Research on Fast Circuit Fault Localization Technology Based on Convolutional Neural Network Algorithm

Rui Ding
Shandong Jiaotong University, Jinan, Shandong, 250357

Abstract. In order to improve the accuracy and efficiency of circuit fault diagnosis, a fast fault location technique based on Convolutional Neural Network (CNN) is proposed. By constructing a multi branch CNN model and combining time-domain and frequency-domain feature extraction, the model's ability to identify different types of faults has been enhanced. The experimental results show that the proposed model exhibits excellent performance in the diagnosis of various typical faults, especially in the recognition of short circuit and open circuit faults with an accuracy rate of over 98%. In addition, the robustness and real-time performance of the model under complex operating conditions have been effectively verified, which can meet the needs of online monitoring and fault diagnosis in power systems. The results indicate that the CNN model can effectively improve the accuracy and efficiency of circuit fault location, providing a new technical means for circuit maintenance and fault diagnosis.

Keywords: circuit fault diagnosis, convolutional neural network, real-time performance, robustness, deep learning.

1. Introduction

The rapid localization and accurate diagnosis of circuit faults play a crucial role in ensuring the reliability and safety of electronic devices. With the continuous development of electronic technology, the complexity and types of faults in circuit systems are increasing, and traditional fault detection methods are no longer able to meet the requirements of high efficiency and accuracy. The key to fault diagnosis lies in the ability to identify the type of fault in a timely manner and accurately locate the fault location. However, existing methods often rely on manual experience or preset rules, which are easily limited by signal changes and external interference under complex working conditions. In recent years, deep learning techniques, especially convolutional neural networks (CNN), have made significant breakthroughs in the fields of image processing and pattern recognition, gradually demonstrating their potential in fault diagnosis. CNN can automatically extract high-dimensional features from signals, avoiding the tedious manual feature extraction process in traditional methods and providing an efficient and robust solution. Therefore, CNN based circuit fault diagnosis technology has important theoretical significance and application prospects, and urgently needs in-depth research and practical verification.

2. The Working Principle of Convolutional Neural Network Algorithm in Fault Diagnosis

Convolutional neural networks (CNN), as a classic algorithm of deep learning, have been widely used in circuit fault diagnosis. The core principle is to extract local features from the input data through convolution operations, then reduce the dimensionality through pooling layers, and finally complete the classification task through fully connected layers [1]. In circuit fault diagnosis, CNN can directly learn effective fault features from raw signals or preprocessed signals, without relying on manual feature extraction. The convolutional layer filters the input data through convolutional kernels to generate feature maps that can capture subtle patterns of circuit faults. The pooling layer further compresses the data dimension and enhances the robustness of features by downsampling the feature maps. Finally, through multiple fully connected layers, the network completes the mapping from extracted features to fault type classification [2]. Figure 1 shows the basic structure of CNN, clearly indicating the relationship between convolutional layers, pooling layers, and fully connected layers. The mathematical expression for convolution operation is as follows [3]:

$$y(x,y)=(f*g)(x,y)=\sum_{m}\sum_{n}f(m,n)g(x-m,y-n)$$

Among them, f is the input signal, g is the convolution kernel, and * represents the convolution operation. By optimizing network parameters through backpropagation algorithm, CNN can automatically learn the optimal convolution kernel, improving the accuracy and robustness of fault diagnosis.

