Fire recognition with convolutional neural network based on transfer learning

Xiang Liu¹, Guangcun Wei²*, Wansheng Rong³ and Xinguang Xiao⁴

¹College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao 266590, Shandong, China
²College of Intelligent Equipment, Shandong University of Science and Technology, City, Taian 271019, Shandong, China
³College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao 266590, Shandong, China
⁴College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao 266590, Shandong, China
*Corresponding author’s e-mail: weigc@sdust.edu.cn

Abstract. The fire recognition model based on deep learning can avoid many defects in the traditional method, but its construction requires a large amount of data to train the network parameters, and it takes a lot of time. In order to improve the accuracy of the model, this paper proposes a fire recognition model TNVGG-19 (Transfer learning + Newly fully connected layer module + VGG-19) with convolutional neural network based on transfer learning. First, we use the strategy of transfer learning to train the feature extraction network. Secondly, based on the VGG-19 model, this paper adds a newly designed fully connected layer module. Considering that flame data belongs to small sample data, we adopted a data augmentation strategy. Experiments show that the TNVGG-19 fire recognition model based on transfer learning proposed in this paper can effectively improve the accuracy of fire prediction and reduce the false alarm rate.

1. Introduction

Fire is a kind of natural disaster with strong destructiveness, great harm and strong suddenness. It can occur in many different environments, such as forests, private spaces and community public places. The hazard of fire not only manifests itself in destroying property and causing casualties, but also destroys the ecological environment [1]. It will burn down various buildings and production and living materials in the forest, and endanger human life. Whether such damage can be recovered requires considerable effort. By detecting fires in time through machine monitoring, we can take more targeted prevention and rescue measures. Therefore, it is of great practical significance for scholars to study the use of deep learning methods to build fire recognition models [2]. This paper proposes a fire recognition model TNVGG-19 with convolutional neural network based on transfer learning. Experiments show that under the condition of selecting the optimal parameters and data expansion, the accuracy of the model is significantly improved. The average accuracy rate of the final fire image recognition model reached 98.5%. The contributions of this article include the following:
This paper adopts a data expansion strategy to crop and flip the image, which can reduce the model's dependence on certain attributes. By increasing the number and diversity of fire samples, the generalization ability of the model is improved.

In order to improve the accuracy of the model, we combine transfer learning with CNN, and use a newly designed fully connected layer to replace the fully connected layer of VGG-19 to build a new fire recognition model. It can reduce the dimensionality of the input vector to remove redundant information, and input the reduced result into the classifier model for classification.

Since the selection of parameters and training methods have a great impact on the accuracy of the model, this paper conducts experiments by setting different learning rates and batch sizes and uses different training methods to test the performance of the model.

We have conducted an experimental comparison on the test set. And the results show that, compared with the previous method, the TNVGG-19 fire recognition model proposed in this paper has the highest accuracy.

2. Related work
With the continuous advancement of deep learning technology, image-based fire recognition technology has been rapidly developed. Researchers have proposed many different methods to improve accuracy of fire identification. Chen [3] et al. the flame's dynamic behavior and irregularity detection flame in RGB and HSI color spaces were studied. Celik and Demirel [4] et al. the classification rules were designed to distinguish the chrominance components and brightness features of YCbCr space, and the accuracy of flame detection improved by detecting fire images with small distance and large size. Zhou Bo long [5] et al. used smoke color, motion, area change rate and convex features to extract these features to detect smoke. Frizzi and Kaabi [6] et al. used a large number of samples to train a model with a 9-layer classification network, and then fine-tuned tens of layers of convolutional neural networks to achieve fire classification, further improving the network's recognition ability. Han [7] et al. proposed an effective algorithm that uses a combination of CNN and RNN in a continuous manner, and achieved good prediction results in various fire video scenes. Zeng [8] et al. constructed a flame smoke recognition model by improving the CNN-based object detection method. Some researchers combined transfer learning with the CNN model and apply the training-adjusted feature extraction network to the target datasets. After fine-tuning, the performance of the model has been significantly improved.

3. Data description
The original datasets used for model pre-training in this article is a large number of well-organized ImageNet datasets. The datasets contain not only fire scenes, but also images with similar flame color features. We need models to learn these features. For the target task datasets, this article uses the image query in the Google search engine to select images from the Web to generate the datasets. However, there are two problems with the fire sample datasets used for the target recognition task: The first is that the fire datasets are unbalanced because the number of total images for each class is different. Unbalanced datasets will cause the classifier to over fit to certain classes. We balance the datasets to better train the model by querying different types of fire scenes, including negative sample images with similar fire color scenes (hotel and street images, etc.). In addition, the number of fire samples is too small to train machine learning models. This paper proposes a data expansion strategy to increase the number of samples. The images in the training set are cropped and flipped, and then added to the target datasets for training. Finally, the total datasets are 8000 pictures. The fire sample datasets become more balanced because each class has the same number of images. There are 4000 positive samples and 4000 negative samples. Image samples are divided into training set (80%) and test set (20%).
4. Proposed approach

