International Journal of Automation and Computing
www.ijac.net

18(2), April 2021, 300-310
DOI: 10.1007/s11633-020-1269-5

Fire Detection Method Based on Depthwise Separable Convolution and YOLOv3

Yue-Yan Qin[1] Jiang-Tao Cao[1] Xiao-Fei Ji[2]

[1]School of Information and Control Engineering, Liaoning Shihua University, Fushun 113001, China

[2]School of Automation, Shenyang Aerospace University, Shenyang 110136, China

Abstract: Recently, video-based fire detection technology has become an important research topic in the field of machine vision. This paper proposes a method of combining the classification model and target detection model in deep learning for fire detection. Firstly, the depthwise separable convolution is used to classify fire images, which saves a lot of detection time under the premise of ensuring detection accuracy. Secondly, You Only Look Once version 3 (YOLOv3) target regression function is used to output the fire position information for the images whose classification result is fire, which avoids the problem that the accuracy of detection cannot be guaranteed by using YOLOv3 for target classification and position regression. At the same time, the detection time of target regression for images without fire is greatly reduced saved. The experiments were tested using a network public database. The detection accuracy reached 98% and the detection rate reached 38 fps. This method not only saves the workload of manually extracting flame characteristics, reduces the calculation cost, and reduces the amount of parameters, but also improves the detection accuracy and detection rate.

Keywords: Fire detection, depthwise separable convolution, fire classification, You Only Look Once version 3 (YOLOv3), target regression.

Citation: Y. Y. Qin, J. T. Cao, 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. http://doi.org/10.1007/s11633-020-1269-5

1 Introduction

1.1 Introduction

Fire is a serious natural and social disaster. On the one hand, the occurrence of fire will cause great threat to people's lives and property safety[1], on the other hand, it will also cause huge loss to the natural and socioeconomic environment. According to the statistics of the World Fire Statistics Center, the number of fires worldwide each year is an astonishing number. In recent years, the incidence of fires has generally been on the rise, and the situation is very serious. Therefore, preventing and reducing fires as much as possible has always been a topic of active exploration.

For many years, researchers have continued to research and experiment on fire detection methods. The original fire detection method is based on sensors, and according to different sensor types and applications[2], it can generally be divided into five categories: light-sensitive, temperature-sensitive, smoke-sensitive, gas-sensitive and composite. Due to the characteristics of heat release and dense smoke during a fire, temperature and smoke sensors are commonly used. However, this detection method based on sensors has significant drawbacks in terms of the detection range and the detection speed[3]. Then, with the application and popularization of video surveillance technology, researchers have obtained fire images through video surveillance and used their color characteristics to detect fires. However, there is a greater false detection rate in fire detection using only color features[4].

In recent years, on the basis that the advancement of video surveillance technology can be seen in many public and private fields, the image processing technology in the field of machine vision has also made significant research progress. Through the video monitoring system, the color, shape change, texture structure, flicker and other related scene information of the fire image can be obtained intuitively[5], and the transmission and sensing speed have been improved. Therefore, fire detection technology based on computer vision came into being and promoted the diversity of fire detection methods. The fire detection method based on computer vision obtains fire images through video surveillance and manually extracts their features, and builds a detection model based on these features. Specific modeling methods can be divided into feature-level and decision-level model construction. The feature-level fusion fire detection method makes good use of the

Research Article
Manuscript received June 10, 2020; accepted November 16, 2020; published online February 2, 2021
Recommended by Associate Editor Hui Yu
Colored figures are available in the online version at https://link.springer.com/journal/11633

complementarity between different flame features, but it is not easy to achieve the fusion of heterogeneous features. Consequently, researchers have further studied the decision-level fusion of multiple flame features for this problem. The decision-level fusion fire detection method has a certain fault tolerance, but its preprocessing cost is relatively high.

At present, fire detection methods based on feature-level and decision-level modeling have made certain research progress. However, this detection method relies on manually extracting visible features of the flame. These features only reflect the shallow features of the flame and may cause information loss in the process of manual extraction. With the continuous development of research, the fire detection methods using manual extraction of traditional features have entered a bottleneck in terms of application scenarios, detection accuracy and detection speed. In recent years, with the success of convolutional neural networks (CNN) in static image classification and the breakthrough progress of deep learning theory in the field of machine vision, fire detection using its powerful feature representation ability and modeling ability has important research value and application prospects[6]. After deep learning has used convolutional neural networks to achieve image classification, researchers have introduced new target detection algorithms on this basis, such as two-step target detection based on region convolutional neural network networks (R-CNN) and end-to-end target detection algorithms based on You Only Look Once (YOLO) and single shot multibox detector (SSD) networks. These algorithms and models make up for the problem that traditional convolutional neural networks can only classify but cannot locate fire targets.

