Fire Detection Method Based on Improved Convolutional Neural Network with Random Inactivation

Yuanbin Wang, Yuanyuan Li, Huaying Wu, Yu Duan

***Abstract*—In view of the poor generalization ability in the process of fire detection, a fire detection method of the convolutional neural network (CNN) optimized by dropout based on inactivation probability is proposed. Firstly, CNN is trained by setting different inactivation probabilities for different convolution layers. Then the trained model is applied to the test set to determine the accuracy, false alarm rate, and accuracy rate. Multiple sets of data are obtained through repeated experiments. The inactivation probability of each convolution layer is finally predicted using a variety of data from the processes above. The optimal inactivation probability is used for optimizing the network to improve the generalization ability. Compared with other common methods, the proposed method has a lower false alarm rate and a higher detection accuracy under the same experimental settings. The accuracy of fire detection in various environments is above 91%, which effectively improves the generalization of the CNN. It plays a specific role in promoting fire detection research.**

***Index Terms*—image type, fire detection, convolutional neural network, dropout**

I. INTRODUCTION

Fire will seriously destroy the environment and the safety of human life. Timely fire detection is the key to fire prevention and control. At present, image-based fire detection is the research frontier in fire prevention. This method is not constrained by the scene or spatial distance and possesses the qualities of rapid response and high accuracy. In image-based fire detection, the method based on deep learning is the current research frontier which is applied widely since it has a higher detection accuracy and eliminates the need for human feature selection during the detection phase.

Many researchers have made exemplary achievements. Maria [1] proposed a transfer learning-based method to fire detection. It used the data enhancement technology under the tenfold cross-validation to enhance the data set and then utilized the transfer learning method to detect the fire image. This method has a fast detection speed, but the application scene is single.

Qin [2] proposed a detection model combining classification and target detection. This method utilized depth separable convolution to classify the fire image. Then the Yolo was used to determine the location information of the image whose detection result is fire. The detection accuracy is high, but the generalization ability of the model is poor.

Byoungjun [3] applied the Faster Region-based Convolutional Neural Network (R-CNN) to detect the suspected fire image according to the fire characteristics in the frame, which are summarized and accumulated through the Long Short-Term Memory (LSTM). The method achieves high detection accuracy, but poor generalization.

Li [4] transformed images into the HIS color space, and then the input image was trained and extracted with the Inception_Resnet_V2 network and deformable convolution network (DCN). SVM is eventually employed to detect the features extracted for fire detection. It has high detection accuracy, but the generalization ability is limited.

Zhong [5] proposed a new fire detection algorithm that first extracted the features of suspected fire areas in the image and then carried out fire detection on the extracted feature map according to CNN. The detection speed is fast, but the false alarm rate is high.

Chen [6] designed a cascaded CNN based on the LeNet [7]. This method uses dynamic and static characteristics to train two CNNs, respectively. The accuracy of the algorithm is high for fire detection with prominent dynamic features, but it is not ideal for fire detection with only static features. And the generalization ability is also insufficient.

Luo [8] employed a smoke detection algorithm based on the motion characteristics of smoke to identify fire characteristics in the suspected smoke regions. Then, CNN is used to extract the features for smoke identification automatically. The detection accuracy is high, but the generalization ability is poor.

Wang [9] extracted the suspected fire area by using the Haar feature and proposed a fire detection method using CNN to detect suspected fire areas. The detection rate of the model is high, but the false alarm rate is also high, and the model's generalization is weak.

Hu [10] built a multi-task framework based on the convolution network that combined inter-frame spatial information and motion information for fire detection. Although the model is easy to implement and has high accuracy detection speed, its generalization capacity falls short.

The majority of deep learning-based fire detection methods have high fire detection accuracy in a fixed scene but low detection rates in others, resulting in unsatisfactory generalization ability. These methods detect quickly but with low detection rates.

Manuscript received March 21, 2022; revised November 1, 2022. This work (Grants 61703329) was supported by National Natural Science Foundation of China and National Key Research and Development Program of Shaanxi Province, China(2019KW-046).

Yuanbin Wang is an Associate Professor of Electrical and Control Engineering, Xi'an University of Science and Technology, Xi'an 710054, China (e-mail: 13379232752@163.com).

Yuanyuan Li is a postgraduate student of Electrical and Control Engineering, Xi'an University of Science and Technology, Xi'an 710054, China (e-mail: 2764845392@qq.com).

Huaying Wu is a postgraduate student of Electrical and Control Engineering, Xi'an University of Science and Technology, Xi'an 710054, China (e-mail: 1661349504@qq.com).

Yu Duan is a postgraduate student of Electrical and Control Engineering, Xi'an University of Science and Technology, Xi'an 710054, China (e-mail: 307073082@qq.com).

In order to address the issues of weak generalization and low detection accuracy in the fire recognition process, this paper proposes a fire detection method of the CNN optimized by dropout based on the prediction of inactivation probability, implementation steps are as follows:

1) Firstly, the CNN model is established with four convolution and pooling layers.

2) Different inactivation probabilities are randomly selected for each convolution layer to train the network.

3) Then the trained model is tested on the test set to obtain the data such as test accuracy.

