

An Improved Alexnet-Based Method for Fault Diagnosis of Rolling Bearings in Multiple Operating Conditions

Chuanhao Wang¹, Yongjian Sun^{2*}, Xiaohong Wang³

¹School of Automation and Electrical Engineering, University of Jinan, Jinan, China.

²University of Jinan, China

³University of Jinan, China

*Corresponding author: Yongjian Sun, Electrical Engineering, University of Jinan, Jinan, Shandong, China. Email: sunyongjian2006@163.com

Citation: Wang, Z., Wang, J., Sun, Y. (2023) An Improved Alexnet-Based Method for Fault Diagnosis of Rolling Bearings in Multiple Operating Conditions. *J Artif Intell Mach Learn & Data Sci*, 1(1), 30-38.

Received: 30 December, 2022; **Accepted:** 02 Marh, 2023; **Published:** 10 March, 2023

Copyright: © 2023 Sun, Y., et al., This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

ABSTRACT

Fault diagnosis is particularly important in mechanical engineering systems, where early detection of problems can save costs and time. Traditional fault diagnosis methods rely heavily on features extracted by expert experience, but with the development of intelligent manufacturing, deep learning provides an effective idea for automatically extracting data features. Convolutional neural networks are very effective deep learning methods. In this study, an improvement scheme of convolutional neural network Alexnet algorithm is presented. First, the rolling bearing vibration signal is converted to a symmetric snowflake image in polar coordinates according to the symmetrized dot pattern (SDP) principle. Then, it is fed into the CNN, and the fault feature is extracted. The impact of manual feature extraction is eliminated. According to the ratio of accuracy and time proposed in this paper, the size of each layer of convolution kernel is adjusted accordingly to determine the optimal model, and test experiments are carried out to verify the robustness of the method under different bearing conditions, and the accuracy rate reaches 98.5%. Comparison with other methods shows that the proposed method has achieved significant improvement.

Keywords: Rolling bearing, Symmetrized Dot Pattern, convolutional neural network, fault diagnosis.

Introduction

In the cement production industry, rolling bearings are common parts of rotating equipment, and their health status affect the safety of the installation, stability, and reliability. Equipment by a long-time load operation and the harsh working environment, rolling bearing small failure is difficult to directly go to observe, if not timely maintenance, it will cause serious losses. Therefore, it is vital to realize the accurate identification of rolling bearing health status. Rolling bearing fault diagnosis is divided into data acquisition, pre-processing, feature extraction, and fault classification [1,2]. In the past decades, scholars have published a large number of articles on rolling bearing fault diagnosis, and these materials are very important references for future research. For example, scholars used Western Reserve University data as the research object of bearing failure, and this literature

introduces how to better use Western Reserve University bearing data, which is a representative article in this field [3]. Previously, the traditional fault diagnosis was first based on manual feature extraction by experts, which was extremely tedious. Later, with the development of artificial intelligence technology, machine learning such as Support Vector Machine (SVM) [4], k-nearest neighbour (KNN) [5], BP neural network [6][7], etc. Although these methods have gained some effect in the area of fault diagnosis, the shallow network structure is hard to obtain the deep feature information and reduces the accuracy of rolling bearing fault diagnosis. The increasing maturity of deep learning theory and the continuous improvement of computer computing power has led to a wide range of applications of convolutional neural networks in fault diagnosis. Among these, CNN has strong prediction and classification functions. Han proposed a

combination of convolutional neural network and SVM, using the time domain map of rolling bearings as input, to classify and diagnose faults through CNN feature extraction [8]. A novel fault diagnosis method combining morphable CNN and Long-Short Term Memory (LSTM) with each other was proposed by Wang [9]. Zhao presented normalized convolutional neural networks to diagnose different faults to consider the magnitude and directions of data imbalances and variable operating conditions [10]. The detection of complicated signals by using CNN was presented by Lu [11], which transformed the raw signal into an image, and put these into the CNN for categorization. There are very many models of CNN such as Lenet-5, Alexnet, and Googlenet models, etc. Many scholars use these models to classify and compare rolling bearings. Rakibul Islam used lenet-5 model to classify and identify CT images of new coronary pneumonia [12]. Yilmaz used AlexNet, GoogLeNet and ResNe-50 to classify and identify images of his skin necrosis in the differential diagnosis of benign and malignant [13]. After understanding, we realize that CNN-based fault diagnosis methods usually require a large number of images, which can easily cause errors if the number is insufficient. To resolve this issue, Yu proposed a One-Dimension Residual Convolutional Auto-Encoder(1-DRCAE) [14]. From the raw vibrational signal, the encoder can rebuild the signal by the reverse pleat product operation, and ultimately utilize the residuals to learn the features. To resolve this issue of Multi conditions integration, Zhao proposes a framework for deep multi-scale CNN (MSCNN) based on primitive extended convolution [15]. Yuan proposed CNN and SVM. Above all, the continuous wavelet transform is used to turn the 1D raw vibration signal into a 2D temporal frequency image [16]. In addition, the acquired temporal frequency image inputs are trained. In the end, the diagnostics of the fault position and severity are accomplished. These methods provide an effective classification of the kinds of rolling bearing failures.

