor: Yi Qin

Copyright © 2022 Yihao Bai et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper proposes a fault diagnosis method for rotating machinery based on evolutionary convolutional neural network (ECNN). With the time-frequency images as the network input, with the help of the global optimization ability of the genetic algorithm, the structure of the convolutional neural network can evolve autonomously, and the adaptive configuration of the structural hyperparameters for the target task is realized. In this paper, the proposed method is verified by the measured signal of the planetary gearbox. The results show that the proposed method is helpful to obtain a convolutional neural network structure with better performance and achieve higher fault diagnosis accuracy.

## 1. Introduction

Rotating machinery is widely used in various fields of industrial engineering, and its fault diagnosis is particularly important. In practical industrial applications, mechanical equipment is often in a state of variable speed operation, and its vibration signal presents a nonstationary state. Time-frequency transformation is an important means to analyze nonstationary signals. By transforming one-dimensional vibration signals into two-dimensional images, it can show more abundant time-frequency characteristics of signals. How to establish the connection between time-frequency images and fault categories is a hot issue in research [1].

In recent years, there are many remarkable results achieved in image recognition and other aspects using convolutional neural network (CNN). CNN has many variants with different structures for different targets, which greatly improve the network performance in different ways [2–6]. However, the design of network structures is often limited by the researchers' empirical knowledge of the disciplines involved in the target task.

Facing the problem of mechanical fault diagnosis, early researchers designed the network structure by trial and error to realize the classification of time-frequency images of vibration signals [7–12]. However, the design of the structure

often lacks theoretical support and is rarely mentioned in the research. Some scholars draw on the network model of CNN that has outstanding performance in image recognition tasks and apply it to fault diagnosis. Li et al. [13] modified some parameters based on LetNet-5 and tried to find the network model suitable for bearing fault diagnosis. Verstraete et al. [14] designed the network structure based on the idea of VGG which can realize bearing fault diagnosis well. Hoang and Kang [15] designed the network structure based on LiftingNet which has better robustness to noise in signals. Wen et al. [16] constructed a hierarchical CNN structure based on LetNet-5, which improved the accuracy of fault diagnosis. Furthermore, Qian et al. [17] proposed a new deep transfer learning network based on convolutional autoencoder (CAE-DTLN) to realize mechanical fault diagnosis in target domain without labelled data. Qin et al. [18] proposed a novel domain adaptation mechanism named intermediate distribution alignment (IDA) to solve the problems of poor convergence speed and robustness of existing domain adaptation mechanisms in the training process. The above studies have fully demonstrated the feasibility and application prospects of migrating mature CNN models in the field of image recognition to mechanical fault diagnosis, but the structural adjustment of network models still relies on researchers' continuous attempts. This is because the time-frequency

image of the vibration signal is fundamentally different from the traditional image. The vibration signal often contains a large number of periodic harmonic components, so the time-frequency image often contains a large number of repeated and similar components. It is hard to extract the features of the time-frequency images and often needs a more complex network structure to distinguish the images.

In the field of computer science, some scholars have proposed the combination of CNN and evolutionary algorithms to realize the automatic design of network structure. Bochinski et al. [19] proposed an evolutionary algorithm-based framework to automatically optimize the CNN structure by means of hyperparameters in computer vision tasks. Essiet et al. [20] proposed an ensemble of evolutionary algorithms which combined with CNN for gas detection. Zhang et al. [21] built a multilevel convolutional neural network (ML-CNN) via a hyperparameter importance-based evolutionary strategy for lung nodule classification.

However, these kinds of research results have not been applied to the fault diagnosis of rotating machinery. In response to this situation, the study encodes the network structure of CNN and realizes the adaptive evolution of the CNN network structure for fault diagnosis through supervised learning based on genetic algorithm. The rest of this article is organized as follows. Section 2 describes the basic principles of the algorithm. Section 3 describes the experimental environment in detail and analyzes the experimental results. Finally, Section 4 gives the phased conclusions and prospects.

## 2. Algorithm Principle

**2.1. CNN.** CNN is a feedforward neural networks that includes convolution computation and has a deep structure. It is one of the representative algorithms of deep learning [22, 23]. An example CNN structure is shown in Figure 1.

**2.1.1. Convolutional Layer.** The function of the convolutional layer is to extract features of the input data. It contains multiple convolution kernels inside. The neurons in the convolution kernel are connected to multiple neurons in an area of the input. The size of the area depends on the size of the convolution kernel. This area is called the receptive field. When the convolution kernel is working, it will regularly scan the input features and convolve with the input in the receptive field to obtain the output. Figure 2 shows a schematic diagram of a  $3 \times 3$  convolution kernel and the input convolution operation.