4) By repeating the above operation, multiple groups of data are used as the training data of the network to predict the inactivation probability of each layer.

5) Finally, the predicted optimal inactivation probability is substituted into the CNN model constructed for training, and the trained model is finally obtained as we need.

II. Fire Detection Model Based on Convolutional Neural Network

This work creates a CNN model whose structure is shown in Fig. 1 to detect suspected fire images based on the analysis of classical CNN models such as LeNet, AlexNet, and VGG. The specific composition of each layer of the network is as follows:

1) **Input layer:** The data set of 6600 images with pixel size converted into a format of 224 × 224 are sent to the input layer. The data set is divided into three test sets, as shown in Table I.

TABLE I
Division of the Data Sets

Data set	Fire image	Non-fire image
Training data set	2200	2200
Validation data set	800	800
Test data set	300	300

2) **Convolution layer:** To complete the operation of the first convolution layer, the input layer transmits the image to layer C1 which has the same mode convolution with a step size of 1. The acquired result is mapped to the feature maps using the ReLU activation function after a bias is added to the output result. Other convolution layers are created using the same procedures as above.

3) **Pooling layer:** The maximum pooling with a size of 2×2 and a step size of 2 is selected to down-sampling the output results of the convolutional layer.

4) **Fully connected layer:** In the model, there are two fully connected layers with 1024 neurons and 512 neurons, respectively. The fully connected layers classify the input image utilizing the learned features. The above operation is realized by the convolution, and then the prediction result is obtained through activation function processing.

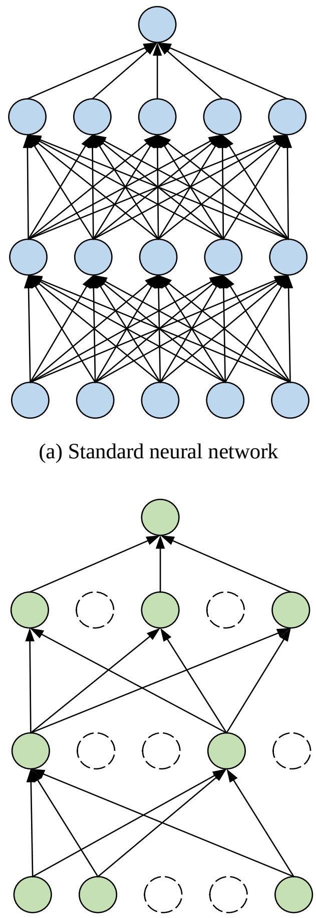
5) **Output layer:** The image classification studied in this subject is only to determine whether there is a fire, so there are only two nodes in the output layer, which are "1" (fire image) and "0" (non-fire image).

III. Regularization of the Dropout Network Based on the Prediction of Inactivation Probability

In deep learning, the trained model is prone to overfitting if there are too many parameters in the model and not enough training samples. To solve the issue mentioned above, many scholars have studied it including the dropout. It was proposed by Hinton [11], which can effectively alleviate the occurrence of overfitting.

Dropout is the most commonly adopted method to avoid overfitting in the neural network. The dropout network model makes each neuron randomly inactivated with probability p during training (set the neural node parameter to 0). All neural nodes return to normal during the test.

The working principle is shown in Fig.2 and the hollow dotted line in the figure represents the inactivated neurons.



(a) Standard neural network

(b) Neural network with dropout

Fig 2. Schematic diagram of neural network with dropout.

Random inactivation of neural nodes makes the inactivated nodes not work during training. Each training is equivalent to training a new network model, which can enhance the generalization ability to some extent. As shown in Fig. 3, the two layers network with three neural nodes in each layer is employed for illustration.

The neural network in Fig. 3 illustrates the dropout effect by randomly inactivating one neural node per layer. When each layer of the network randomly inactivates a neural node, the network evolves into a total of 9 different network structures, which is equivalent to training 9 other network models at the same time. It improves the generalization dramatically and reduces the possibility of overfitting.

