Experimental Discussion on Fire Image Recognition Based on Deep Learning

Yongyi Cui* and Fang Qu
College of Safety Engineering, Shenyang Aerospace University, Shenyang, China

*Corresponding author e-mail: 20052696@sau.edu.cn

Abstract. Fire detection technology based on video images is an emerging technology that has its own unique advantages in many aspects. With the rapid development of deep learning technology, Convolutional Neural Networks based on deep learning theory show unique advantages in many image recognition fields. This paper uses Convolutional Neural Networks to try to identify fire in video surveillance images. This paper introduces the main processing flow of Convolutional Neural Networks when completing image recognition tasks, and elaborates the basic principles and ideas of each stage of image recognition in detail. The Pytorch deep learning framework is used to build a Convolutional Neural Network for training, verification and testing for fire recognition. In view of the lack of a standard and authoritative fire recognition training set, we have conducted experiments on fires with various interference sources under various environmental conditions using a variety of fuels in the laboratory, and recorded videos. Finally, the Convolutional Neural Network was trained, verified and tested by using experimental videos, fire videos on the Internet as well as other interference source videos that may be misjudged as fires.

Keywords: Deep Learning, Neural Networks, Image Recognition, Fire Recognition, Video Image

1. Introduction
Image-based fire detection methods mainly fall into two categories: fire detection based on traditional image processing and fire detection based on deep learning. The former mainly combines image processing and pattern recognition technology. Firstly, the suspected flame area is segmented, then feature extraction is performed on the area, and finally the pattern recognition technology is used to classify and recognize these features to complete the fire detection work [1]. The latter is to design a network model in advance, and then directly send the original image data to the network model for training, and use the trained model to identify the test data. Using traditional image processing technology to identify a fire, its features are mainly extracted by manual experience or feature conversion methods [2]. Feature processing generally requires manual intervention, and the use of human experience to select good features will limit the recognition accuracy due to the reasonability of feature selection [3, 4].

Deep learning is a sub-problem of machine learning, whose main purpose is to automatically learn
effective feature representations from data [5]. Fire detection algorithms based on deep learning are more concise in steps than fire detection methods based on traditional image processing. Applying deep learning technology on the field of image recognition can avoid blindness in the feature extraction process to a large extent, and theoretically can extract more and deeper features, which greatly improves the accuracy of fire image recognition. In recent years, many researchers have applied Convolutional Neural Network technology to fire image recognition work [6].

2. Image Preprocessing

The environment of fire is relatively complex, and the collected images will be subject to various noise pollution and some degree of damage, so image preprocessing is needed to reduce the impact on the image [7]. In this paper, we use image enhancement method for preprocessing. Image enhancement can highlight some information in an image, and shrinking, suppress or even filter out the unwanted information, thus the useful information in the image is enhanced, the useless information is reduced, and the sharpness of the image can be greatly improved. The purpose of fire image enhancement is to highlight the features of the flame area in the image, remove or weaken the area information that may interfere with the flame in the background, and make it more suitable for subsequent recognition.

In this paper, the fire image is preprocessed by smoothing and sharpening methods. Median filtering and mean filtering are two commonly used methods for fire image smoothing [8]. In the fire image, there are often sporadic sparks around the flame area, which will make the subsequent fire identification difficult. Through experiments, median filtering can effectively remove this part of noise interference. But after smoothing and filtering, the contour of the image is not obvious and the boundary is blurred. In order to enhance the edge of the image and make the edge of the image clear, we use image sharpening technology to enhance the detailed description of the fire image. By using the frequency characteristics of the image, a high-pass filtering method is used to filter out low-frequency components to achieve sharpening of the fire image.

3. Deep Learning Recognition Method

There are many types of deep learning methods. Among them, Convolutional Neural Network (CNN) has outstanding advantages in original image processing and is a typical feedforward artificial neural network. Convolutional Neural Network, as a pattern recognition method that combines deep learning and artificial neural network, has a good performance in image classification due to its simple structure and strong adaptability [9]. In addition, the Convolutional Neural Network adopts a weight sharing mechanism, which makes the calculation simple and efficient, reduces the complexity of the network, and has a high recognition rate and wide practicability. In this paper, we use Convolutional Neural Network for fire recognition. The Convolutional Neural Network is mainly composed of two parts: feature learning and classification recognition. A common Convolutional Neural Network consists of an input layer, a convolutional layer, an activation layer, a pooling layer, a fully connected layer and the final output layer.