**2.1.2. Pooling Layer.** The function of the pooling layer is to downsample the input data. Through the preset pooling function, the feature statistics of the pooled area in the input are used as the output, which reduces the amount of calculation and suppresses overfitting to a certain extent. Figure 3 shows a schematic diagram of the pooling operation.

**2.1.3. Fully Connected Layer.** The function of the fully connected layer is to classify the input feature images. The

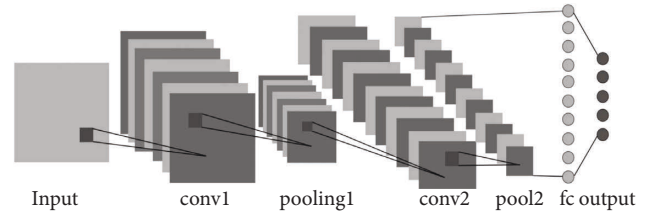


FIGURE 1: Schematic diagram of convolutional neural network structure.

input feature image loses its spatial topology in the fully connected layer and is expanded into vector output after activation function.

**2.2. Genetic Algorithm.** Genetic algorithm is a kind of evolutionary algorithm proposed by John H. Holland [24]. The algorithm is an adaptive global search optimization method which originated from the computer simulation research of biological evolution and genetic system.

The flowchart of the genetic algorithm is shown in Figure 4. For a given initial individual, an initial population is generated as the first generation by mutation. Taking the individuals in the initial population as the parent generation, several new individuals are generated as offspring through three basic genetic operators: selection operator, crossover operator, and mutation operator. The fitness of the parent and the offspring is evaluated, and the offspring with high fitness replaces the parent with low fitness and joins the population to obtain the second-generation population. The standard of fitness evaluation depends on the specific optimization problem. By repeating this process, the individuals with the best fitness in several generations are retained in the population and the individuals with poor fitness are eliminated. When the termination condition is met, usually the number of iterations reaches a preset value or there is no individual with better fitness in successive generations, the algorithm terminates, and the individual with the best fitness in the population at this time is output as the optimal solution. Key operations in the process are explained later.

**2.2.1. Fitness.** Fitness is a measure of an individual's dominance in surviving in a population. It has different definitions for different target tasks, and its definition is directly related to the algorithm's convergence speed. It is usually the only criterion for judging the quality of the solution when the algorithm performs a global search.

### 2.2.2. Three Basic Operators

**(1) Selection Operator.** The selection operator is responsible for selecting a suitable individual from the current population as the parent, usually with fitness as the selection criterion, that is, the higher the fitness of an individual, the greater the probability of being selected. Suppose the population number is  $M$ , and the fitness of the  $i$  individual is  $f_i$ ; then, its probability of being selected is

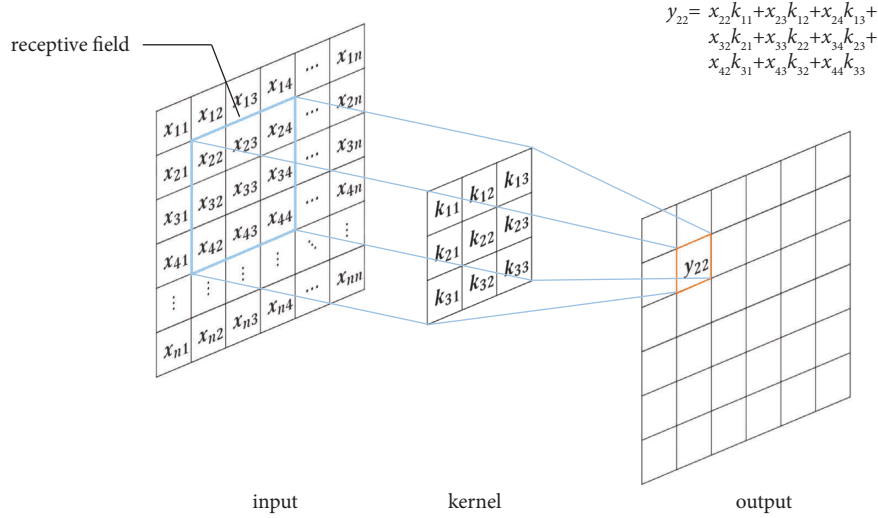


FIGURE 2: Schematic diagram of convolution operation.