In the latest years, with the successful application of AI technology in the domain of facial identification and other image identification various transformations in bearing vibration signals to two-dimensional images have also gradually emerged. Li proposed a method based on WPT and CNN for solving bearing fault diagnosis without human intervention [17]. A fault diagnosis method based on Deep Convolutional Auto Encoder (DCAE) and Convolutional Neural Network (CNN) for the rolling bearing of helicopter auto-tilter with complex working conditions and high noise interference by Wan [18]. Besides, Gu proposed a method for bearings in different conditions by various terms by angular domains re-sampling, SDP, and DCNN [19]. The fault diagnosis method by using the EMD and Chebyshev distance in SDP image to solve problems by Sun [20][21]. Xu proposes hybrid deep learning combining CNN and gcForest [22]. Chen proposed that they choose the ResNet -50 as a pre-trained model network for DCNN and based on the idea of migration to solve the bearing fault classification problem learning [23]. First, the original fault signal is turned into a temporal frequency figure transform using continuous wavelets. Then, the image is further converted to RGB format as input to Discrete-Time CNN (DTCNN). Sun proposes a new metric to determine the convolutional neural network model based on accuracy and time ratio [24]. Xu developed an improved Multi-Scale CNN integrated with a Feature Attention Mecha (IMS-FACNN) algorithm that integrated CNN to solve the heritage CNN models in unstable and complicated conditions [25].

For the sake of solving the running time conditions under high precision, this paper proposes a diagnosis method combining SDP images and improved Alexnet. This paper proposes a novel assessing index comprising accuracy and running time to help determine the optimal structure of the convolutional neural network Alexnet. The second part details the design and introduction of the current scheme of the new fault diagnosis method; the third part identifies the experimental installation and carries out a comparative experiment to determine the optimal structure of the Alexnet model, the fourth part compares and analyzes its results into. Finally, the work of this paper is summarized.

Basic Theoretical Analysis

Symmetrized dot pattern

Compared with the traditional method of converting signals into time-frequency images, the following three formulas convert one-dimensional vibration signals into symmetrical shapes in polar coordinate systems, and extract the characteristics of snowflake shapes from the perspective of image processing. The following is the calculation process. In the discrete data of vibration signal, the vibration amplitude of moment n is x_n and the vibration amplitude of moment $n + l$ is x_{n+l} . By using these SDP formulas, a symmetric image in the polar coordinate space $p[\gamma(n), \theta(n), \phi(n)]$ is obtained. the schematic diagram of SDP is shown in fig.1.

$$\gamma(n) = \frac{x_n - x_{min}}{x_{max} - x_{min}} \quad (1)$$

$$\theta(n) = \theta + \frac{x_{n+l} - x_{min}}{x_{max} - x_{min}} g \quad (2)$$

$$\phi(n) = \theta - \frac{x_{n+l} - x_{min}}{x_{max} - x_{min}} g \quad (3)$$

Where $\gamma(n)$ represents the radius in the polar coordinate system; $\theta(n)$ represents the angle of rotation of the polar coordinates counterclockwise along the initial line; $\phi(n)$ while is the angle of rotation of the polar coordinates clockwise along the initial line. Here, x_{min} and x_{max} are the minimum value and the maximum in the discrete sequence of vibration signals; g is the angle amplification factor, θ is the rotation angle of the mirror plane of symmetry, and l is the time interval of the parameter.

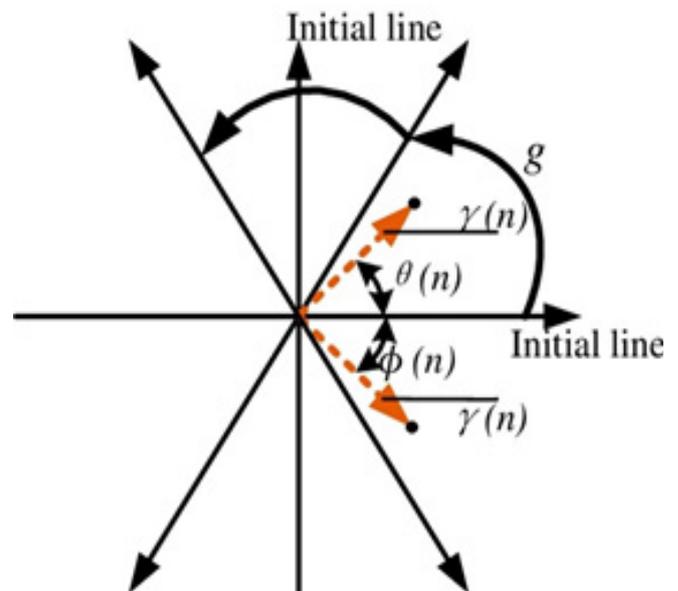


Figure 1: SDP schematic diagram.

Convolutional neural network

In recent years, Convolutional neural network (CNN) has been widely used in the field of fault diagnosis because of their great advantages in image recognition. CNN is a type of neural network specifically designed to cope with data with a reticular formation. Recently, CNN has been widely used due to the huge superiority in fault classification. The *CNN* model is formed of five layers. As shown in fig.2.