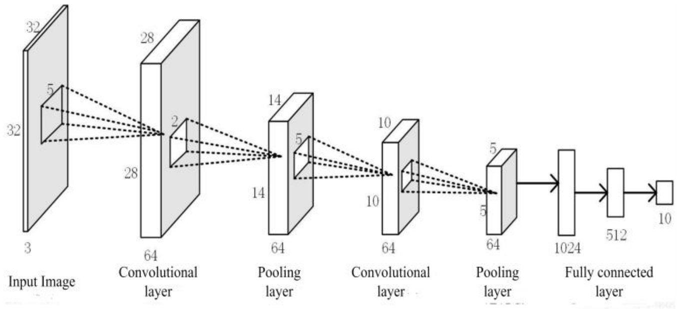

Figure 1. Basic structure of CNN

3. Circuit Fault Dataset and Preprocessing

3.1 Dataset Construction

In this study, the construction of the dataset is a key step in training the circuit fault diagnosis model, which mainly includes two aspects: data acquisition and label annotation. Firstly, in order to simulate different types of circuit faults, various common circuit fault data were generated through simulation testing, including short circuits, open circuits, poor contacts, and other fault types. The simulation data for each fault is based on different circuit operating states (such as voltage, current, power waveform, etc.) to fully cover various possible fault modes. In order to ensure the comprehensiveness and diversity of the dataset, circuit signals were also collected in a healthy (fault free) state as a control group for subsequent comparative analysis [4]. In the process of data acquisition, high-precision oscilloscopes and data acquisition cards were used to ensure the accuracy and reliability of the collected signals, and to truly reflect the circuit behavior under various fault conditions. For example, using the Tektronix MSO58 oscilloscope to collect circuit signals, the sampling rate reaches 10MS/s and the sampling accuracy is 14 bits, ensuring high-quality signal data. After the signal acquisition is completed, all fault data is accurately labeled and labeled according to different fault types. Table 1 shows the voltage and current waveform characteristics of different fault types and their corresponding signal features, including short-circuit faults, open circuit faults, and poor contact faults. It clarifies the characteristic waveforms and acquisition conditions of each fault, providing a rich and structured data foundation for subsequent model training.

TABLE I. Different Types of Faults And Their Corresponding Signal Characteristics

Fault Type	Voltage Signal Characteristics	Current Signal Characteristics	Remarks
Short Circuit	Decrease in peak value, waveform distortion	Sudden current increase	Common in power circuits
Open Circuit	Abnormal voltage fluctuations	Current is zero	Circuit interruption
Poor Contact	Frequent voltage fluctuations	Unstable current fluctuations	Signal interference

3.2 Data preprocessing methods

In the research of circuit fault diagnosis, data preprocessing is a key step to ensure the quality and reliability of model training data. Firstly, a bandpass filter is used to smooth the high-frequency noise in the original signal, removing unnecessary noise components and ensuring that the effective information of the fault characteristic signal is preserved. The design of bandpass filters aims to preserve fault characteristics within the frequency band while suppressing high-frequency interference signals, which is crucial for subsequent signal analysis and fault diagnosis. Secondly, due to the significant differences in amplitude and frequency of circuit signals, in order to avoid the impact of inconsistent scales of eigenvalues on model training, standardization methods were used to normalize the signals. Specifically, by normalizing the zero mean unit variance of each signal feature, the mean of the signal is adjusted to 0 and the standard deviation is adjusted to 1. The formula is as follows [5]:

$$x_{norm} = \frac{x-\mu}{\sigma}$$

Among them, x is the original signal value, and μ and σ are the mean and standard deviation of the signal, respectively. This method ensures that different signal features are processed at the same scale, effectively improving the training effectiveness and convergence speed of the model. In order to further enhance the robustness and diversity of the data, the study also adopted data augmentation techniques. Specifically, it includes adding random noise to the original signal and simulating possible signal interference in actual work; By using methods such as time shift and signal scaling, the diversity of the dataset can be increased and the generalization ability of the model can be improved. These preprocessing steps effectively improve the quality of the dataset, providing high-quality input data for the training of CNN models and ensuring the stability and accuracy of the training process.

3.3 Data augmentation techniques

In order to improve the generalization ability and robustness of the model, this study adopted various data augmentation techniques to increase the diversity of training data and enhance the model's adaptability to various complex working conditions and noise. Firstly, an additive white noise enhancement method was employed to address potential noise interference in circuit signals. This method simulates the signal noise environment that may occur in actual work by adding Gaussian noise with different amplitudes to the original signal. The formula is as follows[6]:

$$x_{aug} = x + \varepsilon$$

Among them, x is the original signal, and ϵ is Gaussian noise with a mean of 0 and a variance of σ 2. In this way, the robustness of the model in noisy environments is enhanced, enabling it to better handle interference signals in practical applications. Secondly, offset and scaling techniques on the timeline were employed to simulate the performance of circuit signals under temporal variations and amplitude fluctuations. The specific method is to randomly select the starting time point of the signal and linearly scale the signal with different amplitudes, which can simulate the signal changes under different workloads and enhance the model's adaptability to temporal fluctuations. Finally, in order to enhance the adaptability of the model to signals of different frequencies, frequency transformation techniques were introduced in the study. By performing Fourier transform on the signal and randomly selecting different frequency bands for modification, simulate the changes in signal characteristics under different frequency bandwidths. This technology effectively enhances the model's ability to recognize complex frequency distribution signals, enabling it to cope with frequency characteristic changes under different circuit faults. Through these data augmentation techniques, the research not only expands the scope of training data, but also improves the robustness and generalization ability of the model in complex and noisy environments, providing more robust training data for subsequent fault diagnosis.