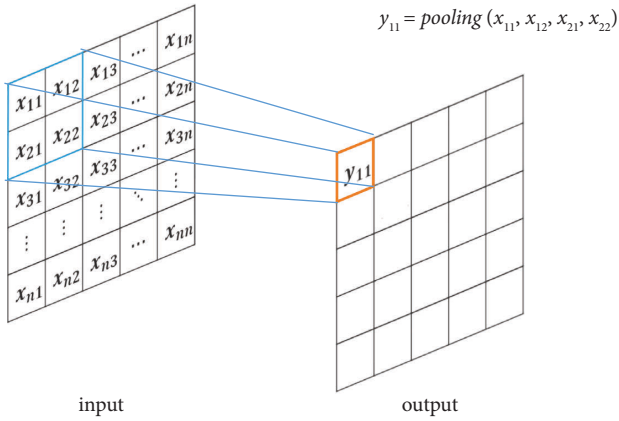


FIGURE 3: Schematic diagram of pooling operation.

$$P_i = \frac{f_i}{\sum_{k=1}^M f_k} \# \quad (1)$$

(2) *Crossover Operator.* The crossover operator is responsible for generating a new individual from the selected two parent individuals by means of crossover combination. In practical applications, the most common one is the single-point crossover operator, which randomly selects a crossover position among the two parents and then crosses the parents at that position. The process is shown in Figure 5.

(3) *Mutation Operator.* The mutation operator is responsible for generating new individuals by mutating the selected individuals, and its purpose is to prevent the algorithm from falling into the optimal solution during the optimization process. The process is shown in Figure 6.

2.3. *Evolutionary Convolutional Neural Networks.* This paper proposes an evolutionary convolutional neural network (ECNN) that combines genetic algorithm and CNN. The

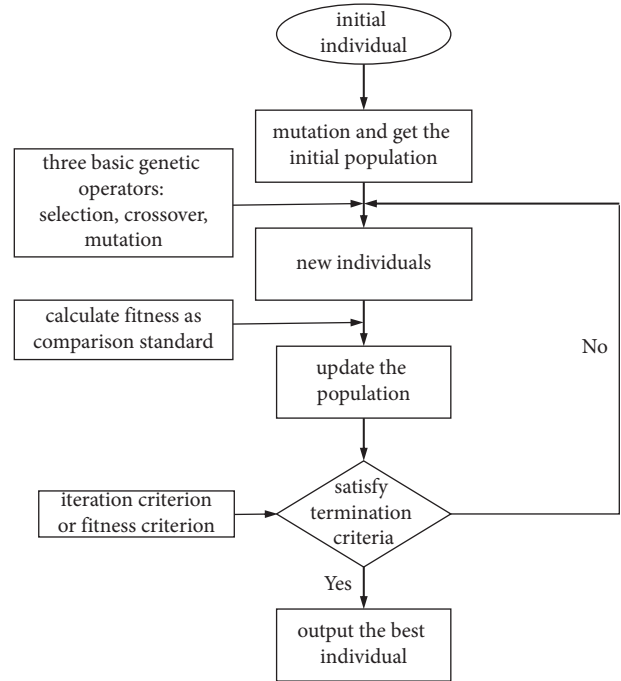


FIGURE 4: Genetic algorithm flowchart.

genetic algorithm is used to optimize the structure of a given network and find the optimal solution of the network structure when facing the problem of gearbox fault diagnosis. The pseudocode of the algorithm is shown in Figure 7. Subsequent parts of this section describe the key steps in the algorithm.

2.3.1. *Encoding.* Encoding is the process of converting candidate solutions of specific problems into machine codes and establishing the mapping relationship between the solution space and the coding space. There are many types of encoding, such as binary encoding, real vector encoding, general data structure encoding, and so on. Common binary

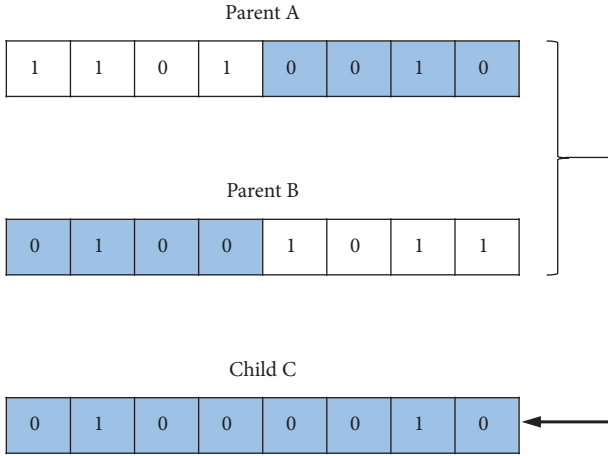


FIGURE 5: Schematic diagram of crossover operator operation.