1.2 Literature review

The video and image of the flame have rich visible features, such as color features, texture features, flicker features, flame sharp angles, and shape changes. Among those flames, color feature is the earliest and most widely used. Shidik et al.[7] proposed the use of multiple color spaces as criteria for fire detection based on the uniqueness of the flame color. Han et al.[8] combined a variety of color feature rules to model fire detection methods after preprocessing the fire video. But, the fire scene is usually diverse, so using a single feature for fire detection will cause a high false detection rate.

So, researchers prefer to use multi-feature fusion for fire detection. The fusion methods can be divided into two types: feature-level fusion and decision-level fusion. The feature-level fire detection method is to comprehensively analyze and process multiple features of the flame. Zeng et al.[9] used the weighting method to fuse multiple features of flames, and their weighting coefficients were obtained by an analytic hierarchy process. Prema et al.[10] used support vector machines to identify the flicker feature and texture feature of the flame for fire recognition. Prema et al.[11] used extreme learning machines for fire detection. The decision-level fusion fire detection method is to classify or identify each flame feature to form corresponding results, and then fuse the results to give the final decision. Shi et al.[12] proposed to use two color space discrimination rules and flame motion characteristics to perform fire recognition, and each recognition result was processed in parallel to obtain the final detection result. Foggia et al.[13] proposed to use a multi-expert system for fire identification and detection, and then fuse the recognition results to obtain the final classification result. Li et al.[14] used color attributes, geometric attributes and motion attributes to perform fire detection respectively, and the detection results obtained were fused again to obtain the final decision.

Based on manually extracting features and modeling for fire detection, many scholars have also begun to use deep learning models for fire detection. Frizzi et al.[15] proposed a fire detection method for feature maps directly using the classic convolutional neural network AlexNet model. Muhammad et al.[16] used the method of transfer learning to fine tune the GoogleNet convolutional neural network for fire detection. Mahammad et al.[17] proposed a method of using the squeezeNet model of smaller convolutional kernels to identify the fire on the basis of a traditional convolutional neural network. Saeed et al.[18] proposed a fire detection method which is based on powerful machine learning and deep learning algorithms, their proposed model has three main deep neural networks, i.e., a hybrid model which consists of Adaboost and many multilayer perceptron (MLP) neural networks, the Adaboost local binary patterns (LBP) model and finally a convolutional neural network. Compared with traditional computer vision based fire detection methods, fire detection based on convolutional neural network models has made certain progress and has better stability. However, traditional convolutional neural network models can only classify fire images and cannot accurately locate the location of fire occurrences. Therefore, based on the use of convolutional neural networks for stable classification, deep learning based object detection algorithms have received more attention and applications, which have been extended to the research for fire recognition detection. Kim and Lee[19] used a faster R-CNN two-step target detection algorithm for fire detection. Liau et al.[20] proposed to use an efficient squeezeNet network to replace the back-end network of the SSDs network, and use residual connection and group convolution to expand the SSD framework based on the squeezeNet network for fire target detection. Shen et al.[21] used the algorithm-optimized YOLO model for fire detection. Du et al.[22] improved the candidate frame extraction and feature-level fusion algorithm of the end-to-end YOLOv2 model to detect fire targets. Ren et al.[23] used the improved YOLOv3 network model for fire classification and location regression. Although the target detection algorithm based on

deep learning makes up for the problem that typical convolutional neural networks can only classify but cannot locate the fire position, due to the complexity of the target detection model and the need to consider both classification and position regression tasks, the detection time may increase.