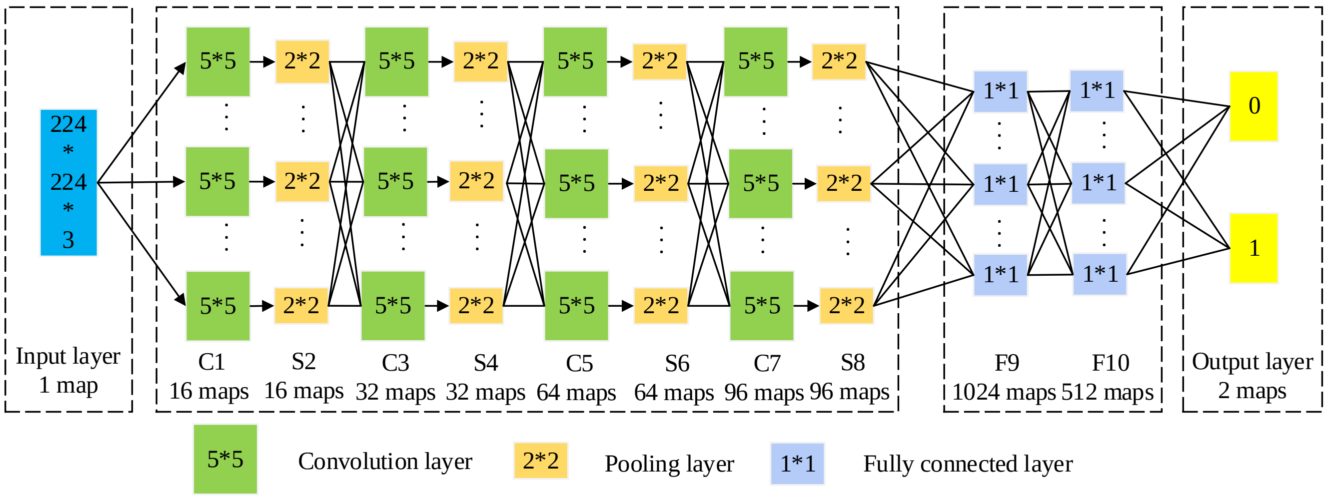


Fig 1. Fire image detection based on CNN.

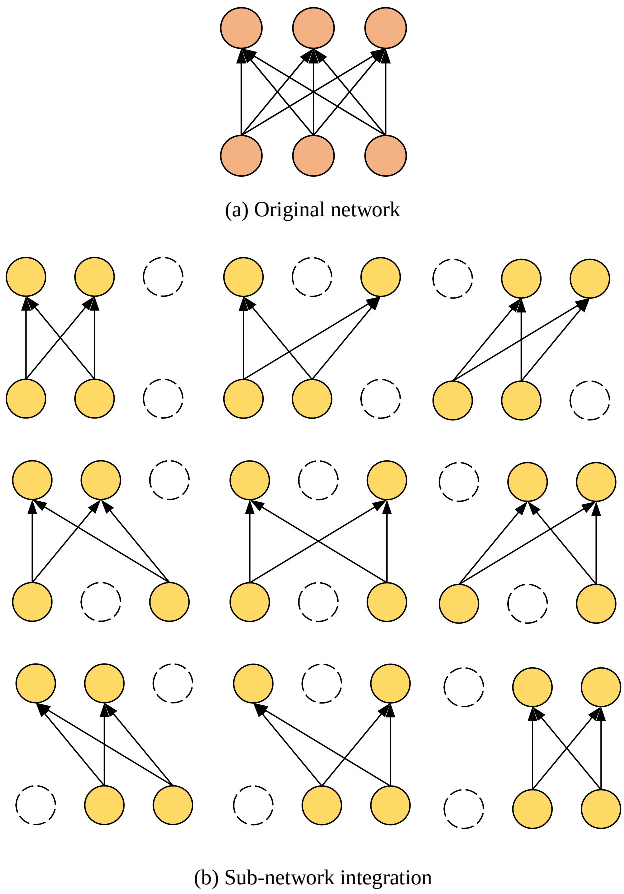


Fig 3. Schematic diagram of the dropout effect.

After selecting inactivation probability p, the standard dropout inactivates the neuron in each layer according to the probability p. Different values of p have different effects on the network model. Based on the standard dropout, an improved dropout with inactivation probability is proposed, the specific operation steps are as follows:

1) During the inactivation operation, CNN is trained by manually adjusting the inactivation probability p_i for the convolution layer i.

2) The test set is tested by the trained model to obtain the data including test accuracy rate q_j, false alarm rate w_j, accuracy rate f_j (the ratio of actual fire to the total number), and n_j (the ratio of the non-fire image to the total number). q_j, w_j, f_j and n_j are the data obtained from the j-th training.

3) Steps 1) and 2) are performed n times to obtain n sets of data used for estimating the inactivation probability of each layer and to acquire the best inactivation probability p_i'.

4) The predicted optimal inactivation probability is substituted into the model in step 1) for training to obtain the final trained model.

5) The trained model is tested by the test set to acquire the final detection results.

The flow chart of the algorithm is shown in Fig. 4.

The dropout based on the inactivation probability makes the best prediction for the inactivation probability value of each layer. Compared with the traditional dropout, which only performs the same inactivation probability on the whole network, this method refines the inactivation operation in each convolution layer. And the inactivation probability between different convolution layers is entirely independent. The inactivation probability predicted by the neural network is most suitable for the current CNN model and selected data set. It can promote the generalization of the network model better than the traditional dropout.

To avoid the inaccurate prediction caused by insufficient data and excessive workload caused by too much data, we collect 50 groups of training data to predict the inactivation probability. The specific steps of CNN with four-layer designed in this paper are described as follows:

1) The original CNN model is trained manually by setting the inactivation probabilities p_1, p_2, p_3, and p_4 of four convolution layers.

2) The trained network model is tested against the test set to obtain q_1, w_1, f_1, and n_1.

3) The above operations are repeated 50 times to achieve training data of the neural network to re-predict new inactivation probability p_1', p_2', p_3', and p_4'

4) The p_1', p_2', p_3', and p_4' are chosen as the best inactivation probability of each layer for training to obtain the final model.

5) The final trained model is used on the test set to obtain detection results.