- Input Layer. The image to be recognized is preprocessed for input.
- Convolutional Layer. The convolutional layer is unique to CNN. The formula is shown in eq.4.

$$X_j^l = f \left(\sum_{i \in R_i} X_j^{l-1} * K_{ij}^l + b_j^l \right) \quad (4)$$

- Where X_j^l refers to the output feature image j in the current layer l . K_{ij}^l is convolution kernel and b_j^l is bias. R_i is the congregate of input characteristic images. X_j^{l-1} is the output feature image j in the current layer l .
- Maxpool Layer: Use pooling layers to reduce the size of the model and increase the speed of computation while improving the robustness of the extracted features. The purpose of maximum pooling is to retain the original features while reducing the parameters of neural network training, so that the training time is reduced.
- Fully Connected Layer: The fully connection layer can combine the local information with the classification in the convolution layers or pooling layers to improve the performance of the network. The fully connected layer is the traditional neural network. It can be expressed as:

$$X_j^l = \sigma(W_j^l * X_j^{l-1} + b_j^l) \quad (4)$$

where $\sigma()$ is the activation function and W_j^l is the weight. The other parameters are the same as above.

- Output layer: The output layer is passed through the Softmax layer to obtain the problems of a probability distribution for which the current sample belongs to different kinds.

Structure optimization index

Accuracy and running time are very important for the CNN model. So, a metric on time and accuracy is used to describe the structure of CNN, as shown in eq.7.

$$I_{new} = \frac{acc}{\frac{t}{t_{max}}} \quad (4)$$

Where acc is the accuracy of the *CNN* in solving the rolling bearing faults, t is the time taken for the whole model to end its work, and t_{max} is the maximum amount of time it takes to run this *Alexnet*. This indicator of time and accuracy is used to evaluate the various structure of *CNN*. The various layers are reflected in the size of the convolutional kernel. This formula essentially normalizes the running time by unitizing it, thus eliminating the effect of time units, and thus allowing to express the accuracy achieved per unit time.

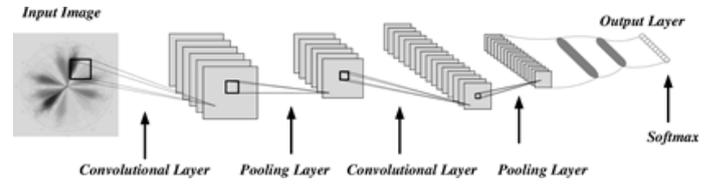


Figure 2: Structure of a CNN model.

Simulation experiment

Experimental apparatus and data acquisition

Experimental equipment

The test rig is illustrated in fig.3. It includes a 1.5kW motor, a power tester, and a torque transmitter. The sampling frequency is set to 12kHz, and the motor load is between 0 and 3 horsepower. The diameters of bearing were 0.1778 mm, 0.3556 mm, and 0.5334 mm. Many different kinds of rolling bearings which are composed of are used to test. Damage bearing components by using EDM for the sake of getting the bearing failure data. To collect the vibration signals of these fault bearings, people placed the sensor on the bearing at the drive end.

$$f_r = \frac{n}{60} \quad (4)$$

$$f_{ic} = \frac{1}{2}Z \left(1 + \frac{d}{D} \cos\alpha \right) f_r \quad (4)$$

$$f_{oc} = \frac{1}{2}Z \left(1 - \frac{d}{D} \cos\alpha \right) f_r \quad (4)$$

$$f_b = \frac{1}{2} \left(1 - \frac{d}{D} \cos\alpha \right) f_r \quad (4)$$

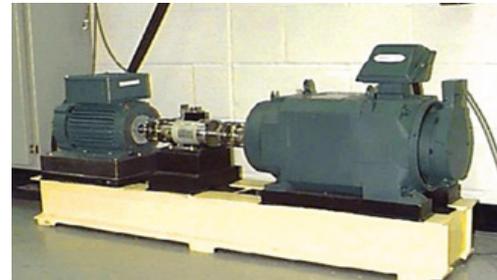


Figure 3: Bearing test bench.

Where the characteristic frequency of each part of the rolling bearing can be got. Further, the frequency and the characteristic orders of the fault features can be obtained. The damaged rolling bearings were placed on the test rig. Experiments were carried out under different horsepower to collect vibration signals. In this paper, ten sets of data were selected, including one set of normal data, three sets of outer ring fault data, three sets of rolling element fault data, and three sets of inner ring fault data. The CPU is Intel Core i5 7300HQ, and the graphics card is NVIDIA GeForce GTX 1050. The software used to run Alexnet model in MATLAB R2019a.

Data acquisition

In this paper, to test the diagnosis of multiple working conditions, we selected 10 working conditions, one of which is normal, and the other selected outer ring failure, inner ring failure and rolling body failure with different damage radius. To ensure that there are enough data sets, we select 3000 sets of data to train and validate this model. Figure 4 shows the images of the selected time domain signals for the ten operating

conditions, which paves the way for subsequent conversion into SDP images.

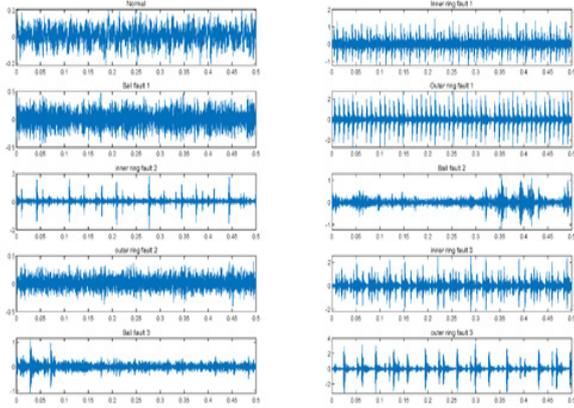


Figure 4: The time-domain signal under 10 working conditions.