Due to the special nature of fire, its detection needs to weigh both detection time and accuracy. Therefore, a fire detection algorithm based on the combination of depthwise separable convolution and YOLOv3 is proposed under the premise of considering both detection speed and detection accuracy. Firstly, depthwise separable convolution is used to classify the fire image, which can greatly reduce the detection time without losing the detection accuracy. Then, the target regression function of YOLOv3 is used to output the position of the image whose classification result is fire. In the real scenario, the probability of fire is far lower than the probability of no fire, so only using its regression function saves a lot of time in detecting no fire. Compared with traditional video-based fire detection methods, the workload of manual feature extraction is reduced, and the detection accuracy is improved. Compared with the classic convolutional neural network, the method proposed in this paper can achieve the location of the fire target. In addition, the requirements on hardware are reduced, and the amount of calculation and parameters are greatly reduced. Compared with the typical deep learning target detection algorithm, the detection accuracy and detection rate of this algorithm can meet the requirements of fire detection.

The rest of the paper is organized as follows. Section 2 introduces fire detection model. Section 3 gives the network training method for the fire detection model. Section 4 discusses the testing results. Section 5 concludes the paper with some future work suggestions.

2 Fire detection model

The proposed method uses two stages for detecting fire from the input video. The first stage uses the classification model to classify the input image with or without fire. The classification model uses depthwise separable convolutional neural networks (DS-CNN). The second stage uses the target regression function of YOLOv3 to locate the fire position information for the image with fire and then output, and directly output for the image without fire. The various stages of the algorithm are shown in Fig. 1.

2.1 Classification model

In recent years, convolutional neural networks have made breakthrough progress in the field of image classification. The classic structure of traditional convolutional neural network models includes the first LeNet model for digital recognition, and models that won the ImageNet competition championship in 2012 and after, such as

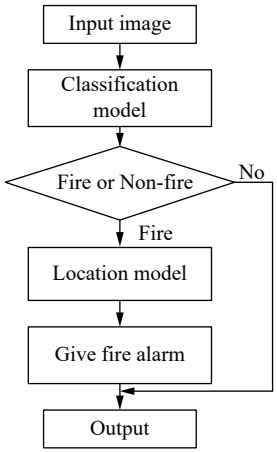


Fig. 1 Fire detection algorithm

AlexNet, VGGNet, GoogleNet, and ResNet models, etc.[24] But, traditional convolutional neural networks use large-scale convolution kernels, such as 11×11 convolution kernels in AlexNet. Traditional convolutional neural networks usually use large-scale convolution kernels, such as 11×11 convolution kernels in AlexNet. The larger the convolution kernel is, the larger the receptive field will be, but the number of parameters of the model will also increase. The model after AlexNet has improved this, for example, GoogleNet uses multiple 3×3 small-size convolution kernels to cascade while keeping the original image receptive field unchanged[25], which greatly reduces the amount of parameters. But, as the depth of the network increases and the convolution kernel needs to act on each channel of the input image, the amount of calculation is still large. Aiming at the problems that the traditional convolutional neural network has a large amount of calculation and many parameters, the depthwise separable convolution was proposed in 2013[26]. Depthwise separable convolution is an improvement and innovation based on standard convolution. The core is to decompose the standard three-dimensional convolution into two-dimensional and one-dimensional convolution.

1) Depthwise separable convolution

The basic idea of depthwise separable convolution is to decompose the standard convolution into depth-wise convolution and point-wise convolution.

Step 1. Depthwise convolution is to carry out 2D convolution for each channel of the input image to reduce the amount of parameters.

Step 2. Pointwise convolution is based on depthwise convolution, using a 1×1 convolution kernel to convolute all channels, greatly reducing the amount of calculation. The difference between standard convolution and depthwise separable convolution in the convolution process is shown in Fig. 2. Fig. 2(a) is the standard convolution process. Figs. 2(b) and 2(c) correspond to depthwise convolution and pointwise convolution of depthwise separable convolution.

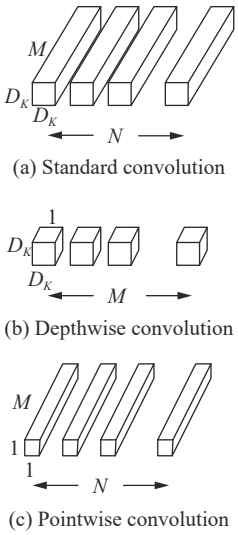


Fig. 2 Standard and depthwise separable convolution process

Assuming that the input feature map size is $D_f \times D_f \times M$, the output feature map size is $D_f \times D_f \times N$, and the convolution kernel size is $D_k \times D_k$. The following is a calculation and comparison of the parameters and calculations involved in the standard convolution and the depthwise separable convolution process.