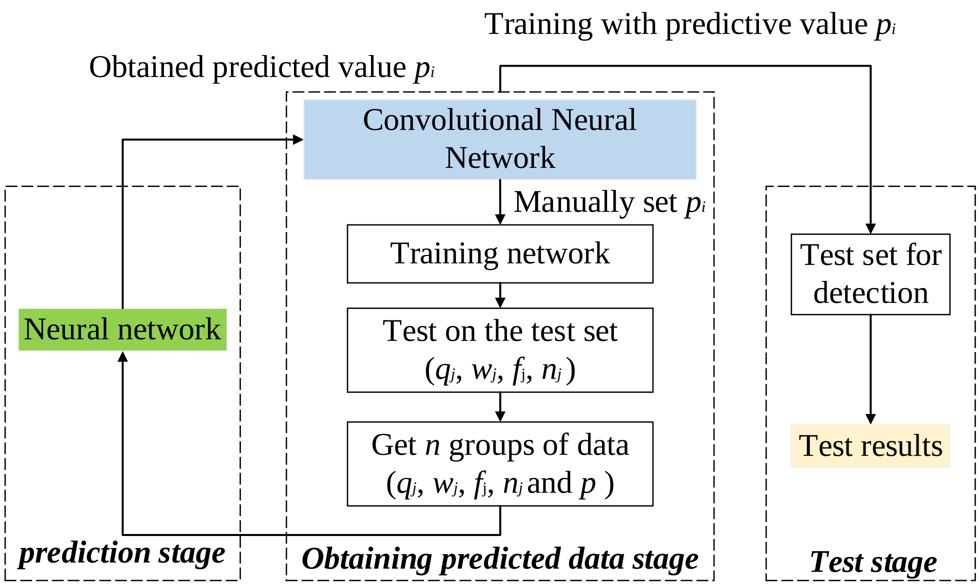


Fig 4. Flow chart of the improved dropout with inactivation probability.

IV. Experimental Results and Analysis

A. *Experimental environment*

The comparison and analysis of experimental results need to be carried out in the same environment. All environments in this section are shown in Table II.

TABLE II
EXPERIMENTAL ENVIRONMENT IN THIS SECTION

	Lab environment	Experimental conditions
Hardware	Operating system	Windows10
	GPU	NVIDIA GTX 1050Ti
	RAM	8G
Software	Frame	TensorFlow 2.3
	Development environment	PyCharm
	Programming language	Python3.7

B. *Hyperparameter settings*

The different selection of hyperparameters in deep learning algorithms has different effects on the running time, learning ability, and judgment ability of the model. In this paper, hyperparameters are set through several experiments.

Optimizer

At present, the commonly used optimizers for deep learning include SGD [12], Adagrad [13], and Adam [14]. We select Adam as the backpropagation algorithm. It has the advantage that the value of the learning rate calculated by Adam is within a definite range every time, which makes the neuron parameter update and the model optimization faster. The calculation formula is as follows:

$$g_t = \frac{\partial J(w)}{\partial w} \tag{1}$$

$$m_t = \beta_1 \cdot m_{t-1} + (1-\beta_1) \cdot g_t \tag{2}$$

$$v_t = \beta_2 \cdot v_{t-1} + (1-\beta_2) \cdot g_t^2 \tag{3}$$

$$\hat{m}_t = \frac{m_t}{1-\beta_1^t} \tag{4}$$

$$\hat{v}_t = \frac{v_t}{1-\beta_2^t} \tag{5}$$

$$w_t = w_{t-1} - \alpha \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \varepsilon} \tag{6}$$

Where $J(w)$ is the loss function, w is the weight; m_t and v_t denote the first-order and second-order moment estimation, respectively; $\hat{m}_t$ and $\hat{v}_t$ represent the first-order and second-order moment correction, respectively; β_1 is the exponential decay rate for the first estimate taken as 0.9, β_2 takes the value of 0.999 for the second estimated exponential decay rate; ε is set as 10^{-8} and α is the learning rate.

Initial learning rate

The different values of the initial learning rate *lr* can affect the change speed of the neuron weights. The initial learning rate can be determined by optimization or can be set artificially to a fixed value. With the increase in iteration number, the initial learning rate gradually decreases. When the initial learning rate is large, the model will keep oscillating, and it is difficult to converge. When the initial learning rate is small, the model converges slowly and the training time is long. In this paper, starting learning rate is set at 0.001 which produces the best network performance.

Batch size

Batch size specifies the number of images that the network can train at one time. The network training time is long when the batch size is low, and the network model is prone to overfitting. Large batch sizes result in much higher memory utilization, which prevents the model from adequately training. After several experiments, we set the batch size to 32, which is the best performance of the model.

C. *performance analysis*

According to the designed CNN, the network structure shown in Table III is set for experimental analysis of the performance of the final network.