4. Design of Circuit Fault Localization Model Based on CNN

4.1 Model Structure Design

The CNN model designed in this study adopts a multi branch parallel structure, including two feature extraction branches in the time domain and frequency domain. The time-domain branch consists of three convolutional blocks, each containing two convolutional layers and one max pooling layer. The convolution kernel sizes are 3 × 1, 5 × 1, and 7 × 1, respectively, to capture time-domain features at different scales [7]. In the frequency domain branch, the input signal is first subjected to Fast Fourier Transform (FFT) to convert the time-domain signal into a frequency domain representation, thereby capturing the frequency components in the signal. Then, frequency domain features are further extracted through two convolutional blocks. The features of these two branches will be combined through a feature fusion module to form a unified feature representation. Finally, fault classification will be performed through three fully connected layers to complete the final fault diagnosis task. In order to improve the robustness and generalization ability of the model, batch normalization and ReLU activation functions are added after each convolutional layer in the network structure, and Dropout layers (dropout rate of 0.5) are used between fully connected layers to effectively prevent overfitting. In addition, the input dimension of the model is 1 × 2048, indicating that each input signal contains 2048 sampling points, and the output dimension is the number of fault categories [8]. Table 2 provides a detailed list of the specific parameter configurations for the network structure. This multi branch design enables the model to analyze circuit signals from different perspectives in both time and frequency domains, improving the accuracy and reliability of fault diagnosis.

TABLE II. Configuration of CNN Model Structure Parameters

Layer	Input Dimension	Output Dimension	Kernel Size	Stride	Parameter Count
Conv1_1	1×2048	32×2046	3×1	1	128
Conv1_2	32×2046	32×2044	3×1	1	3,104
MaxPool1	32×2044	32×1022	2×1	2	0
Conv2_1	32×1022	64×1018	5×1	1	10,304
Conv2_2	64×1018	64×1014	5×1	1	20,544
MaxPool2	64×1014	64×507	2×1	2	0
Conv3_1	64×507	128×501	7×1	1	57,472
Conv3_2	128×501	128×495	7×1	1	114,816
MaxPool3	128×495	128×247	2×1	2	0
FC1	31,616	512	-	-	16,187,904
FC2	512	128	-	-	65,664
FC3	128	6	-	-	774

4.2 Loss Function and Optimization Algorithm

Considering the particularity of circuit fault diagnosis tasks, this study adopts an improved cross entropy loss function as the optimization objective of the model. The cross entropy loss function is a commonly used loss function in classification tasks, but due to the problem of class imbalance in fault datasets (some fault types have fewer samples while others have more samples), simply using the standard cross entropy loss function may result in poor performance of the model on a few types of faults. Therefore, this article introduces a category weight adjustment factor to solve the problem of sample imbalance. Specifically, the definition of the loss function is [9]:

$$L = -\sum_{i=1}^{C} w_i \cdot y_i \cdot \log(\hat{y}_i)$$

Among them, wi is the weight coefficient of the i-th class sample, yi is the true label, and α i is the model prediction probability. The weight coefficient is calculated by the reciprocal of the sample size [10]:

$$w_i = \frac{N}{n_i}$$

Among them, N is the total number of samples in the dataset, and ni is the number of samples in class i. This weighted cross entropy loss function effectively alleviates the negative impact of class imbalance, thereby improving the classification accuracy of minority class faults.