Determine parameters

SDP parameters were determined according to the selection of three parameters of γ , θ , and ϕ . Here, we use the Pearson correlation parameter to verify the correlation, and the formula is shown in eq.12.

$$\rho(X, Y) = \frac{cov(X, Y)}{\sigma_X \sigma_Y} \quad (4)$$

Where, the Pearson correlation coefficient $\rho(X, Y)$ of two variables (X, Y) is equal to the product of their covariance $Cov(X, Y)$ divided by their respective standard deviations (σ_X, σ_Y). Through a large number of experiments to prove that θ is 60° , the effect is better. By choosing this angle, the formed SDP images can be divided equally, the overlap is not obvious, and the extractable features are more obvious. Therefore, we transform the data of 10 working conditions into snowflake-like images by the principle of SDP, and come to extract their features. The snowflake image is displayed in fig.5.

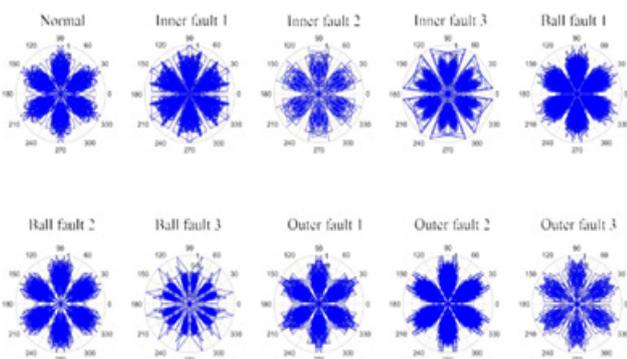


Figure 5: The SDP images of 10 working conditions.

Specific implementation

The process of fault diagnosis under multiple conditions is shown in fig.6.

- 2400 images of snowflakes formed by the SDP principle were selected for training.
- Selecting the Alexnet model.
- The processed images are fed into the Alexnet and the size of each convolution kernel is continuously changed based on the correctness rate and the determined metrics to determine a more appropriate Alexnet model.

The size of the convolution kernel is determined, and images

of 10 randomly selected working conditions are input into its model for testing to verify the stability and accuracy of the model.

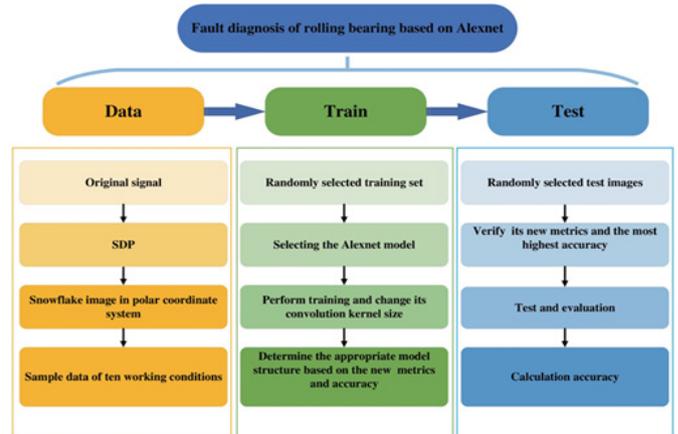


Figure 6: Procedure of bearing fault diagnosis under different conditions.

Determine the best Alexnet model

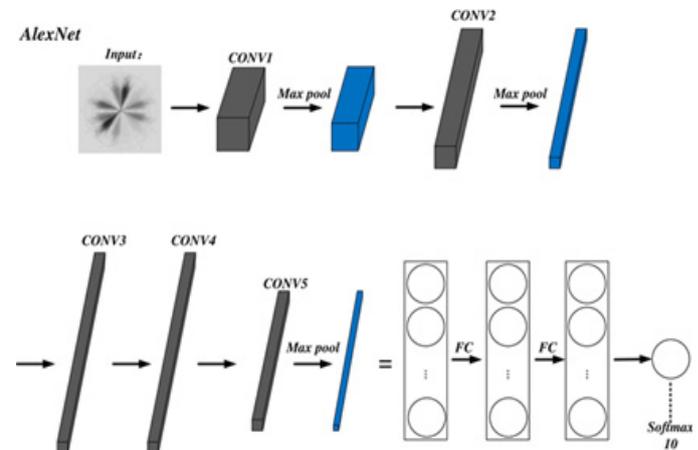


Figure 7: Structure of the Alexnet model.

In picture7, the principle and process of Alexnet operation are clearly shown. Based on the new metrics and the high correct rate, to adjust the size of the convolution kernel for each layer by Alexnet. The experiments mainly changed the size of convolutional kernels in each layer of the model from 2×2 to verify the effects of convolutional kernel size on its fault diagnosis and to test its acc and running time. The results are shown in tab.1.