TABLE III
NETWORK STRUCTURE

Network number	Network type	Feature map number	Nuclear size	Step size
Input	Input layer	3	—	—
C1	Convolutional layer	16	5×5	1
S2	Pooling layer	16	2×2	2
C3	Convolutional layer	32	5×5	1
S4	Pooling layer	32	2×2	2
C5	Convolutional layer	64	5×5	1
S6	Pooling layer	64	2×2	2
C7	Convolutional layer	96	5×5	1
S8	Pooling layer	96	2×2	2
F9	Fully connected layer	Number of neurons: 1024		
F10	Fully connected layer	Number of neurons: 512		
Output	Output layer	Number of output categories: 2		

In order to verify the influence of dropout based on inactivation probability on the CNN, 50 sets of p_1, p_2, p_3, and p_4 are manually set up to train the model. The trained models are tested against the test set to obtain the corresponding 50 sets of q_j, w_j, f_j, and n_j, respectively. The specific settings in training are shown in Table IV and the structure of the prediction network is shown in Fig. 5.

TABLE IV
SPECIFIC SETTING OF THE NETWORK

Input value	50 groups of q_j, w_j, f_j and n_j
Output value	50 groups of p_1, p_2, p_3 and p_4
Network settings	Number of neurons (in input layer): 4 Number of neurons (in output layer): 4 Number of neurons (in hidden layer): 9 Number of hidden layers: 1 Loss function: mean square error Training time: 50

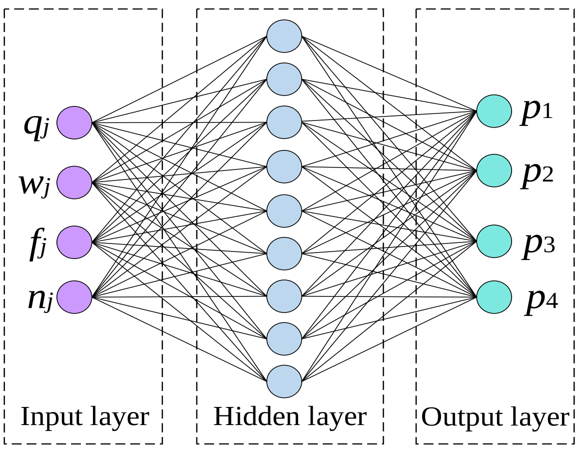


Fig 5. Structure diagram of prediction network.

The loss curve in the training process is shown in Fig. 6. With the iteration times increase, the loss decreases continuously. The minimum value unit of p_1', p_2', p_3', and p_4' are both 0.01 while the final prediction error is less than 0.001, so the error value has little impact on the prediction results. The prediction results are p_1'=0.62, p_2' =0.77, p_3' =0.82, and p_4'=0.96, which are used as the best inactivation probability with each layer of the CNN for the final training. And the resulting network model is tested against the test set to obtain the final detection results.

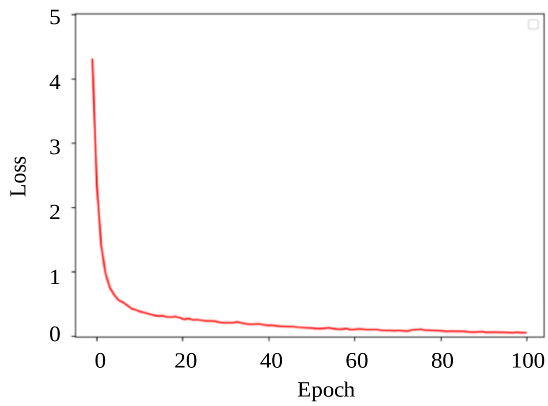
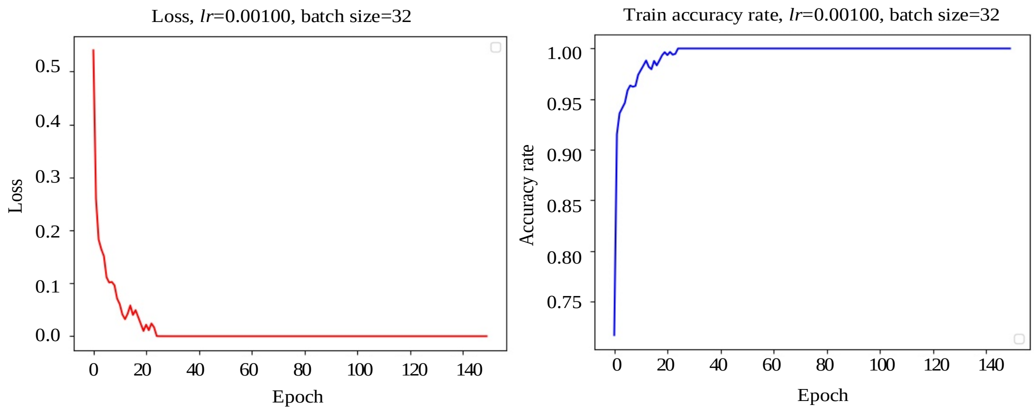