In terms of optimization algorithms, this study chose the Adam optimizer with momentum, which is a commonly used adaptive optimization algorithm that can automatically adjust the learning rate to accelerate convergence. The initial learning rate is set to 0.001 and the cosine annealing strategy is used to dynamically adjust the learning rate. Momentum parameters β 1=0.9, β 2=0.999, ϵ=1e-8. In order to avoid the occurrence of gradient explosion during the training process, this paper adopts gradient pruning technique to limit the gradient norm within the interval of [-5,5], ensuring the stability of gradient updates. In addition, to further alleviate the overfitting problem, the model also introduces an L2 regularization term with a regularization coefficient set to λ =0.0001, thus balancing the complexity and generalization ability of the model during the optimization process. Through these optimization measures, the model in this study can achieve better performance in complex circuit fault diagnosis tasks.

4.3 Model Training Process

The model training adopts batch stochastic gradient descent method, with batch size set to 64. The training process is divided into three stages: pre training, fine-tuning, and ensemble learning. During the pre training phase, a simulated dataset is used for initialization, with 100 epochs of training; During the fine-tuning phase, actual fault data is used for training, with 50 epochs of training; Finally, an ensemble learning strategy was adopted to train five models with different initialization parameters to form a model ensemble, and the generalization ability of the model was improved through a voting mechanism. During the training process, an early stopping strategy is adopted to prevent overfitting, and training is stopped when the validation set loss does not decrease for 5 consecutive epochs. To ensure the stability of training, a learning rate preheating strategy is adopted, gradually increasing the learning rate from 0 to the initial value within the first 5 epochs. The changes in key indicators during the training process are shown in Table 3. The model is trained on GPU using NVIDIA Tesla V100 accelerator, with an average training time of about 120 seconds per epoch.

TABLE III. Key Indicators of Model Training Process

Training Phase	Epoch	Training Loss	Validation Loss	Training Accuracy	Validation Accuracy
Pre-training	20	0.425	0.412	0.856	0.842
	40	0.308	0.325	0.912	0.895
	60	0.245	0.278	0.934	0.921
	80	0.198	0.256	0.945	0.928
	100	0.176	0.243	0.953	0.935
Fine-tuning	10	0.156	0.225	0.962	0.944
	20	0.134	0.212	0.968	0.951
	30	0.121	0.208	0.973	0.957
	40	0.115	0.205	0.975	0.960
	50	0.112	0.204	0.976	0.961

5. Experimental verification and performance analysis

5.1 Experimental Design

The experiment uses a workstation based on Intel Core i9-12900K processor as the basic hardware platform, equipped with 128GB DDR5 memory and NVIDIA Tesla V100 GPU accelerator. The software environment is based on Ubuntu 20.04 LTS operating system, using Python 3.8 as the development language and PyTorch 1.9.0 deep learning framework. The experimental data was collected using a Tektronix MSO58 oscilloscope, with a sampling rate set to 10MS/s and a sampling accuracy of 14 bits. To ensure the reliability of the experiment, a standardized circuit fault simulation testing platform is built, including three typical circuits: DC-DC converter, inverter, and rectifier. The test dataset contains 6 typical fault types, with 1000 sets of samples collected for each fault type and 500 sets of health status samples, for a total of 6500 sets of data. During the data collection process, different working environments are simulated by changing the load conditions (20%, 40%, 60%, 80%, 100% rated load) and operating temperatures (0 ℃, 25 ℃, 50 ℃). Table 4 provides a detailed list of the hardware and software configuration parameters of the experimental platform.

TABLE IV. Details of Experimental Platform Configuration Parameters

Configuration Category	Item	Parameter Specifications
Hardware Platform	CPU	Intel i9-12900K
	Memory	128GB DDR5-4800
	GPU	NVIDIA Tesla V100
	Storage	2TB NVMe SSD
Data Acquisition Device	Oscilloscope Model	Tektronix MSO58
	Sampling Rate	10MS/s
	Sampling Precision	14-bit
	Bandwidth	1GHz
Software Environment	Operating System	Ubuntu 20.04 LTS
	Programming Language	Python 3.8
	Deep Learning Framework	PyTorch 1.9.0
	Data Processing Library	NumPy 1.21.2

5.2 Model Performance Evaluation

The model evaluation adopts a five fold cross validation method, which comprehensively evaluates the performance of the model through indicators such as accuracy, precision, recall, and F1 score. At the same time, comparative experiments will be conducted between the proposed dual branch CNN model and traditional machine learning methods (SVM, random forest) as well as other deep learning models (LSTM, ResNet). The experimental results show that the model proposed in this paper outperforms the comparative methods in all indicators. Especially under complex working conditions, the model exhibits stronger robustness. Figure 2 shows the performance comparison data of different methods on the test set, and Table 5 presents the diagnostic accuracy data of our model for different types of faults. The experiment found that the model proposed in this paper has a high accuracy rate (both exceeding 98%) in identifying short-circuit faults and open circuit faults, while it is relatively low (94.2%) in identifying poor contact faults, mainly due to the strong randomness and uncertainty of the signal characteristics of poor contact faults.