3.1 Convolutional Layer

In the convolutional layer, the feature maps connected between layers are extracted by one or more convolution kernels to extract pixel-level image features through convolution operations [10]. In mathematics, convolution is a mathematical operator that generates the third function through two functions f and g, and formula (1) is a mathematical expression of its definition.

\[
y'(x) = f(x) * g(x) = \int_{-\infty}^{\infty} f(\tau)g(x - \tau)d\tau
\]

In image processing, a digital image can be regarded as a discrete function in a two-dimensional space. When a two-dimensional image is used as input, the corresponding convolution operation can be expressed by formulas (2) and (3):
In the formula above, \( f(x, y) \in \mathbb{R}^{M \times N} \) represents the input image, \( w(x, y) \in \mathbb{R}^{U \times V} \) represents the convolution kernel, and \( U \times V \) is the size of the kernel, generally \( U \ll M \), \( V \ll N \). \( g(x, y) \) represents the output image, and \( g_{i,j} \) represents an element in the output image.

The convolution operation is equivalent to filtering the image with a trainable convolution kernel. Suppose the size of an image is \( M \times N \), and the size of the convolution kernel is \( U \times V \). In calculation, the convolution kernel is multiplied by each \( U \times V \) size image region of the image, which is equivalent to extracting the image features of this \( U \times V \) region. The local features of the image are extracted bit by bit, and a new two-dimensional feature image will be mapped out every time after each filtering of the convolution kernel. The image input to the computer is essentially a matrix of numbers representing the color of each pixel, and the entire operation process of the Convolutional Neural Network is essentially a matrix operation. Usually a small size matrix of 3×3 or 5×5 is selected as the convolution kernel. The operation of the convolution operation starts from the upper left corner of the image matrix, and finds an area of the same size as the convolution kernel, each pixel in this area is multiplied by the value of the convolution kernel at the corresponding position and then summed to complete a convolution operation. Let the convolution kernel slide on the input picture with a fixed step length successively, and the sliding direction is from left to right and from top to bottom. Each time it slides, the convolution kernel does a dot product on the input image corresponding to the position of the sliding window and obtains a value. After completing the convolution operation of the entire picture, a new picture will be generated finally, and this new picture is the feature map.

Before the convolution of the input image, there are two boundary pixel filling methods to choose from, namely Same and Valid [11]. The Valid method is to directly convolve the input image without any pre-processing and pixel filling of the input image. The disadvantage of this method is that some pixels in the image may not be captured by the sliding window; the Same method is to add a pixel boundary with a specified number of layers which values all being 0 to the outermost layer of the input image, so that all the pixels of the input image can be captured by the sliding window. After the input image undergoes a round of convolution operation, the width and height of the output image can be calculated by formula (4) and (5):

\[
W_{\text{output}} = \frac{W_{\text{input}} - W_{\text{filter}} + 2P}{S} + 1 \tag{4}
\]

\[
H_{\text{output}} = \frac{H_{\text{input}} - H_{\text{filter}} + 2P}{S} + 1 \tag{5}
\]

Where, \( W \) and \( H \) respectively represent the width (Weight) and height (Height) of the image. The subscript input represents the relevant parameters of the input image: the subscript output represents the relevant parameters of the output image; the subscript filter represents the relevant parameters of the convolution kernel: \( S \) represents the step size of the convolution kernel; \( P \) (is the abbreviation of Padding) represents the number of boundary pixel layers added to the edge of the image. If the Same method is selected as the boundary pixel filling method of the image, then the value of \( P \) is equal to the number of boundary layers added to the image. If the Valid method is selected, then \( P=0 \). Figure 2 and figure 3 show the feature maps obtained by using two common operators as convolution kernels.
for convolution calculations.