According to the table, the Alexnet model achieves the highest accuracy and the highest index when the convolution core size is 15×15 and 8×8 . But the runtime, on the other hand, seems to be in the low to medium range. To verify that the multi-layer convolutional kernels are optimized, the highest correct rate of the first layer (15×15) and the highest I_{new} (8×8) are selected to continue to tuning the second layer of convolutional kernels. Using the method of controlling variables, as shown in the results of Table 1, we use the ratio of accuracy to time as a metric to evaluate the impact of changes in the first layer of convolution kernels.

In order to reflect the improvement of the convolutional neural network by the adopted indicators, we conduct a comparative experiment below. The convolution kernel size with the highest accuracy rate of the first layer and the convolution kernel size with the highest index were used for subsequent improvement. The results of the second layer of comparison are shown in Table 2, Table 3 and Table 4.

The rationale for not beginning the first convolution kernel size with 1×1 is that the running time and accuracy are too low. The provided findings have minimal reference value. According to tab.2, the second layer still has the greatest accurate rate for a kernel size of 5×5 and the biggest I_{new} for a kernel size of 14×14 when the first layer is 8×8 . As demonstrated in tab.3 when the first layer is 15×15 , the second layer has the greatest accurate rate for a kernel size of 2×2 and the maximum I_{new} for a kernel size of 17×17 . As seen in tab.4, tab.5, tab.6, tab.7, second layer utilizes these four convolution kernel sizes to estimate the size of the third layer convolution kernel. The convolution and size obtained by the correctness and I_{new} , as well as their combination, change the size of the convolution kernel of the third layer in the first two layers. The optimal network is thus discovered.

Table 1: Running time and correctness for changing the size of the first layer convolution kernel.

Convolution kernel size	Accuracy	Running time	I_{new}
2×2	81%	2425	0.94828
3×3	90.83%	2374	1.08621
4×4	88.33%	2299	1.09077
5×5	92.83%	2312	1.13989
6×6	96.5%	2316	1.18291
7×7	99%	2371	1.18541
8×8	98%	2332	1.19306
9×9	93.5%	2403	1.10464
10×10	98.17%	2383	1.16955
11×11	97.83%	2387	1.16355
12×12	97.67%	2440	1.13641
13×13	97.33%	2527	1.09347
14×14	97.5%	2611	1.06014
15×15	99.83%	2794	1.01437
16×16	98.17%	2795	0.99715
17×17	95.33%	2806	0.96451
18×18	97%	2839	0.97
19×19	98.67%	2817	0.99440
20×20	96%	2806	0.97129
21×21	99%	2812	0.99950

Table 2: Running time and correctness for changing the size of the second layer convolution kernel 1.

Convolution kernel size	Accuracy	Running time	I_{new}
2×2	98%	2332	0.98798
3×3	95.33%	2311	0.96980
4×4	95.33%	2303	0.97317
5×5	99%	2342	0.99380
6×6	93.67%	2293	0.96039
7×7	96%	2351	0.96
8×8	91.33%	2323	0.92431
9×9	92.17%	2292	0.94543
10×10	95.33%	1950	1.14934
11×11	96.33%	1940	1.16738
12×12	97.67%	1915	1.19907
13×13	93%	1957	1.11723
14×14	97.67%	1877	1.22335
15×15	96.33%	1891	1.19763
16×16	92.5%	1878	1.15797
17×17	96.83%	1869	1.21802

18×18	96.67%	1863	1.21992
19×19	96.33%	1864	1.21498
20×20	96.67%	1883	1.20696
21×21	92.33%	1866	1.16328
22×22	96.83%	1870	1.21736
23×23	85.83%	1870	1.07907
24×24	81.33%	1859	1.02854
25×25	91%	1879	1.13859
26×26	81%	1893	1.00597

Table 3: Running time and correctness for changing the size of the second layer convolution kernel 2.

Convolution kernel size	Accuracy	Running time	I_{new}
2×2	99.83%	2794	0.9983
3×3	96.67%	2581	1.04648
4×4	97.83%	2566	1.06523
5×5	96.83%	2340	1.15617
6×6	97.83%	2388	1.14463
7×7	96.17%	2292	1.17233
8×8	97.33%	2308	1.17825
9×9	98.33%	2294	1.19762
10×10	95.5%	2292	1.16417
11×11	98%	2301	1.18997
12×12	96.67%	2290	1.17946
13×13	99%	2179	1.26942
14×14	97.33%	2160	1.25898
15×15	98.5%	1937	1.42080
16×16	93%	1944	1.33663
17×17	98.33%	1927	1.42571
18×18	94.5%	1928	1.36946
19×19	95%	1925	1.37886
20×20	91.5%	1864	1.37152
21×21	98.17%	1963	1.39728
22×22	84.83%	1943	1.21984
23×23	94.17%	1914	1.37467
24×24	83%	1919	1.20845
25×25	84.33%	2271	1.03751

Table 4: Running time and correctness for changing the size of the second layer convolution kernel 3.