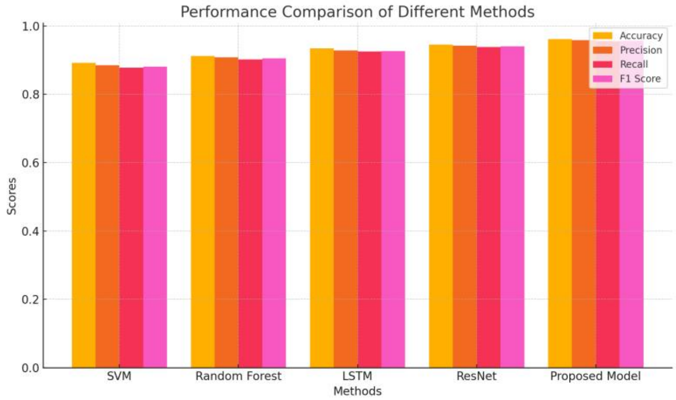

Figure 2. Performance Comparison of Different Methods

TABLE V. Diagnostic accuracy data for different types of faults

Fault Type	Training Set	Validation Set	Test Set
Short Circuit	0.989	0.983	0.981
Open Circuit	0.985	0.978	0.976
Poor Contact	0.952	0.945	0.942
Parameter Shift	0.968	0.962	0.958
Component Aging	0.973	0.965	0.963
Signal Interference	0.961	0.954	0.951

5.3 Algorithm Performance Analysis

In terms of algorithm performance analysis, the focus was on evaluating the real-time performance, robustness, and computational efficiency of the model. Real time testing shows that the average time for a single fault diagnosis is 12.5ms, which meets the requirements of real-time monitoring. By adding Gaussian white noise with different signal-to-noise ratios (-5dB to 20dB) to test the anti-interference ability of the model, the results showed that when the signal-to-noise ratio was higher than 5dB, the model could still maintain a diagnostic accuracy of over 90% (Figure 3). In terms of computational efficiency, the forward propagation time for a single batch (batch size=64) of the model on Tesla V100 GPU is 8.2ms, the backward propagation time is 15.3ms, the model parameter count is 16.4M, and the video memory usage is 2.8GB. To evaluate the adaptability of the model under different operating conditions, drift experiments were conducted to test the stability of the model performance by changing the load conditions and ambient temperature (Figure 4). Despite fluctuations in environmental load and temperature, the model can still maintain a high diagnostic accuracy, indicating that the model has strong adaptability and stability in practical applications, and can effectively cope with fault diagnosis tasks in different working environments. In addition, through sensitivity analysis of model parameters, it was found that the size of the convolution kernel and the depth of the network are key factors affecting the performance of the model, with the best performance achieved when the size of the convolution kernel is 5 × 1.