Figure 1. Original image
Figure 2. Gausse operator processed image
Figure 3. Laplacian operator processed image

3.2 Activation Layer
The role of the activation layer is to add an activation function between all hidden layers. After adding appropriate activation function, the function of the neural network will become more powerful. Generally, non-linear functions are used as activation function, because in practical applications, most of the data are non-linearly distributed. Only in this way can the deep neural network have the ability of non-linear mapping learning. There are many commonly used activation functions, and whether the selected activation function is appropriate has a great impact on the final deep neural network model. This paper chooses the Sigmoid function and the tanh function as the activation function. The mathematical expression of Sigmoid function is shown in formula (6), and the mathematical expression of tanh function is shown in formula (7).

\[ f'(x) = \frac{1}{1 + e^{-x}} \]  
(6)

\[ f'(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \]  
(7)

The main advantage of Sigmoid as an activation function is that a series of processes from input to output after activation by the Sigmoid function are very similar to the working mechanism of biological neural networks. However, the shortcomings of Sigmoid as an activation function are also very obvious. The value interval of the Sigmoid derivative is (0, 0.25). If the model level reaches a certain depth, the backward propagation will cause the gradient value to become smaller and smaller until the gradient disappears. The derivative range of the tanh function becomes larger, and the problem of vanishing gradient is alleviated between (0,1). But the computation is much larger than that of Sigmoid, and there is still a possibility that the gradient disappear.

Faced with a variety of activation functions, there is currently no universally accepted conclusion on which one is more suitable. Generally, it should be analyzed in detail according to the specific characteristics of the research problem, and selection should be made according to it. We need a lot of modeling work to explore, which is also a focus of this paper. We try to build models with different activation functions in order to find the activation function that is most beneficial to our work.

3.3 Pooling Layer
The pooling layer compresses the information of the original feature layer, which is an important step in the Convolutional Neural Network. Pooling can be seen as a way to extract the core features of input data in Convolutional Neural Networks, which not only realizes the compression of the original data, but also greatly reduces the parameters involved in model calculations. There are many methods of pooling, the most commonly used are the average pooling layer and the maximum pooling layer.
The input data processed by the pooling layer is generally a feature map generated after a convolution operation. The pooling layer also needs to define a sliding window similar to the convolution kernel in the convolution layer, but this sliding window is only used to extract important features in the feature map and has no parameters itself. For example, select a $2 \times 2$ window to slide on the input feature image in a certain step, and for each slide, take the maximum value of the number in the four small windows as the output result of this part. After completing the sliding value of the entire image, a new target feature map is obtained. This is the maximum pooling layer method. As to the average pooling layer, is to add all the numbers in the window and then average them, and use this value as the final output result.

Generally, pooling layers are inserted between successive convolutional layers. Deep learning models need to complete training on a large amount of data, which will lead to serious over-fitting problem if it is fully trained. The pooling layer can merge similar features, reduce the feature vector output by the convolutional layer through the pooling operation, and effectively preventing over-fitting. The width and height of the input feature map after a round of pooling operation can be calculated using formula (8) and (9). Figure 4 shows the result obtained after pooling the convoluted feature image.

$$W_{output} = \frac{W_{input} - W_{filter}}{S} + 1$$

$$H_{output} = \frac{H_{input} - H_{filter}}{S} + 1$$

Where, $W$ and $H$ respectively represent the width and height of the feature map, the subscript $input$ represents the related parameters of the input feature map, the subscript $output$ represents the related parameters of the output feature map, and the subscript $filter$ represents the related parameters of the sliding window. $S$ represents the step size of the sliding window, and the depth of the input feature map is consistent with the depth of the sliding window. Figure 4 shows the result obtained after pooling on the convoluted feature image.

![Maximum pooling on Figure 2](image1)

Maximum pooling on Figure 2

![Average pooling on Figure 2](image2)

Average pooling on Figure 2

![Maximum pooling on Figure 3](image3)

Maximum pooling on Figure 3

![Average pooling on Figure 3](image4)

Average pooling on Figure 3

**Figure 4.** Feature maps under different image pooling method

Pooling is equivalent to compressing the image according to some algorithm. Generally speaking, the feature data obtained by mean pooling is more sensitive to background information, while the
feature data obtained by maximum pooling is more sensitive to texture information.

### 3.4 Fully Connected Layer

The fully connected layer is essentially equivalent to a classifier. After multiple convolutional layers and pooling layers, one or more fully connected layers are connected [12]. The essence of the convolutional layer before the fully connected layer is to extract features, while the function of the fully connected layer is to classify. After many times of convolution, activation and pooling, the feature learning part will output many feature maps, and each feature map has only one important feature of the overall image. This requires adding one or more fully connected layers to integrate all the features. Logistic regression is used to map the feature vector extracted from the upper layer into a feature vector according to different weights. This vector summarizes all the feature information of the overall image, after activation, it enters the classification layer as input. After statistical calculation in the classification layer, the probability that the input image belongs to a certain category is output to complete the image recognition.