2) Parameter amount

The standard convolution parameter amount is

$$Par_S = D_k \times D_k \times M \times N. \tag{1}$$

The parameter amount of the depthwise separable convolution is

$$Par_{D-P} = D_k \times D_k \times M + M \times N. \tag{2}$$

3) Calculation amount

The calculation amount of the standard convolution Cal_S is as follows:

$$Cal_S = D_f \times D_f \times M \times N \times D_k \times D_k. \tag{3}$$

The calculation amount of depthwise convolution in depthwise separable convolution is shown in (4), and the calculation amount of point-wise convolution is shown in (5). The total calculation amount is shown in (6).

$$Cal_D = D_f \times D_f \times M \times D_k \times D_k \tag{4}$$

$$Cal_P = D_f \times D_f \times M \times N \tag{5}$$

$$Cal_T = D_f \times D_f \times M \times D_k \times D_k + D_f \times D_f \times M \times N. \tag{6}$$

The ratio of the calculation amount of the depthwise separable convolution to the standard convolution is

$$\frac{Cal_S}{Cal_T} = \frac{D_f \times D_f \times M \times N \times D_k \times D_k}{D_f \times D_f \times M \times D_k \times D_k + D_f \times D_f \times M \times N} = \frac{1}{N} + \frac{1}{D_k \times D_k}. \tag{7}$$

According to the calculation, the reduction of the calculation amount of the depth separable convolution is related to the size of the convolution kernel $D_k \times D_k$ and the number of output channels N. The neural network models used in practical applications usually have multiple convolutional layers, and the convolution kernels usually use 3×3 and above convolution kernels. Thus, depthwise separable convolution can greatly reduce the number of parameters and calculations without losing accuracy, which also makes it possible to effectively apply deeper and wider neural network architectures. Even in resource-constrained micro controllers, it can run normally. Consequently, in this paper, the depthwise separable convolutional neural network is chosen as the fire classification model.

4) Depthwise separable convolutional neural networks

Based on depthwise separable convolutions, this paper proposes to use end-to-end depth separable convolutional neural networks to classify images with and without fire. The specific network structure is shown in Fig. 3. The specific network structure is shown in Fig. 3. It mainly consists of 4 convolutional layers, 3 pooling layers, and the pooling method is Max pooling, 2 fully connected layers, and 1 softmax regression layer. The first 9 layers of the network are used for feature extraction, and the last layer is used for classification. In addition to the above main structure, it also includes the activation function layer between the convolutional layer and the pooling layer. The activation function uses a rectified linear unit (Relu function) and a batch normalization (BN). And the dropout layer is added between the full connec-

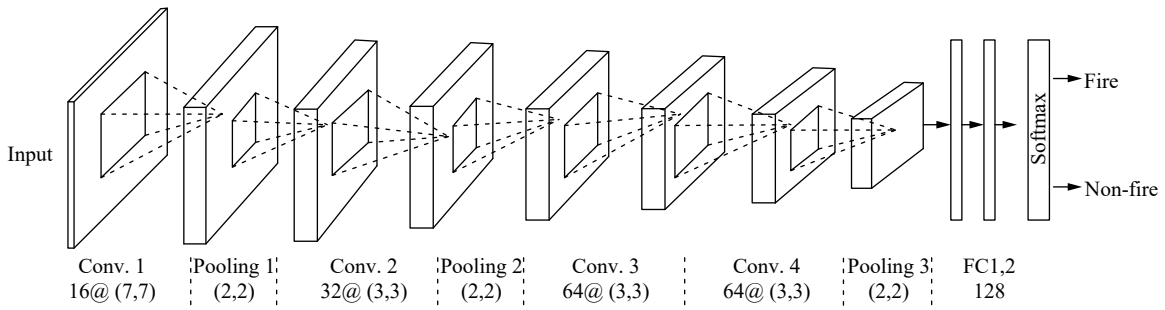


Fig. 3 DS-CNN fire classification structure

tion layers, in which the probability is 0.5.

The structure of the fire classification network model:

Input: The network input uses red-green-blue (RGB) image data, resizes the original image as 128×128×3, and standardizes each channel. The output result is used as the input of the first convolutional layer.

Standard convolution module: A 7×7 conventional convolution is used between the input layer and the output layer of the network, followed by the BN layer and the RELU layer (Fig. 4(a)).