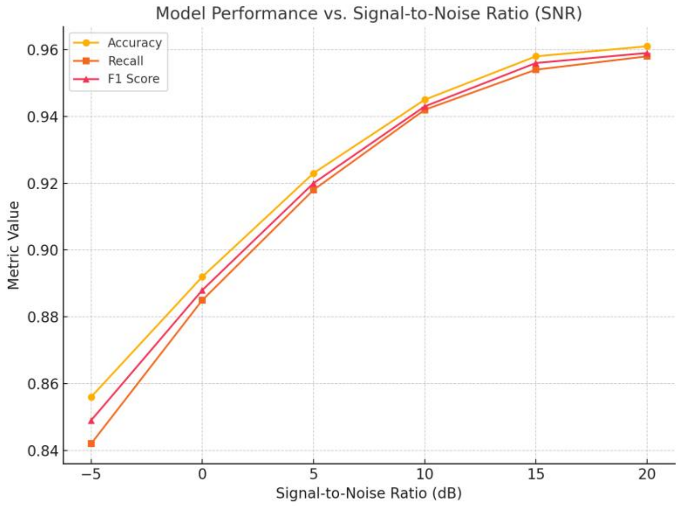

Figure 3. Model performance under different signal-to-noise ratios

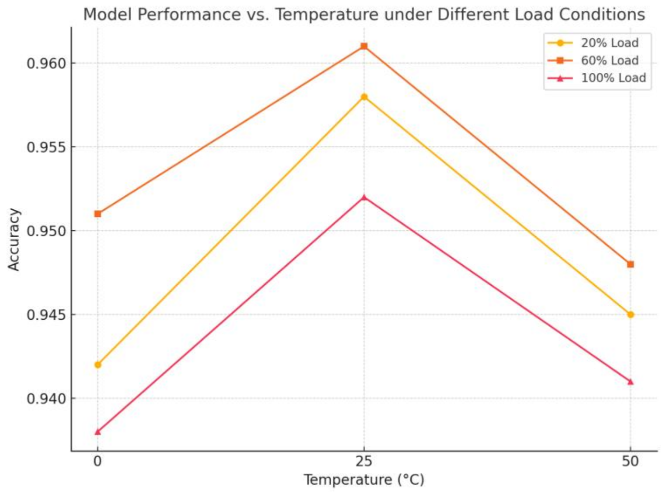

Figure 4. Diagnostic accuracy under different operating conditions

Conclusion

The circuit fault localization technology based on convolutional neural networks demonstrates the enormous potential of deep learning in circuit diagnosis, effectively improving the accuracy and robustness of fault recognition through the design of a multi branch structure. Although the model exhibits high diagnostic performance in most types of faults, there are still certain challenges when dealing with faults with strong signal feature randomness. Future research can further optimize the model structure and training strategies to cope with more complex fault modes and actual operating conditions, thereby improving the universality and real-time performance of the system.

References

[1] Zhao M, Barati M. A real-time fault localization in power distribution grid for wildfire detection through deep convolutional neural networks[J]. IEEE Transactions on Industry Applications, 2021, 57(4): 4316-4326.

[2] Zhao K, Di S, Li S, et al. FT-CNN: Algorithm-based fault tolerance for convolutional neural networks[J]. IEEE Transactions on Parallel and Distributed Systems, 2020, 32(7): 1677-1689.

[3] Liang J, Jing T, Niu H, et al. Two-terminal fault location method of distribution network based on adaptive convolution neural network[J]. IEEE Access, 2020, 8: 54035-54043.

[4] Moradzadeh A, Moayyed H, Mohammadi-Ivatloo B, et al. Turn-to-turn short circuit fault localization in transformer winding via image processing and deep learning method[J]. IEEE Transactions on Industrial Informatics, 2021, 18(7): 4417-4426.

[5] Khalil K, Eldash O, Kumar A, et al. Machine learning-based approach for hardware faults prediction[J]. IEEE Transactions on Circuits and Systems I: Regular Papers, 2020, 67(11): 3880-3892.

[6] Aziz F, Haq A U, Ahmad S, et al. A novel convolutional neural network-based approach for fault classification in photovoltaic arrays[J]. IEEE Access, 2020, 8: 41889-41904.

[7] Tsotsopoulou E, Karagiannis X, Papadopoulos T, et al. Advanced fault location scheme for superconducting cables based on deep learning algorithms[J]. International Journal of Electrical Power & Energy Systems, 2023, 147: 108860.

[8] Mirshekali H, Keshavarz A, Dashti R, et al. Deep learning-based fault location framework in power distribution grids employing convolutional neural network based on capsule network[J]. Electric Power Systems Research, 2023, 223: 109529.

[9] Paulachan P, Siegert J, Wiesler I, et al. An end-to-end convolutional neural network for automated failure localisation and characterisation of 3D interconnects[J]. Scientific Reports, 2023, 13(1): 9376.

[10] Tian X, Liu Z, Liu J, et al. Identification of overhead line fault traveling wave and interference clutter based on convolution neural network and random forest fusion[J]. Energy Reports, 2023, 9: 1531-1545.