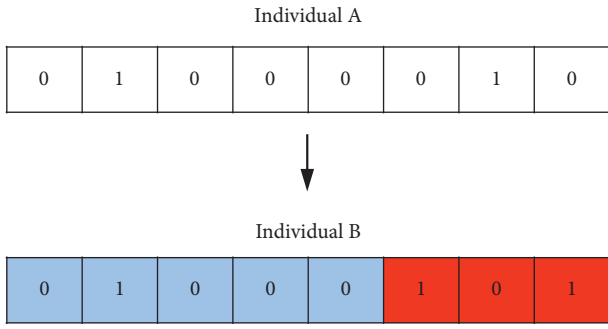


FIGURE 6: Schematic diagram of mutation operator operation.

---

*Algorithm:* Evolutionary Convolutional Neural Network

*Input:* basic CNN architecture net\_1  
population size P, number of generation G, offspring size O

```

mutate net_1, generate net_{1,2,...,P} as initial population P1
for i = 1:G
  for j = 1:O
    select two individuals from Pi as parents
    cross and mutate the parents to get childj
    if fitness childj > net_P
      net_P = childj
    end
  end
  sort Pi by fitness
end
generate Pi+1 = Pi
i = i+1
end
Output: the individual with the highest fitness from PG

```

---

FIGURE 7: Pseudocode of ECNN.

encoding is more complex and less readable. With the help of TensorFlow framework, in this paper, the general data encoding method is adopted. This encoding method stores the single-layer network structure in the convolutional neural network as a separate list object, which contains the type and specific parameters of the network structure of this layer as shown in Table 1. A complete CNN structure can be represented as shown in Figure 8. Taking a list object as a gene, the genetic algorithm is used for the middle layer structure of CNN as an individual.

**2.3.2. Fitness.** For the target, optimize the CNN structure for gearbox fault diagnosis, and the fitness function is defined as the reciprocal of the loss function of the network on the test set of the data:

$$\text{fitness} = \frac{1}{\text{test}_{\text{loss}}}. \quad (2)$$

The higher the fitness of the network structure, the better the performance of the structure on the test set.

**2.3.3. Crossover.** This paper adopts a single-point crossover. Different from the general way, the crossover positions taken by the two parents do not have to be unified, as shown in Figure 9. The advantage is that the number of network layers contained in the generated child network may be different from that of the parents. This avoids the fixed depth of the network, which is one of the keys to affecting the performance of the network.

**2.3.4. Mutation.** To enhance the randomness of individuals, two methods of mutation are adopted in this paper. The first method is layer mutation as shown in Figure 10. Randomly pick a gene, delete it, or add a new gene which deepcopied it. It is important to note that layer mutation does not occur in the pooling layer, since successive multiple pooling can be equated to one larger pooling of a core. Also, to ensure the integrity of the network structure, deletion does not occur when the selected gene is a single fully connected layer or convolutional layer. The second method is parameter mutation. Randomly pick a gene and change its parameter by multiplying by the coefficient 1/2 or 2 as shown in Figure 11. When the selected gene is convolution layer, the changed parameter is the kernel number. When the selected gene is a fully connected layer, the parameter changed is the unit number.

## 3. Experiment

### 3.1. Experimental Design

**3.1.1. Experimental Data.** The experimental data come from the measured vibration signal of the planetary gearbox experimental bench, as shown in Figure 12. There are 5 sun gears under different kinds of health conditions in the experiment, namely, healthy gear, worn gear, root crack gear, tooth broken gear, and tooth missing gear. Under each health condition, 10 independent experiments were carried out on the experimental bench for a total of 50 experiments. Every time the experimental sun gear is disassembled and assembled. In each experiment, the operating speed of the experimental bench is 0–50 Hz variable speed, the effective duration of each experiment is 40 s, the sampling frequency is 48000 Hz, and the sampling length is 1,920,000.

After slicing the vibration signal, discrete wavelet transform is performed to obtain a time-frequency image matrix of 50 \* 50. The matrix is normalized to 0-1 and then used as a sample. Figure 13 shows randomly selected sample images of five different health conditions. 100 samples can be

TABLE 1: Storage content of one list object.

Structure	Parameter	Statement
Conv	Kernel number, kernel size, stride size, padding type	List = [convolutional(kernel = a, kernel_size=(), stride_size=(), padding = ”)]
Pooling	Pooling size, stride size, padding type	List = [pooling(pool_size=(), stride_size=(), padding = ”)]
fc	Unit number	List = [FullConnected(unit = b)]