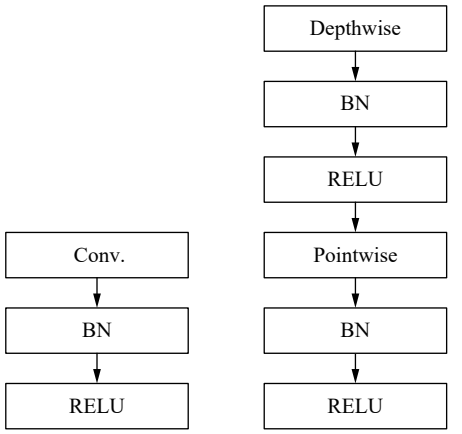
Depthwise separable convolution module: It consists of 7×7 depthwise direction convolution followed by BN layer and RELU layer and 1×1 pointwise convolution layer followed by BN layer and RELU layer. The point convolution step is 1 (Fig. 4(b)).

Pooling module: Using pooling layer parameters (2, 2), the feature map output by depthwise separable convolution module is down sampling to half, so as to reduce the dimension of fire characteristics.

Output: After passing through the depthwise separable module and pooling module, the convolution in 3 × 3 depthwise direction convolution followed by BN layer and RELU layer and 1 × 1 pointwise convolution followed by BN layer and RELU layer, continued for three times, and the pointwise convolution step is 1. And three down sampling with pooling parameters is (2, 2). After the depthwise separation and pooling operations, the two-dimensional fire feature maps are transformed into a one-dimensional vector by using two full connection layers. The number of neural unit nodes in the full connection layer is 128. And a softmax activation function output is connected to obtain the classification of fire and non-fire at the same time.

2.2 Location model

Based on the classification results of fire data, the target regression of YOLOv3 is used to further locate the images that are classified as a fire. YOLOv3 uses the network structure of Darknet-53. Darknet-53 introduces a residual block in the network. The gradient problem of the deep network is solved, so that the training difficulty of the network is reduced. There is no pooling layer and fully connected layer in the entire network. The downsampling of the network is achieved by setting the convolution step to 2[27]. In addition, YOLOv3 can realize multi-scale detection, and the specific form of multi-scale detection is the operation of up sampling and splicing in the last certain layers of network prediction. The small scale feature maps can provide richer and deeper levels of semantic information, and the large size feature maps can provide target location information more accurately. Combining small-scale feature maps with meso-scale feature maps and large-scale feature maps can both detect large targets and effectively detect small targets[28]. YOLOv3 further uses three different scale feature maps to detect objects, which can detect more fine-grained features. The final output of the network has three scales: 1/32, 1/16 and 1/8, respectively, the 1/32 prediction results have a high sampling ratio and large receptive field of feature map, so it is suitable for detecting objects with a large scale in the image. The 1/16 prediction results have a medium scale receptive field, which is suitable for detection of medium-scale objects. The 1/8 of prediction results have the smallest receptive field, which was suitable for detecting small scale objects. The specific network structure is shown in Fig. 5. During the fire, the fire will change continuously, sometimes it is a small fire, sometimes it may be a large fire. Therefore, YOLOv3 is chosen as the fire location model in this paper.



(a) Standard convolution layer (b) Depthwise separable convolution layer

Fig. 4 Convolution layer structure change

3 Network training

3.1 Classification model training

During the network training process, some parameters and algorithms need to be set and selected in advance. This includes parameters such as the learning rate, the number of iterations, the batch training, and the selection of data augmentation, loss functions, and optimizer.

1) Initialization parameters

The initialization parameters mainly include settings for learning rate, number of iterations, and batch training. For learning rate, run the learning rate finder method through a certain number of iterations before formal training, and generate the result into a learning rate finder plot. The learning rate finder plot in this paper is shown in Fig. 6.

It can be seen from Fig. 6 that the network starts to gain traction between 10^{-5} and 10^{-4} and starts to learn. The lowest loss can be found between 10^{-2} and 10^{-1}. However, at 10^{-1}, the loss starts to increase sharply,

	Type	Filters	Size	Output
	Convolutional	32	3×3	256×256
	Convolutional	64	3×3/2	128×128
1×	Convolutional	32	1×1	
	Convolutional	64	3×3	
	Residual			128×128
	Convolutional	128	3×3/2	64×64
2×	Convolutional	64	1×1	
	Convolutional	128	3×3	
	Residual			64×64
	Convolutional	256	3×3/2	32×32
8×	Convolutional	128	1×1	
	Convolutional	256	3×3	
	Residual			32×32
	Convolutional	512	3×3/2	16×16
8×	Convolutional	256	1×1	
	Convolutional	512	3×3	
	Residual			16×16
	Convolutional	1 024	3×3/2	8×8
4×	Convolutional	512	1×1	
	Convolutional	1 024	3×3	
	Residual			8×8