Convolution kernel size	Accuracy	Running time	I_{new}
2×2	95.33%	2334	1.02110
3×3	97.33%	2337	1.04118
4×4	97.67%	2331	1.04751
5×5	98.33%	2361	1.04119
6×6	97.17%	2500	0.9717
7×7	99%	2263	1.09368
8×8	98.17%	2129	1.15277
9×9	95.83%	2045	1.17152
10×10	99.33%	1951	1.27281
11×11	94.33%	1932	1.22063
12×12	95.33%	1921	1.24063
13×13	97.33%	2020	1.20458
14×14	90%	1965	1.14504
15×15	57.67%	1976	0.72963

Based on the new indicators and the accuracy rate, the size of the first three layers of convolutional kernels is adjusted.

Because the Alexnet model comprises five convolutional layers, in order to diagnose these ten working conditions, the preceding processes must be repeated to establish the right rate and running duration for the five convolutional layers, as well as the requisite structure. Then, in the following layers, continue to modify the size of the convolutional kernels, seeking for the convolutional kernel size with the best accuracy and the largest value of I_{new} . The following tables contains detailed statistics as well as the appropriate convolutional kernel size.

Table 5: Running time and correctness for changing the size of the third layer convolution kernel 2.

Convolution kernel size	Accuracy	Running time	I_{new}
2×2	97.83%	1946	0.98886
3×3	93.17%	1907	0.96101
4×4	93.17%	1871	0.97951
5×5	90.83%	1913	0.93394
6×6	97.5%	1967	0.975
7×7	95.83%	1908	0.98793
8×8	94.5%	1923	0.96662
9×9	96%	1872	1.00872
10×10	94.17%	1894	0.97799

Table 6: Running time and correctness for changing the size of the third layer convolution kernel 3.

Convolution kernel size	Accuracy	Running time	I_{new}
2×2	97.83%	2738	1.01582
3×3	98.5%	2550	1.09818
4×4	94.17%	2843	0.9417
5×5	99%	2694	1.04475
6×6	96.67%	2775	0.99039
7×7	99.17%	2378	1.18562
8×8	97%	2396	1.15096
9×9	98.33%	2394	1.16772
10×10	92.83%	2382	1.10796
11×11	93.17%	2348	1.12812
12×12	96.83%	2356	1.16845
13×13	81%	2361	0.97536
14×14	42%	2351	0.50789

Table 7: Running time and correctness for changing the size of the third layer convolution kernel 4.

Convolution kernel size	Accuracy	Running time	I_{new}
2×2	98.17%	2059	1.01364
3×3	97.17%	1938	1.06596
4×4	96.83%	1978	1.04075
5×5	96%	1968	1.03707
6×6	93.83%	2015	0.98999
7×7	97.17%	2061	1.00235
8×8	95.5%	2126	0.955

Table 8: Running time and correctness for changing the size of the fourth layer convolution kernel 1.

Convolution kernel size	Accuracy	Running time	I_{new}
2×2	96.83%	1915	0.97538
3×3	97.17%	1895	0.98913
4×4	88.83%	1905	0.89949
5×5	98.17%	1886	1.00408
6×6	95.33%	1929	0.9533
7×7	98.5%	1911	0.99427

Table 9: Running time and correctness for changing the size of the fourth layer convolution kernel 2.

Convolution kernel size	Accuracy	Running time	I_{new}
2×2	97.83%	1873	1.19141
3×3	95.5%	1882	1.15747
4×4	90.5%	1913	1.07909
5×5	94.67%	1871	1.15415
6×6	88.67%	2281	0.8867
7×7	95.5%	2239	0.97291
8×8	95.5%	2028	1.07414
9×9	92.67%	1877	1.12616
10×10	98.5%	1877	1.19701

Table 10: Running time and correctness for changing the size of the fourth layer convolution kernel 3.

Convolution kernel size	Accuracy	Running time	I_{new}
2×2	96.33%	1884	0.9633
3×3	95.83%	1847	0.97750

Table 11: Running time and correctness for changing the size of the fourth layer convolution kernel 4.

Convolution kernel size	Accuracy	Running time	I_{new}
2×2	97.5%	2385	1.17613
3×3	98.3%	2612	1.08273
4×4	98.83%	2651	1.07255
5×5	95.67%	2874	0.95770
6×6	96%	2877	0.96
7×7	98.5%	2439	1.16189
8×8	95.83%	2509	1.09885
9×9	95.17%	2416	1.13329
10×10	98.67%	2494	1.13823
11×11	97.17%	2453	1.13966

Table 12: Running time and correctness for changing the size of the fourth layer convolution kernel 5.

Convolution kernel size	Accuracy	Running time	I_{new}
2×2	81.33%	1947	0.87763
3×3	96%	1995	1.01101
4×4	97.67%	1999	1.02654
5×5	93.83%	2001	0.98519
6×6	97.67%	2093	0.98043
7×7	91.33%	1990	0.96484
8×8	98.5%	2101	0.985

Table 13: Running time and correctness for changing the size of the fourth layer convolution kernel 6.

Convolution kernel size	Accuracy	Running time	I_{new}
2×2	81.33%	1935	0.84440
3×3	99%	2009	0.99
4×4	96.33%	1967	0.98387
5×5	94.33%	1930	0.98191
6×6	96.5%	1911	1.01449

As seen in tables, the size of the preceding layer's convolutional kernels leads successive convolutions to be non-identical. The topology of neural networks and the settings of each layer cannot be guaranteed to be optimum without a lot

of theoretical backing. Establishing the ideal set of parameters for this model requires a significant number of experimental data supports, based on the highest accuracy rate and I_{new} .