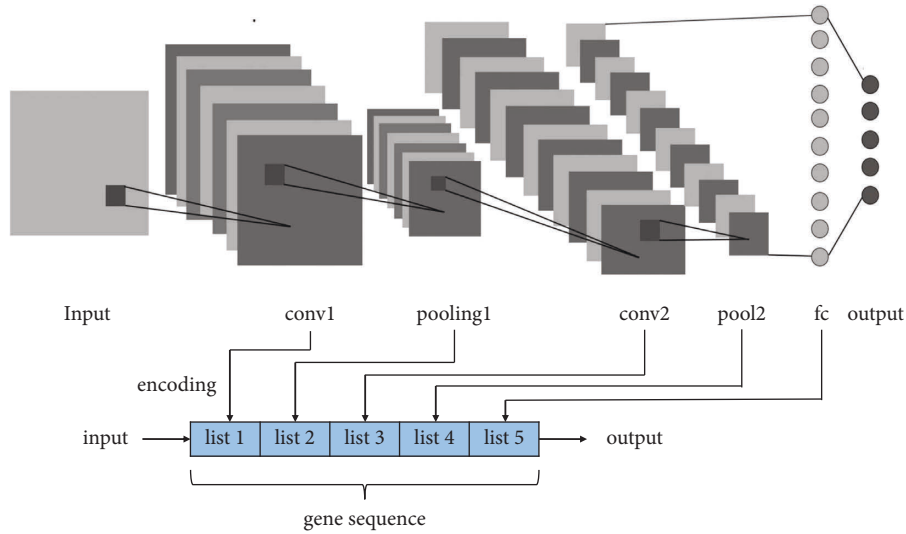


FIGURE 8: Schematic diagram of encoding of ECNN.

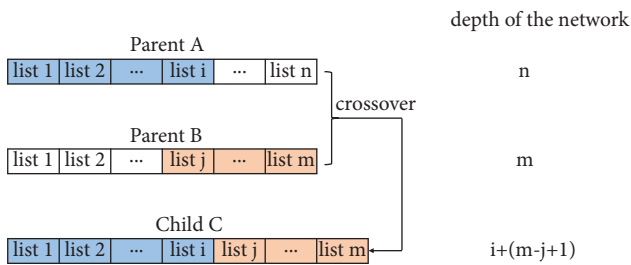


FIGURE 9: Schematic diagram of crossover of ECNN.

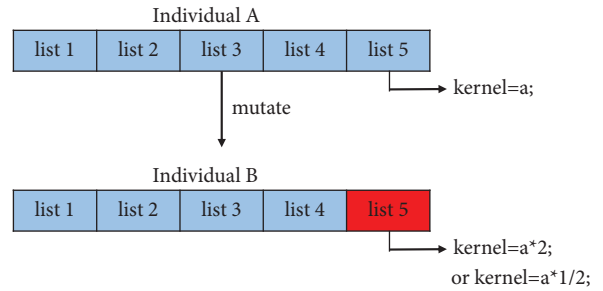


FIGURE 11: Schematic diagram of parameter mutation of ECNN.

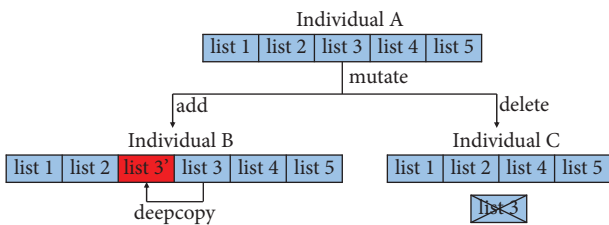


FIGURE 10: Schematic diagram of layer mutation of ECNN.

obtained for each kind of experimental data, 5000 samples can be obtained in 50 experiments, and the samples are labelled according to the health status of the gears. The training set and the test set are randomly divided according to the ratio of 9 : 1.

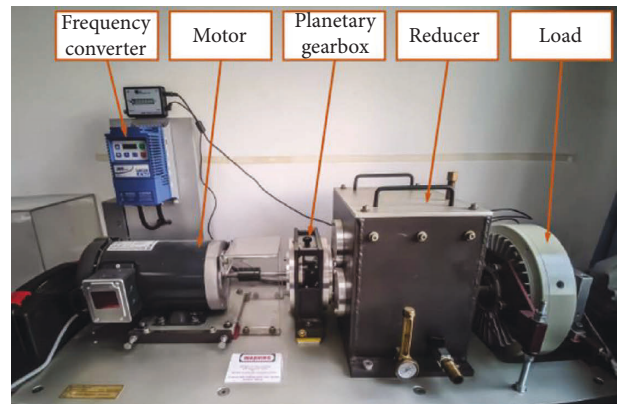


FIGURE 12: Planetary gearbox experimental bench.

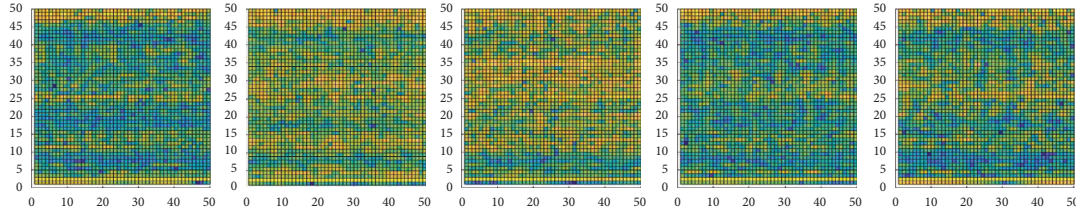


FIGURE 13: Time-frequency matrix under different health conditions (from left to right, healthy, worn, root cracks, tooth broken, and tooth missing).