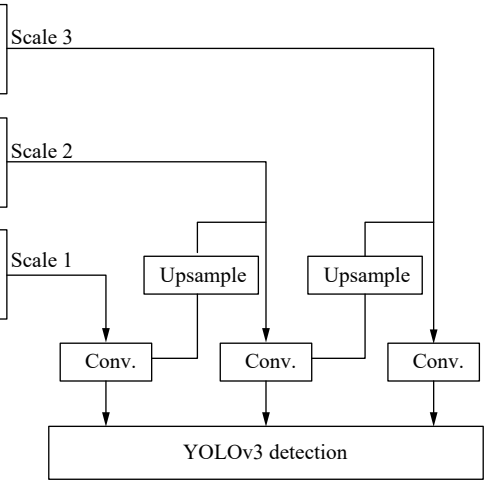


Fig. 5 YOLOv3 network structure

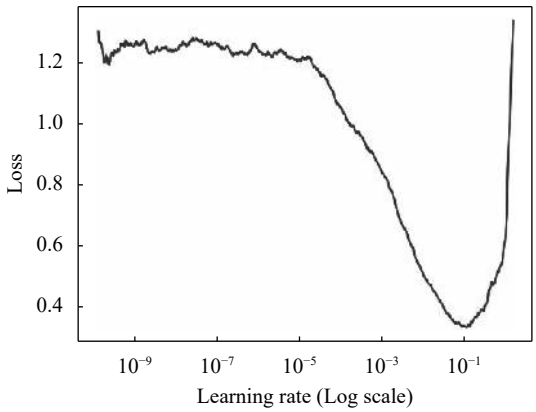


Fig. 6 Learning rate finder plot

which indicates that the learning rate is too large and the network is overfitting. So, in order to prevent the network from overfitting and guarantee the generalization ability, this paper uses an initial learning rate of 10^{-2}. For batch training, if it is too small, the training data will be difficult to converge. If it is too large, the relative processing speed will increase, but the required memory capacity will also increase. Therefore, this paper chooses a batch training of 64 and performs 50 iterative trainings on the entire training set.

2) Date augmentation

In deep learning, the number of samples is generally large enough. When the number of samples is sufficient, the effect of the trained model is better and the generalization ability is stronger. However, in practical applications, the number of samples is often insufficient due to various factors, which requires data augmentation for existing samples to increase the number of samples. Common methods for data augmentation include data flipping, rotation, image scaling, cropping, translation, adding noise, etc.[29] Data augmentation will expand the amount of data. But, if the test samples do not have such randomness, it will not work and will increase the training time. Therefore, according to the characteristics of fire changes in the process of fire, we use rotation, scaling, horizontal translation, vertical translation, cropping and horizontal rotation to enhance the data. The rotation angle is controlled within 30°. The scaled size is within 0.15. The horizontal and vertical translations are controlled within 0.2. And the cropping transformation is controlled within 0.15. During the data augmentation process, the original data will not be modified. Instead, more similar and diverse data are obtained through image processing and other methods, and does not take up more memory space. All processing processes are processed on-the-fly in memory.

3) Loss function

The loss function is also called the cost function. It is a function of measuring the difference between the predicted value and the actual value of the output of the neural network. The loss function is often associated with optimization problems as a learning criterion. The commonly used loss functions are mean square error (MSE)

loss function, binary cross entropy loss function, categorical cross entropy function. The MSE loss function is the most classic and simplest, but the accuracy is relatively poor. The binary cross entropy loss function is generally used for binary classification problems. The categorical cross entropy loss function is usually used in multiple classification cases. Since the fire classification is a binary classification problem, a binary cross entropy loss function is selected in this paper. The loss function expression is as follows:

$$Loss = -\sum_{i}^{n} \hat{y} \log y_i + (1-\hat{y}) \log (1-y_i) \tag{8}$$

where n is the number of samples, $\hat{y}$ is the predicted value, and y_i is the actual value. Differentiate the function with respect to y, the result is shown in the following formula (9):

$$\frac{\partial Loss}{\partial y} = \sum_{i=1}^{n} \frac{\hat{y}_i}{y_i} - \frac{1-\hat{y}_i}{1-y_i}. \tag{9}$$

When $y_i = \hat{y}_i$, the $Loss$ is equal to 0. In addition, $Loss$ is a positive number, and the greater the difference between the predicted value and the actual value, the greater the value of $Loss$.