Table 14: Running time and correctness for changing the size of the fifth layer convolution kernel 1.

forth layer	fifth layer	Accuracy	Running time	I_{new}
5 × 5	2 × 2	97.17%	1900	0.99625
5 × 5	3 × 3	97.67%	1948	0.9767
5 × 5	4 × 4	98.83%	1899	1.01380
7 × 7	2 × 2	98.33%	1908	0.9833

Table 15: Running time and correctness for changing the size of the fifth layer convolution kernel 2.

forth layer	fifth layer	Accuracy	Running time	I_{new}
9 × 9	2 × 2	98.67%	1873	0.98828
9 × 9	3 × 3	99%	1876	0.99
10 × 10	2 × 2	93.83%	1875	0.9383
7 × 7	2 × 2	98.33%	1908	0.9833

Table 16: Running time and correctness for changing the size of the fifth layer convolution kernel 3.

forth layer	fifth layer	Accuracy	Running time	I_{new}
2 × 2	2 × 2	95.17%	1894	0.9517
2 × 2	3 × 3	93.83%	1874	0.94831
3 × 3	2 × 2	94.5%	1877	0.945

Table 17: Running time and correctness for changing the size of the fifth layer convolution kernel 4.

forth layer	fifth layer	Accuracy	Running time	I_{new}
2 × 2	2 × 2	98.17%	2352	0.99547
2 × 2	3 × 3	97.5%	2375	0.97911
2 × 2	4 × 4	97.33%	2377	0.97658
2 × 2	5 × 5	97%	2385	0.97
2 × 2	6 × 6	85%	2374	0.85394
2 × 2	7 × 7	10%	2350	0.10149
2 × 2	8 × 8	10%	2347	0.10162
4 × 4	2 × 2	99.17%	2353	1.00350
4 × 4	3 × 3	96.5%	2381	0.965
4 × 4	4 × 4	10%	2366	0.10063

Table 18: Running time and correctness for changing the size of the fifth layer convolution kernel 5.

forth layer	fifth layer	Accuracy	Running time	I_{new}
4 × 4	2 × 2	97.17%	1960	1.00739
4 × 4	3 × 3	93.83%	2032	0.9383
4 × 4	4 × 4	98.5%	1964	1.01910
4 × 4	5 × 5	95.67%	1992	0.97591
4 × 4	6 × 6	96.67%	1991	0.98661
8 × 8	2 × 2	96.83%	2024	0.9683

Table 19: Running time and correctness for changing the size of the fifth layer convolution kernel 6.

forth layer	fifth layer	Accuracy	Running time	I_{new}
3 × 3	2 × 2	98.67%	1978	1.00189
3 × 3	3 × 3	97.17%	1904	1.00947
3 × 3	4 × 4	95.67%	1921	0.98509
3 × 3	5 × 5	98.33%	1978	0.9833

The comparison experiment is like a bifurcation structure, based on the correct rate of changing the size of each layer of convolution kernel and the highest value of the new index, and then adjusting the size of the convolution kernel of the next layer in turn to obtain the required accuracy rate and the highest value of the new indicator, compare, optimize, and finally obtain the optimal result of each layer, so as to establish the structure of this convolutional neural network. The table above shows the results after each change and also shows the optimization process.

Comparison and analysis

To create a feature picture, the image is twisted. The initial convolutional layer will extract some superficial data features. When more layers are added, more features are extracted, and the features become more complicated. Each layer of the Alexnet model can display the features of the SDP image, and the features contained in the image will be the key to our classification of its faults. At the same time, the images become increasingly blurred. The following five figures show the image characteristic maps of the relative convolutional layers in Alexnet (see Figures 8).

Through the above work, the layers with the highest metrics (15×15 , 17×17 , 2×2 , 4×4 , 4×4) and the layers with the highest correctness (15×15 , 2×2 , 7×7 , 4×4 , and 2×2) were selected. After training and determining the size of the convolution kernel of the model, we test the stability of Alexnet by randomly selecting images of ten working conditions for testing, and the results of the test are shown in the confusion matrix in fig 9, fig 10, and the accuracy can be derived as 97.5% and 98.5%. Through the above table, we get two sets of convolution kernel sizes, and compare them through the validation set to see if it has been effectively improved. The test set results are shown in the confusion matrix in the following figure. Obviously, the accuracy of the neural network structure according to the new indicator is higher than that of the other group. So, we get the optimal structure.

The size of the convolution kernel is determined by the highest accuracy rate, as shown in the confusion matrix. It is difficult to distinguish the three working conditions of inner ring fault 2, inner ring fault 3, and outer ring fault 3 when testing these ten working conditions.



Figure 8: The characteristic map after the convolution layer.