TABLE 2: Architecture of VGG-5 network.

Layer	Parameter and value
Conv 1	Kernel $3 * 3 * 32$ , stride 1, padding(same)
Conv 2	Kernel $3 * 3 * 32$ , stride 1, padding(same)
Pooling 1	Size $2 * 2$ , stride 1
Conv 3	Kernel $3 * 3 * 32$ , stride 1, padding(same)
Conv 4	Kernel $3 * 3 * 32$ , stride 1, padding(same)
Pooling 2	Size $2 * 2$ , stride 1
fc	Units 128

TABLE 3: Design of algorithm parameter.

Framework	Parameter	Value
GA	Population size	2
	Number of generation	15
	Offspring size	2
CNN	Batch size	64
	Epoch	10

3.1.2. *Experimental Environment.* The experiments in this paper are carried out under the TensorFlow-GPU framework, and the configuration environment is CUDA 11.6, CUDNN v 8.4.1. The device graphics card is NVIDIA GeForce GTX 1660 SUPER.

3.1.3. *CNN Architecture.* VGG network model is a CNN model jointly developed by the Computer Vision Group of Oxford University and DeepMind, which has a good performance in transfer learning tasks. Its structure is simple and regular, and the structure of convolutional layer consists of kernel  $3 * 3$ , stride 1, and padding (same), and the pooling layer consists of pooling  $2 * 2$  and stride 2. Such a architecture means that fewer parameters need to be concerned in the evolution of the genetic algorithm. This paper chooses VGG-5 as the initial network net\_0, and its architecture is shown in Table 2.

3.1.4. *Parameter Design.* Considering the computational scale of the experiments in this paper, the design of the algorithm parameters is shown in Table 3.

3.2. *Experimental Results.* Including the initial network structure, the algorithm generated a total of 32 network structures. During the operation of the algorithm, the fitness showed an upward trend of oscillation, showing the global

optimization process of the algorithm, as shown in Figure 14. The fitness of the first offspring reached a peak value of 4.81 in the 12th generation evolution, which was greatly improved compared with the fitness of the initial VGG-5 network of 1.19. The loss function and accuracy value of the network on the test set during the evolution process are plotted as shown in Figure 15.

In this experiment, the first offspring in the evolution of the 12th generation is the optimal solution, and its corresponding network structure is shown in Table 4. It can be seen that the model adds two convolutional layers compared to VGG-5 and adjusts the number of convolution kernels in the convolutional layer and the number of neurons in the fully connected layer. The computational parameters are also reduced from 346,240 to 229,413, which also improves the computational efficiency of the network.

In order to further verify the feasibility of this method, the network structure obtained in this experiment is compared with the deeper VGG structure network because increasing the network depth is an important means to improve the performance of VGG structure network. The depth of the network obtained in this experiment is 10, so the VGG-11 network with similar depth and the deeper VGG-16 network are selected as the comparison networks. Under the same environment configuration, the test results (10 times) of each network structure on the experimental data test set are shown in Table 5.

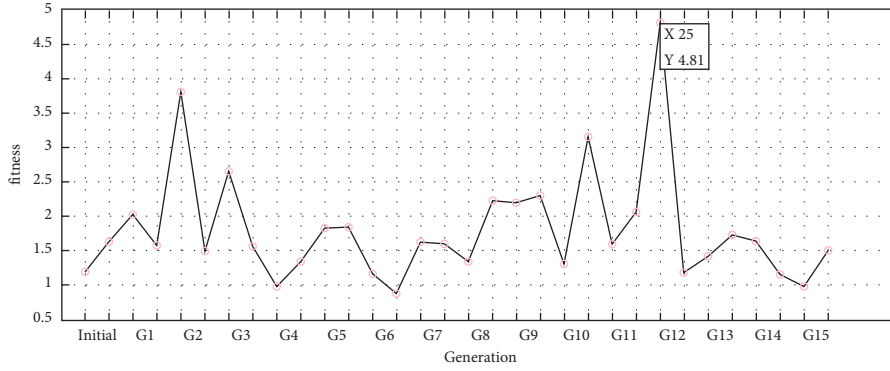


FIGURE 14: Fitness of offspring.