4) Optimizer

The role of the optimizer is to update and calculate network parameters that affect model training and model output, such as learning rate. This makes it approximate or reach the optimal value, thereby minimizing the loss function. The most basic algorithm of the optimizer is the gradient descent method. At present, the three main types of gradient descent methods are batch gradient descent (BGD), stochastic gradient descent (SGD), and mini-batch gradient descent (MBGD). The BGD calculates the gradient for the entire data set in one update, which will cause a large amount of calculation and the calculation speed is very slow. For similar samples, BGD will be redundant when calculating the gradient. When the amount of data is large, the calculation amount of the algorithm becomes very difficult, and new data cannot be invested to update the model in real time. MBGD calculates a small batch of samples at a time, and the convergence is stable. It can make full use of the highly optimized matrix operations in the deep learning library to perform more efficient gradient calculations. But it also has shortcomings. On the one hand, the convergence rate is very slow when the learning rate is too small. On the other hand, the loss function will continue to oscillate at the minimum value when the learning rate is too large. SGD only selects one sample for calculation, which has no redundancy and is relatively fast. It can also add new samples. So, the SGD optimizer is often applied at present. However, because the SGD algorithm is updated frequently, the loss function will have serious oscillations. Therefore, the momentum SGD is used in this paper, where the momentum parameter is set to 0.9. The role of adding momentum parameters in SGD is mainly to accelerate convergence, improve accuracy, and reduce oscillations during convergence. The parameter update expression is as follows:

$$\theta_i = \theta_i - \eta \left(h_\theta(x_0^{(j)}, x_1^{(j)}, \cdots, x_n^{(j)}) - y_j \right) x_i^j \tag{10}$$

where θ is the model parameter, η is the learning rate, j is the sample, $h(x_i)$ is the randomly selected gradient direction, and y_j is the loss function.

3.2 Location model training

In this paper, a depthwise separable convolutional neural network has been used to classify the input fire data. Next, only the data with fire information need to be located. In other words, it is not necessary to use YOLOv3 to perform classification prediction, but to use its positioning function to output the fire location when the classification is known. Therefore, we use the YOLOv3 model to locate the fire through transfer learning. The specific training steps are as follows:

Step 1. Use labeling software to frame the fire sample data and process it into the data format required by the YOLOv3 model to generate a training set of fire images.

Step 2. Modify and adjust the classification prediction function and configuration file parameters in the YOLOv3 model accordingly.

Step 3. Use transfer learning to retrain the YOLOv3 model using our own labeled database.

4 Algorithm test

4.1 Experimental environment

The software environment of the experiment in this paper is the ubuntu 16.04 LTS operating system. We compile the program under the TensorFlow2.0 framework and use python 3.6.6 as the programming language. The microprocessor of the hardware platform is Intel (R) Core (TM) i7-4 790 with 3.6 GHz main frequency and 15.6 GiB memory.

4.2 Experimental data

The experiment uses two types of fire sample data for training and testing, respectively. One of the data sets is composed of fire images on Google and Baidu. They are all images taken at a certain time when the fire occurred. There is no time series relationship and no gradual process, as shown in Fig. 7(a). Another data set is composed

of public fire video set. By dividing the video into frames and selecting them at equal intervals, its purpose is to establish a potential time series relationship between data sets, covering the fire data from small to large fire processes, as shown in Fig. 7(b). Both data sets include different scenarios such as indoor, outdoor, forest, road, and day and night. Negative samples are a natural complement to fire scenes. It is composed of scenes with similar characteristics to fire occurrence and disturbances similar to fire. In order to be able to compare whether different data sets can change the recognition accuracy, the two fire data sets are 1 719 frames, and the negative sample set is 2 689 frames, of which 75% are used for training and 25% are used for testing. During the training process, the model will also apply data augmentation functions to use its rotation, translation, scaling and other operations to enrich the training samples. The loss and accuracy of training and testing are shown in the following Fig. 8.