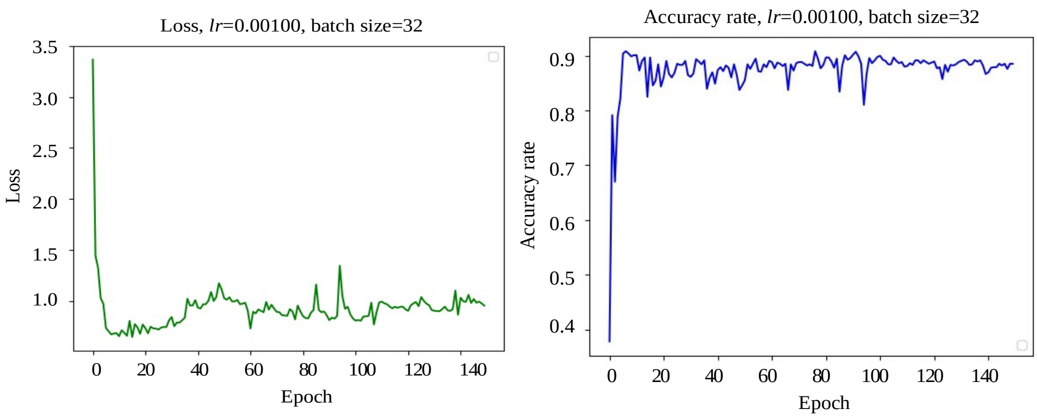
Fig 6. Loss curve of prediction network.

The network structure adopted for training is shown in Table III. The starting learning rate is set to 0.001, the batch size is set to 32, and the epoch is set to 150 according to the final convergence results of multiple experiments for relevant experimental verification.

Fig. 7 and Fig. 8 show the effects of two regularization methods, one based on the dropout, and the other based on the dropout with inactivation probability. Fig. 7(a) and Fig. 7(b) show the experimental results of the training set and validation set, respectively.

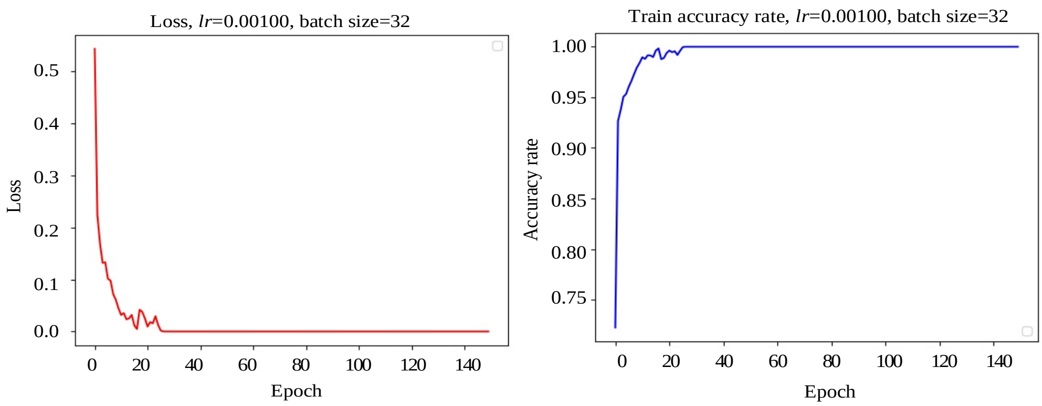


(a) Recognition results on test set

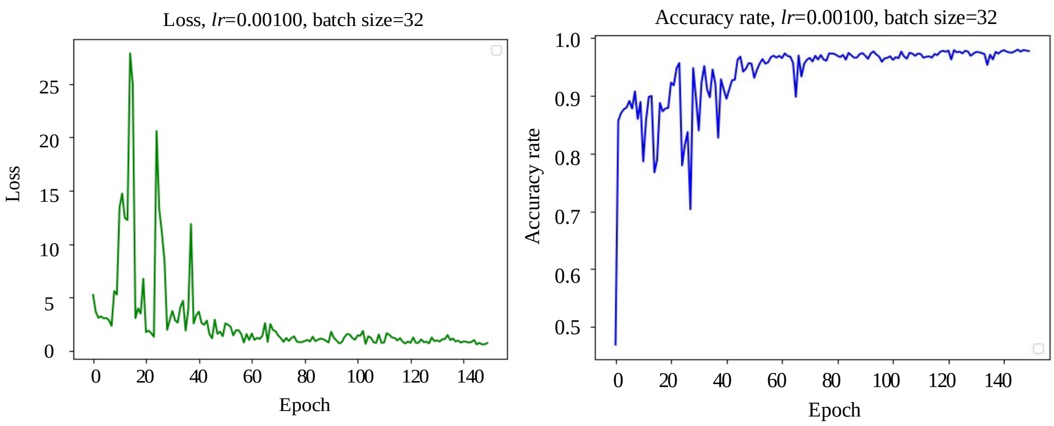


(b) Recognition results on validation set

Fig 7. Regularization results of dropout.



(a) Recognition results on test set



(b) Recognition results on validation set

Fig 8. Regularization results of dropout based on inactivation probability.

It can be observed from Fig. 7 (a) and Fig. 8 (a) that the accuracy of the CNN using dropout remains 100% after 25 times of training, while the accuracy of dropout predicted by inactivation probability keeps 100% unchanged after 23 times. As the result demonstrates that neither of the two models is overfitting, and the dropout training based on the prediction of inactivation probability is slightly better than the traditional dropout training.