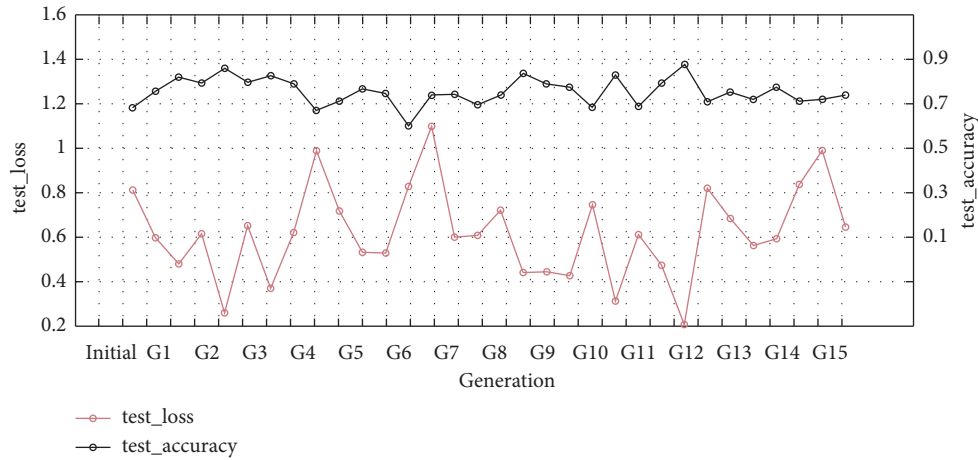


FIGURE 15: Loss and accuracy of generated networks on test set.

TABLE 4: Optimal solution network structure.

Layer	Parameter and value
Conv 1	Kernel 3 * 3 * 16, stride 1, padding(same)
Conv 2	Kernel 3 * 3 * 16, stride 1, padding(same)
Pooling 1	Size 2 * 2, stride 2
Conv 3	Kernel 3 * 3 * 32, stride 1, padding(same)
Conv 4	Kernel 3 * 3 * 8, stride 1, padding(same)
Conv 5	Kernel 3 * 3 * 8, stride 1, padding(same)
Conv 6	Kernel 3 * 3 * 64, stride 1, padding(same)
Conv 7	Kernel 3 * 3 * 128, stride 1, padding(same)
Conv 8	Kernel 3 * 3 * 64, stride 1, padding(same)
Pooling 2	Size 2 * 2, stride 2
fc	Units 256

TABLE 5: Test results on the experimental data test set.

Network structure	Computable parameters	Min accuracy	Max accuracy	Mean of accuracy	Variance of accuracy	Training time (s)
VGG-5	28064	0.770	0.876	0.824	0.00168	25
ECNN	159336	0.878	0.914	0.901	0.00023	28
VGG-11	83232	0.880	0.904	0.888	0.00027	26
VGG-16	129472	0.818	0.920	0.864	0.00162	28

The results show that with similar training time, the network structure obtained in this experiment is superior to the network structure obtained by the general

optimization method in mean and variance of accuracy on the test set. It is proved that the method presented in this paper has certain guiding significance in solving the

problem of network structure design in the fault diagnosis of rotating machinery.

#### 4. Conclusion and Outlook

In this paper, aiming at the problem of neural network structure design in rotating machinery fault diagnosis, we proposed an ECNN model by combining genetic algorithm with CNN, which realized the automation of network structure design in mechanical fault diagnosis. The idea of this paper is using the genetic algorithm to code and optimize the network structure of the preset basic CNN model. The optimization process is a global optimization, which ensures that there will be no overfitting or falling into a local optimal solution. Although the global search of the algorithm has a certain randomness, taking the fitness as the return value provides guidance for the search direction. Experiments show that the method proposed in this paper can adaptively optimize the network model according to the time-frequency dataset of vibration signals, and the final diagnosis effect of the network model is higher than that of the initial model. Future work will consider adding more evolvable parameters to the algorithm and expanding the search range.

#### Data Availability

The data used in this study are all owned by the research group and will not be transmitted.

#### Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

#### Acknowledgments

This research was funded by the Key R&D Program of Shanxi Province (International Cooperation, grant no. 201903D421008). This study was also supported by the Beijing Municipal Natural Science Foundation (grant no. 3192025).