4.1. VGG-19 network

In this paper, we use the VGG-19 network model as a feature extraction network for transfer learning. The model is divided into 5 convolutional layers and 5 down-sampling layers, and finally three fully connected layers are used to classify high-level features. The network input is a 224×224 RGB image, and then it uses a 3×3 size convolution kernel and a 2×2 pooling kernel for training. The convolutional layer of the network learns the local features of the image through a convolution kernel to generate a feature map; the convolution operation is used to extract different features of the input, and its specific formula is as follows:

\[ Y_{\text{conv}} = f \left( \sum_{i=0}^{m} \sum_{j=0}^{n} x_{m(i,j) + j} \ast w_{ij} + b \right) \]  

(1)

Among them, \( m, n \) is the index of the feature map pixels. The value range of \( m \) is \((0, m)\), the value range of \( n \) is \((0, n)\). \( I, j \) is the size of the convolution kernel \( W \); "\( \ast \)" indicates the convolution symbol; \( f \) is the activation function, \( b \) is the additional paranoia, \( Y_{\text{conv}} \) is the output result of the convolution layer. The down-sampling layer is mainly to enhance the adaptability of the model to image transformations (such as displacement and rotation), while retaining the main features in the sample and reducing the main parameters. In the VGG-19 network, we select the maximum value in the image area as the combined value of the area. After the high-dimensional feature data is extracted by the convolutional layer and the pooling layer, the fully connected layer performs dimensionality reduction and tiling on the data, and then performs nonlinear transformation.

4.2. Designed new fully connected module

This paper designs a fully connected layer suitable for new tasks, replacing the original fully connected layer module, as shown in figure 1. After the input image is calculated by the convolutional layer and the pooling layer, the dimension is reduced by fully connecting layer 1. Then inputting the nonlinear activation function, the ReLU activation function is used to solve the problem of unstable gradient when the network layer is deep. Then enter the Dropout layer and randomly (temporarily) delete half of the hidden neurons in the network during the training process to reduce the dependence between neurons, which can well reduce the overfitting phenomenon generated by the model. When the fully connected layer 2 is input, the size is further reduced. Finally, we use the softmax function to calculate the final classification probability. The calculation formula is as follows:

\[ S_q = \frac{e^q}{\sum_{w=1}^{3} e^w} \]  

(2)

Where \( q \) represents the \( q \)-th component in the vector, \( S_q \) represents the classification probability of the \( q \)-th component, and \( w \) represents the serial number of the component.

![Figure 1. Flow chart of the new fully connected module](image-url)
4.3. Transfer learning and model building

The ordinary convolutional neural network model needs to be re-trained every time, wasting a lot of time. Therefore, we can use the method of transfer learning to adjust the model trained on a task experimentally to make the model suitable for a new task, thereby speeding up the learning efficiency and optimizing the network model. Although the fire sample datasets and the image content of ImageNet in this paper have some common flame features. But it also contains some differences. The abstract low-level features such as the edge and texture of the source datasets image belong to the invariant universality feature, and also have similar color features as the flame image. The VGG-19 network can learn the features of the data domain. Therefore, using the features learned from the source domain to transfer model parameters to deepen the research on fire recognition. The specific implementation steps of the algorithm are as follows, as shown in figure 2.

(1) Build a fire experiment database. Collect fire and interference fire images, and expand the data to increase the number of samples to build a database.

(2) Train the VGG-19 network on the ImageNet datasets, and delete the fully connected layer after the training is complete. We replaced the fully connected layer module of VGG-19 with a newly designed fully connected layer, and established a new fire identification model TNVGG-19.

(3) Load the weights and parameters of the already trained VGG-19 convolutional layer into the convolutional layer of the newly constructed model (TNVGG-19).

(4) Finally, we use the collected fire sample to train a new model, explore the effects of parameters and training methods on the model's performance, and analyse the experimental results.

5. Case studies and results

5.1. The influence of learning rate on model performance

The value of the learning rate directly affects the iterative update of the weights, which affects the size of the neuron parameters, which will ultimately affect the performance of the entire network structure. In order to analyze the impact of learning rate on network performance, this article first sets the initial learning rate to 0.1 and keeps other parameters unchanged. Then reduce the learning rate of each experiment. Experiment results on the training set are shown in figure 3.
Obviously, we can see that when the learning rate is set to 0.1, too large will cause the parameter update to be too fast, the gradient of the loss function value will explode, and the final training accuracy is very low. When the learning rate is 0.01, the model is unstable. When the initial learning rate is set to 0.0001, too small will cause the network to not converge in 2000 iterations, and the convergence speed is very slow. When the initial learning rate is 0.001, the network convergence speed is faster, the accuracy rate achieves the best effect in the training process, and the generalization ability is the strongest. Considering time and performance, we set the initial learning rate to 0.001.