The loss values and accuracy rate of the validation set during training are shown in Fig. 7(b) and Fig. 8 (b). In the figure, the accuracy of the convolution model based on dropout fluctuates around 90%, and the loss fluctuates around 1%. The accuracy of the convolution model based on dropout with random inactivation probability is about 97%, and the loss is much less than 1%. Since the prediction of the optimal probability for each convolutional layer produces the inactivation probability value most suitable for the network, the generalization is greatly improved, so it has better performance.

To further illustrate the superiority of dropout based on inactivation probability, the test data set is divided into three test sets, as shown in Table V.

TABLE V
DIVISION OF TEST DATA SETS

Test set	Specific composition
Test set 1	urban fire image: 100 urbans non-fire image: 100
Test set 2	forest fire image: 100 forest non-fire image: 100
Test set 3	fire image in the open environment: 100 non-fire image in the open environment: 100

Tests were conducted to compare four methods including the convolution network based on dropout, the convolution network based on dropout predicted by inactivation probability, the method in literature based on the color model [15], and the method in literature based on machine learning [16]. Accuracy and false alarm rate are selected as the objective evaluation indexes in the test, and the results take the average of three tests, as shown in Table VI.

From Table VI, the accuracy of dropout based on inactivation probability for three different test sets is higher than that of other methods, the proposed algorithm increases by 5.47%, 7.59%, and 5.85% on average, respectively. And the false alarm rate of the proposed algorithm is lower than others, which are reduced by 42.04%, 40.34%, and 46.90% on average, respectively. The results verify that dropout based on inactivation probability can improve the generalization ability of the model, which reflects the superiority of our method. Bold letters in parentheses after each indicator indicate the increased percentage compared to other methods.

The time-consuming comparison of different algorithms for fire detection is shown in Table VII.

TABLE VII
NETWORK STRUCTURE

Method	Time/Second
Literature [15]	0.84
Literature [16]	0.11
Dropout	0.14
Proposed method	0.13

In Table VII, the segmentation algorithm based on the literature [15] requires the longest time, taking 0.84 seconds to complete. The algorithm based on the literature [16] needs only 0.11 seconds. The proposed algorithm achieves significant performance improvement while losing only a little real-time performance. It takes 0.13 seconds, which is sufficient for real-time performance.

TABLE VI
NETWORK STRUCTURE

Test set	Detection method	Accuracy	False alarm rate
Test set 1	Literature [15]	86.63% (+**5.06%**)	5.23% (-**17.02%**)
	Literature [16]	88.15% (+**3.42%**)	5.76% (-**24.65%**)
	Dropout	84.32% (+**7.93%**)	9.53% (-**54.46%**)
	Proposed method	91.01%	4.34%
Test set 2	Literature [15]	88.65% (+**4.62%**)	3.68% (-**19.02%**)
	Literature [16]	86.35% (+**7.41%**)	4.65% (-**35.91%**)
	Dropout	83.43% (+**11.17%**)	8.79% (-**66.10%**)
	Proposed method	92.75%	2.98%
Test set 3	Literature [15]	87.86% (+**6.37%**)	4.26% (-**38.73%**)
	Literature [16]	90.34% (+**3.45%**)	3.87% (-**32.56%**)
	Dropout	86.75% (+**7.73%**)	8.53% (-**69.40%**)
	Proposed method	93.46%	2.61%

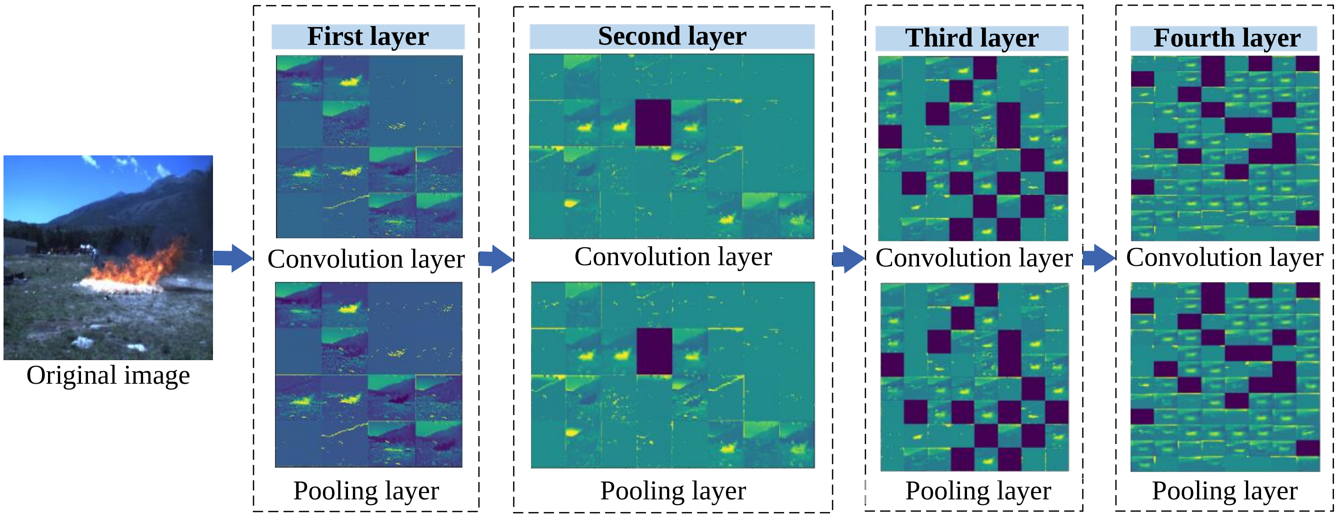


Fig 9. Hidden layer features of the original fire image in the network.