#### References

- [1] S. Tang, S. Yuan, and Y. Zhu, "Data preprocessing techniques in convolutional neural network based on fault diagnosis towards rotating machinery," *IEEE Access*, vol. 8, pp. 149487–149496, 2020.
- [2] K. He, Z. Xiangyu, R. Shaoqing, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, May 2016.
- [3] G. Huang, L. Zhuang, V D M Laurens, and Q. W. Kilian, "Densely connected convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, HI, USA, July 2017.
- [4] A. Krizhevsky, I. Sutskever, and G. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, vol. 23, pp. 1097–1105, 2012.
- [5] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, pp. 84–90, 2017.
- [6] C. Szegedy, L. Wei, J. Yangqing, and P. Sermanet, "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Boston, MA, June 2015.
- [7] R. Chen, X. Huang, Y. Li-xia, B. Tang, Y. Teng-wei, and J. Zhou, "Rolling bearing fault identification based on convolutional neural network and discrete wavelet transform," *Journal of Vibration Engineering*, vol. 31, no. 5, pp. 883–891, 2018.
- [8] L. Wang, Y. Xie, Z. Zhou, Y. Zhang, and X. Zhao, "Motor fault diagnosis based on convolution neural network," *Journal of Vibration, Measurement & Diagnosis*, vol. 37, no. 6, pp. 1208–1215+1283, 2017.
- [9] X. Zeng, Y. Liao, and W. Li, "Gearbox fault classification using S-transform and convolutional neural network," in *Proceedings of the 10th international conference on sensing technology (ICST)*, pp. 1–5, IEEE, Nanjing, China, November 2016.
- [10] D. Belmiloud, T. Benkedjouh, M. Lachi, A. Laggoun, and J. P. Dron, "Deep convolutional neural networks for Bearings failure prediction and temperature correlation," *Journal of Vibroengineering*, vol. 20, no. 8, pp. 2878–2891, 2018.
- [11] P. Bai, H. Xu, and Li Sun, "A recognition algorithm for modulation schemes by convolution neural network and spectrum texture," *Xibeigongye Daxue Xuebao/Journal of Northwestern Polytechnical University*, vol. 37, no. 4, pp. 816–823, 2019.
- [12] A. Kumar, G. Vashishtha, C. P. Gandhi, H. Tang, and J. Xiang, "Tacho-less sparse CNN to detect defects in rotor-bearing systems at varying speed," *Engineering Applications of Artificial Intelligence*, vol. 104, Article ID 104401, 2021.
- [13] H. Li, Q. Zhang, X. Qin, and Y. Sun, "Fault diagnosis method for rolling bearings based on short-time Fourier transform and convolution neural network," *Journal of Vibration and Shock*, vol. 37, no. 19, pp. 124–131, 2018.
- [14] D. Verstraete, A. Ferrada, E. L. Droguett, V. Meruane, and M. Modarres, "Deep learning enabled fault diagnosis using time-frequency image analysis of rolling element bearings," *Shock and Vibration*, vol. 2017, Article ID 5067651, pp. 1–17, 2017.
- [15] D. T. Hoang and H. J. Kang, "Rolling element bearing fault diagnosis using convolutional neural network and vibration image," *Cognitive Systems Research*, vol. 53, pp. 42–50, 2019.
- [16] L. Wen, X. Li, and L. Gao, "A new two-level hierarchical diagnosis network based on convolutional neural network," *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 2, pp. 330–338, 2020.
- [17] Q. Qian, Y. Qin, Y. Wang, and F. Liu, "A new deep transfer learning network based on convolutional auto-encoder for mechanical fault diagnosis," *Measurement*, vol. 178, no. 2021, Article ID 109352, 2021.
- [18] Yi Qin, Q. Qian, Y. Wang, and J. Zhou, "Intermediate distribution alignment and its application into mechanical fault transfer diagnosis," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 10, pp. 7305–7315, 2022.
- [19] E. Bochinski, S. Tobias, and T. Sikora, "Hyper-parameter optimization for convolutional neural network committees based on evolutionary algorithms," in *Proceedings of the IEEE international conference on image processing (ICIP)*, IEEE, Beijing, China, September 2017.



- [20] I. Essiet, Y. Sun, and Z. Wang, "Big data analysis for gas sensor using convolutional neural network and ensemble of evolutionary algorithms," *Procedia Manufacturing*, vol. 35, pp. 629–634, 2019.
- [21] M. Zhang, H. Li, S. Pan, J. Lyu, S. Ling, and S. Su, "Convolutional neural networks-based lung nodule classification: a surrogate-assisted evolutionary algorithm for hyperparameter optimization," *IEEE Transactions on Evolutionary Computation*, vol. 25, no. 5, pp. 869–882, 2021.
- [22] I. Goodfellow, Y. Bengio, and A. Courville, "Deep learning," *Cambridge Times*, vol. 1, pp. 326–366, 2016.
- [23] J. Gu, Z. Wang, J. Kuen et al., "Recent advances in convolutional neural networks," *Pattern Recognition*, vol. 77, pp. 354–377, 2018.
- [24] D. E. Goldberg and J. H. Holland, "Genetic algorithms and machine learning," *Machine Learning*, vol. 3, no. 2/3, pp. 95–99, 1988.