![Figure 3: The model training under different parameter settings. (a) represents the accuracy change curve of different learning rates. (b) represents the change curve of the loss function value of different learning rates](image)

**5.2. The influence of batch size on model performance**

Batch size refers to how many pictures are read at a time for training, and the degree and speed of model optimization are determined during the training process. Therefore, we need to conduct experiments and use different batch sizes to train 500 fire data and 500 non-fire sample images on the model to find the best parameter selection.

| Batch size | 1   | 8   | 32  | 64  | 128 |
|------------|-----|-----|-----|-----|-----|
| Total epochs | 500 | 500 | 500 | 500 | 500 |
| Total iterations | Cannot | | | | |
| Time of one epoch(s) | converge | | | | |
| Final training error(500epochs) | 0.0028 | 0.0015 | 0.0031 | | |
| Test score(%) | 64.8% | 83.5% | 45.6% | | |

Experiments show when the batch volume is 1 and 8, the insufficient processing volume cannot converge within 500 epochs; as the batch size increases from 32 to 64, the time required to process the same amount of data will also increase. The number of periods required for the network to reach the same accuracy will increase as shown in table 1. When the batch size is 128, the model performance is not as good as before. Considering the training time of the network and the score of the network on the test set, we chose a batch size of 64 for training. In addition, we use two different methods to train the model for testing. The results show that there are significant differences in the indicators of the two training methods on the test samples. In contrast, the fire recognition model used to train all levels shows higher accuracy and lower false alarm rate on the test set, and has better results.
Table 2. The effect of the model on the test set under different training methods.

| Model indicators                      | FNR  | FPR  | Recall | Accuracy |
|---------------------------------------|------|------|--------|----------|
| Only train fully connected layers     | 0.05 | 0.18 | 0.95   | 0.72     |
| Train all layers                      | 0.02 | 0.01 | 0.98   | 0.985    |

\(^a\) is the proportion of negative samples being mistaken for positive samples.
\(^b\) is the proportion of positive samples being mistaken for negative samples.
\(^c\) represents the probability that all positive samples are divided into positive samples.

5.3. Comparison of results with other methods

The fire recognition model TNVGG-19 proposed in this paper is compared with the previous method in the same test set, as shown in table 3. Model A represents the VGG-19 model after transfer learning. Model B is a model obtained by training the VGG-19 network using the fire datasets after data expansion.

Table 3. Model comparison results.

| Methods   | Mean accuracy(%) |
|-----------|------------------|
| VGG-19    | 72.8%            |
| Model A   | 94.6%            |
| Model B   | 93.2%            |
| TNVGG-19  | 98.5%            |

In contrast, the TNVGG-19 model has an average accuracy rate of 98.5%, achieving the best results. The fire recognition model proposed in this paper is slightly better than them, which shows that the model has good potential in fire recognition.

6. Conclusions and future research

In this article, we propose a fire recognition model TNVGG-19 with convolutional neural network based on transfer learning. We use a data expansion strategy to solve the problem of insufficient fire samples. Experiments show that transfer learning and the newly designed fully connected layer effectively improves the performance of the model. In subsequent research, the depth of the current fire recognition model can be further adjusted to add more datasets, judge video images more accurately and handle more complex situations.

References

[1] Yang, G., and Di, X. (2011) Adaptation of Canadian Forest Fire Weather Index system and its application. In: IEEE International Conference on Computer Science and Automation Engineering. Shanghai. pp.55-58.
[2] Chun, Y. Y., Yong, M. Z., Jun, F. and Jin, J. W. (2009) Texture Analysis of Smoke for Real-Time Fire Detection. In: Second International Workshop on Computer Science and Engineering. Qingdao. pp.511-515.
[3] Chen, T. H., Wu, P. H. and Chiou, Y. C. (2005) An early fire-detection method based on image processing. In: IEEE International Conference on Image Processing. Taiwan. pp.1707-1710.
[4] Celik, T., Demirel, H. (2009) Fire detection in video sequences using a generic color model. Fire Safety Magazine. Volume 44, pp.147-158.
[5] Zhou, B. L., Song, Y. L., Yu, M. H. (2016) Fire smoke detection algorithm based on image disposal. Fire Science and Technology Magazine. pp.390-393.
[6] Frizzi, S., Kaabi, R., Bouchouicha, M. (2016) Convolutional neural network for video fire and smoke detection. In: IEEE Conference on Industrial Electronics Society, IECON. Florence. pp. 877-882.
[7] Han, Z. C., Jin, G., X., Lee C., Han, Y. G. (2019) Predictive Models of Fire via Deep Learning Exploiting Colorific Variation. In: 1st International Conference on Artificial Intelligence in Information and Communication. Okinawa. pp. 579-581.

[8] ZENG, J., LIN, Z. (2018) An improved object detection method based on deep convolution neural network for smoke detection. In: International Conference on Machine Learning and Cybernetics (ICM-LC). Chengdu. pp. 184-189.