		Confusion Matrix										
Output Class	Ball fault 1	60	0	0	0	1	0	0	0	0	0	98.4%
		10.0%	0.0%	0.0%	0.0%	0.2%	0.0%	0.0%	0.0%	0.0%	0.0%	1.6%
	Ball fault 2	0	60	0	0	0	0	0	0	0	0	100%
		0.0%	10.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
	Ball fault 3	0	0	60	0	0	0	0	0	0	0	100%
		0.0%	0.0%	10.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
	Inner ring fault 1	0	0	0	60	0	0	0	0	0	0	100%
		0.0%	0.0%	0.0%	10.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
	Inner ring fault 2	0	0	0	0	55	0	0	0	0	0	100%
		0.0%	0.0%	0.0%	0.0%	9.2%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
Inner ring fault 3	0	0	0	0	0	55	0	0	0	0	100%	
	0.0%	0.0%	0.0%	0.0%	0.0%	9.2%	0.0%	0.0%	0.0%	0.0%	0.0%	
Normal	0	0	0	0	0	0	60	0	0	0	100%	
	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	10.0%	0.0%	0.0%	0.0%	0.0%	
Outer ring fault 1	0	0	0	0	4	3	0	60	0	5	83.3%	
	0.0%	0.0%	0.0%	0.0%	7.7%	5.5%	0.0%	10.0%	0.0%	8.3%	16.7%	
Outer ring fault 2	0	0	0	0	0	1	0	0	60	0	98.4%	
	0.0%	0.0%	0.0%	0.0%	0.0%	2.2%	0.0%	0.0%	10.0%	0.0%	1.6%	
Outer ring fault 3	0	0	0	0	1	0	0	0	0	55	88.2%	
	0.0%	0.0%	0.0%	0.0%	2.2%	0.0%	0.0%	0.0%	0.0%	9.2%	1.8%	
Ball fault 1	100%	100%	100%	100%	91.7%	91.7%	100%	100%	100%	91.7%	97.6%	
Ball fault 2	0.0%	0.0%	0.0%	0.0%	8.3%	8.3%	0.0%	0.0%	0.0%	8.3%	2.6%	
Ball fault 3	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	
Inner ring fault 1	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	
Inner ring fault 2	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	
Inner ring fault 3	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	
Outer ring fault 1	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	
Outer ring fault 2	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	
Outer ring fault 3	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	
		Target Class										

Figure 9: Test results ranged by highest accuracy.

- intelligent fault diagnosis of rolling bearings under variable working conditions and domains, *Neurocomputing*, 407, 24–38. <https://doi.org/10.1016/j.neucom.2020.04.073>
16. Yuan, L., Lian, D., Kang, X., Chen, Y., & Zhai, K. (2020). Rolling bearing fault diagnosis based on convolutional neural network and support vector machine [J]. *IEEE Access*, 8, 137395-137406. <https://doi.org/10.1109/ACCESS.2020.3012053>
 17. Li, G., Deng, C., Wu, J., Chen, Z., & Xu, X. (2020). Rolling Bearing Fault Diagnosis Based on Wavelet Packet Transform and Convolutional Neural Network [J]. *Applied Sciences*, 10(3), 770. <https://doi.org/10.3390/app10030770>
 18. Qiyang, W., Bangshu, X., Xinmin, L., Wei, SUN. (2020). Fault diagnosis for rolling bearing of swashplate based on DCAE-CNN [J]. *VIBRATION AND SHOCK*, 39(11), 273-279. DOI: 10.13465/j.cnki.jvs.2020.11.036.
 19. Gu, Y., Zeng, L., Qiu, G. (2020). Bearing fault diagnosis with varying conditions using angular domain resampling technology, SDP and DCNN [J]. *Measurement*, 156, 107616. <https://doi.org/10.1016/j.measurement.2020.107616>
 20. Yongjian, S., Shaohui, L., Wang, Y., & Xiaohong, W. (2021). Fault diagnosis of rolling bearing based on empirical mode decomposition and improved manhattan distance in symmetrized dot pattern image. *Mech. Syst. Signal Process*, 159(18), 107817. <http://dx.doi.org/10.1016/j.ymssp.2021.107817>
 21. Yongjian, S., Shaohui, L., & Xiaohong, W. (2021). Bearing fault diagnosis based on EMD and improved Chebyshev distance in SDP image. *Measurement*, 176, 109100. <https://doi.org/10.1016/j.measurement.2021.109100>
 22. Xu, Y., Li, Z., Wang, S., & Li, We. (2021). A hybrid deep-learning model for fault diagnosis of rolling bearings [J]. *Measurement*, 169(6), 108502. <http://dx.doi.org/10.1016/j.measurement.2020.108502>
 23. Chen, Z., Cen, J., & Xiong, J. (2020). Rolling bearing fault diagnosis using time-frequency analysis and deep transfer convolutional neural network [J]. *IEEE Access*, 8, 150248-150261. <https://doi.org/10.1109/ACCESS.2020.3016888>
 24. Sun, Y., & Li, S. (2022). Bearing fault diagnosis based on optimal convolution neural network [J]. *Measurement*, 190, 110702. <https://doi.org/10.1016/j.measurement.2022.110702>
 25. Xu, Z., Li, C., & Yang, Y. (2021). Fault diagnosis of rolling bearings using an improved multi-scale convolutional neural network with feature attention mechanism [J]. *ISA transactions*, 110, 379-393. <https://doi.org/10.1016/j.isatra.2020.10.054>
 26. Appana, D. K., Prosvirin, A., & Kim, J-M. (2018). Reliable fault diagnosis of bearings with varying rotational speeds using envelope spectrum and convolution neural networks. *Soft. Comput*, 20, 6719–672. <https://doi.org/10.1007/s00500-018-3256-0>