To verify the feasibility of the proposed algorithm, the middle layer features of CNN are extracted by the dropout network based on the prediction of inactivation probability. Fig. 9 shows the feature maps of the original image in the hidden layer during the training of CNN, and it can directly observe the changing trend of the fire image in each hidden layer of the CNN model. After the first layer of convolution, the fire region and background region features are extracted. Through the processing of the subsequent convolutional layer, more and more features of the fire area are gradually extracted. It can serve as evidence for the network model to determine whether the image is a fire image which fully demonstrated the viability of this experiment.

V. Conclusion

In order to improve the network generalization and recognition accuracy, the CNN model optimized by dropout based on inactivation probability is proposed for fire detection. This method improves the generalization ability of the network model by predicting the optimal inactivation probability of each convolution layer. Experimental results show that the proposed method has higher accuracy, better generalization, and a lower false alarm rate than other common methods, thoroughly demonstrating its viability and superiority.

References

[1] M. J. Sousa, A. Moutinho, and M. Almeida, “Wildfire detection using transfer learning on augmented datasets,” *Expert Systems with Applications*, vol. 142, no. 5, pp. 1-15, 2020.

[2] Y. Y. Qin, J. T. Cao, and X. F. Ji, “Fire detection method based on depthwise separable convolution and YOLOv3,” *International Journal of Automation and Computing*, vol. 18, no. 2, pp. 300-310, 2021.

[3] K. Byoungjun and L. Joonwhoan, “A video-based fire detection using deep learning models,” *Applied Sciences*, vol. 9, no. 14, pp. 1-16, 2019.

[4] J. Li, X. B. Qiu, and E. H. Zhang, “Fire recognition algorithm based on convolutional neural network,” *Computer Applications*, vol. 40, no. S2, pp. 173-177, 2020.

[5] Z. Zhong, M. J. Wang, Y. K. Shi, and W. L. Gao, “A convolutional neural network-based flame detection method in video sequence,” *Signal, Image and Video Processing*, vol. 12, no. 8, pp. 1619-1627, 2018.

[6] J. Z. Chen, Z. J. Wang, H. H. Chen, and L. Y. Zuo, “Dynamic Smoke Detection Using Cascaded Convolutional Neural Network for Surveillance Videos,” *Journal of University of Electronic Science and Technology of China*, vol. 45, no. 6, pp. 992-996, 2016.

[7] C. W. Zhang, X. Y. Yue, R. Wang, N. Li, and Y. Ding, “Study on traffic sign recognition by optimized Lenet-5 algorithm,” *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 34, no. 1, pp. 21-36, 2020.

[8] Y. M. Luo, L. Zhao, P. Z. Liu, and D. T. Huang, “Fire smoke detection algorithm based on motion characteristic and convolutional neural networks,” *Multimedia Tools and Applications*, vol. 77, no. 12, pp. 1-18, 2017.

[9] Z. C. Wang, Z. H. Wang, H. W. Zhang, and X. P. Guo, “A novel fire detection approach based on CNN-SVM using tensorflow,” *Lecture Notes in Computer Science*, Springer, Cham, vol. 10363, pp. 682-693, 2017. https://doi.org/10.1007/978-3-319-63315-2_60.

[10] Y. H and X. Lu. “Real-time video fire smoke detection by utilizing spatial-temporal convnet features,” *Multimedia Tools and Applications*, vol. 77, no. 22, pp. 29283-29301, 2018.

[11] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, “Improving neural networks by preventing co-adaptation of feature detectors,” *Computer Science*, vol. 3, no. 4, pp. 212-223, 2012.

[12] B. C. Zhou, C. Y. Han, and T. D. Guo, “Convergence of stochastic gradient descent in deep neural network,” *Acta Mathematicae Applicatae Sinica*, vol. 37, no. 1, pp. 126-136, 2021.

[13] D. Zhang, G. Z. Li, H. G. Wang, and J. N. Zhang, “Improvement of adaptive learning rate for pose estimation,” *Journal of Electronic Measurement and Instrument*, vol. 33, no. 6, pp. 51-58, 2019.

[14] K. S. Kim and Y. S. Choi, “HyAdamC: A new Adam-based Hybrid optimization algorithm for convolution neural networks,” *Sensors (Basel, Switzerland)*, vol. 21, no. 12, pp. 1-14, 2021.

[15] D. H. Lee, S. H. Lee, T. Byun, and N. I. Cho, “Fire Detection using Color and Motion Models,” *IEIE Transactions on Smart Processing & Computing*, vol. 6, no. 4, pp. 237-245, 2017.

[16] J. J. Mei and W. Zhang, “Early fire detection algorithm based on ViBe and machine learning,” *Acta Optics*, vol. 38, no. 7, pp. 60-67, 2018.