### 4. Experiments and Conclusions

There are many frameworks for deep learning, and the most influential ones in recent years are PyTorch, TensorFlow, Keras, Sklearn, etc. PyTorch is a brand new deep learning framework rewritten by Facebook using Python based on the deep learning framework Torch, it inherits many advantages of NumPy and also supports GPUs computing, it also has more obvious advantages than NumPy in terms of computational efficiency. In addition, it supports dynamic calculation graphs, which is its biggest advantage. This paper chooses the PyTorch framework to build a Convolutional Neural Network for fire recognition. The CNN network structure used in this paper is shown in Table 1.

**Table 1. CNN network structure**

| Layer     | Kernel  | Stride | Output size  |
|-----------|---------|--------|--------------|
| INPUT     | 224×224×3 |        |              |
| Conv1     | 3×3×3  | 1      | 224×224×64   |
| Conv2     | 3×3×64 | 1      | 224×224×64   |
| MaxPool1  | 2×2×64 | 2      | 112×112×64   |
| Conv3     | 3×3×64 | 1      | 112×112×128  |
| Conv4     | 3×3×128| 1      | 112×112×128  |
| MaxPool2  | 2×2×128| 2      | 56×56×128    |
| Conv5     | 3×3×128| 1      | 56×56×256    |
| Conv6     | 3×3×256| 1      | 56×56×256    |
| MaxPool3  | 2×2×256| 2      | 28×28×256    |
| Conv7     | 3×3×256| 1      | 28×28×512    |
| Conv8     | 3×3×512| 1      | 28×28×512    |
| MaxPool4  | 2×2×512| 2      | 14×14×512    |
| Conv9     | 3×3×512| 1      | 14×14×512    |
| Conv10    | 3×3×512| 1      | 14×14×512    |
| MaxPool5  | 2×2×512| 2      | 7×7×512      |

Until now, there is no authoritative public test set for fire image recognition. In this paper, part of the flame video used in our experiment was downloaded from the Internet, and the other part used the laboratory to conduct fire simulation experiments and recorded the video. Experimental results show that the recognition rate of fires with obvious flame features such as diesel and wood flame is very high, while the recognition rate of transparent fires such as alcohol is not ideal, and the recognition rate of interference sources such as candles, electric lights, and flashlights is also high. In the future, we should improve the depth model and build a more effective learning model. On the other hand, more fire and interference source videos should be collected through experiments or other methods, and a more complete and scientific fire recognition training set and test set should be established.
Deep learning requires massive labeled data sets, otherwise it is difficult to achieve the expected results. It is very important to establish a large number of effective fire data set for fire recognition.

References
[1] Guyon I, Elisseeff A. An introduction to feature extraction. Feature extraction. Springer, Berlin, Heidelberg, 2006: 1-25.
[2] Gevers T, Smeulders A W M. Color-based object recognition. Pattern recognition, 1999, 32(3): 453-464.
[3] Haralick R M, Shanmugam K, Dinstein I H. Textural features for image classification. IEEE Transactions on systems, man, and cybernetics, 1973 (6): 610-621.
[4] Rautkorpi R, livarinen J. A novel shape feature for image classification and retrieval.International Conference Image Analysis and Recognition. Springer, Berlin, Heidelberg, 2004: 753-760.
[5] Girshick R, Donahue J, Darrell T, et al. Rich feature hierarchies for accurate object detection and semantic segmentation . Proceedings of the IEEE conference on computer vision and pattern recognition, 2014: 580-587.
[6] Frizzi S, Kaabi R, Bouchouicha M, et al. Convolutional neural network for video fire and smoke detection. IECON 2016-42nd Annual Conference of the IEEE Industrial Electronics Society. IEEE, 2016: 877-882.
[7] RafaelC. Gonzalez, et al. Digital image processing. Electronic Industry Press, 2007.
[8] Yang S B. Introduction of image edge detection technology. Journal of Wuhan Institute of Chemical Technology, 2003, 22(1): 73-76.
[9] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556, 2014.
[10] J Long, E Shelhamer, T Darrell. Fully Convolutional Networks for Semantic Segmentation.Proceedings of the IEEE conference on computer vision and pattern recognition, 2015:3431-3440.
[11] Islam M A, Kowal M, Jia S, et al. Position, padding and predictions: A deeper look at position information in cnns. arXiv preprint arXiv:2101.12322, 2021.
[12] Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems, 2012, 25: 1097-1105.