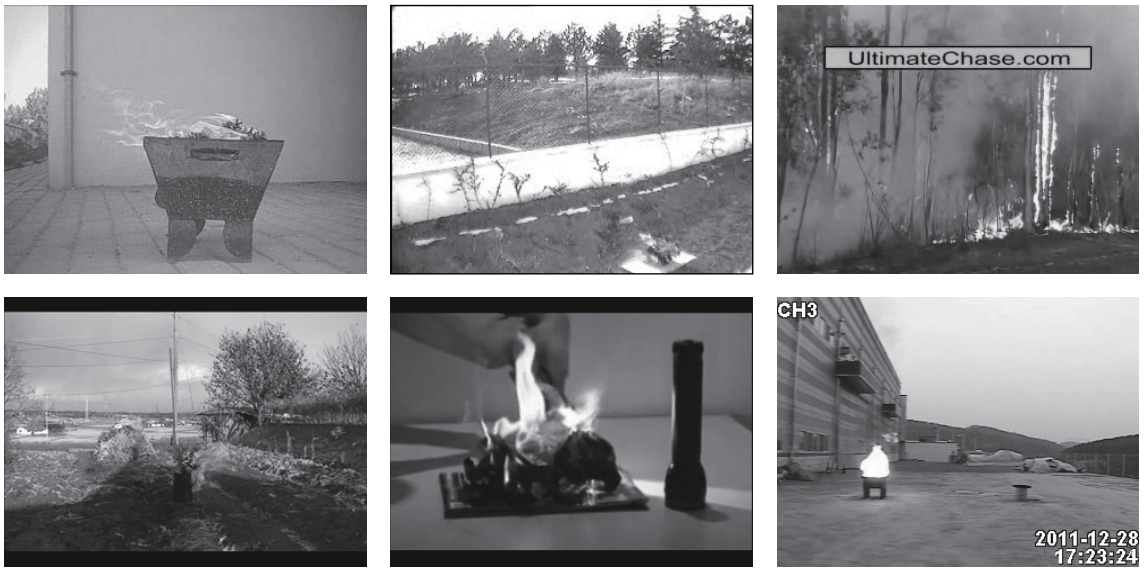
4.3 Experimental results and analysis

The detection results obtained by using the classification model and the location model based on the classification model are shown in Figs. 9(a) and 9(b), respectively.

In order to prove the effectiveness of the algorithm, the experiment makes a comparative analysis from the following aspects:

1) In the same experimental environment, the test results of two different fire data sets using depthwise separable convolutional neural networks are compared and analyzed, as shown in Table 1.

(a) No time series relationship data



(b) Time series relationship data

Fig. 7 Part of the fire data list

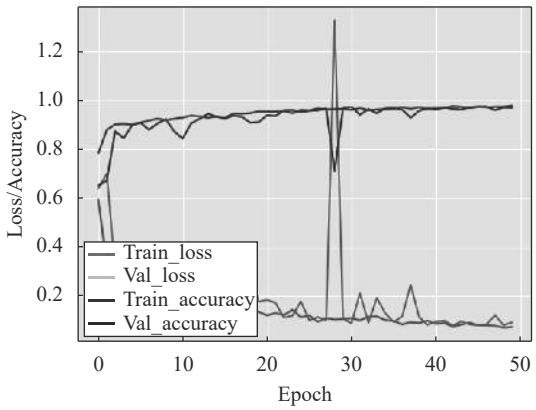


Fig. 8 Training loss and accuracy

In Table 1, by comparing the detection results of the two fire data sets, it can be seen that the fire data with a time series relationship has higher detection accuracy and lower false detection rate than the fire data without a time series relationship.

2) Under the same experimental environment and network structure, the fire data set with a time series is used to compare the detection accuracy and detection rate of the standard convolutional neural network (CNN) and deep separable convolutional neural network (DS-CNN), as shown in Table 2.

In Table 2, there is no significant difference in the detection accuracy of the two network structures, but, great changes have occurred in the detection rate. Therefore, it is proved that the depthwise separable convolutional neural network can greatly improve the detection rate while ensuring the detection accuracy.

In Table 2, there is no significant difference in the detection accuracy of the two network structures, but great changes have occurred in the detection rate. Therefore, it is proved that the depthwise separable convolutional neural network can greatly improve the detection rate while ensuring the detection accuracy.

3) Compare the fire detection algorithms of this paper and related literatures, as shown in Table 3.

In Table 3, through comparative analysis of different algorithms, we can see that the algorithm proposed in this paper has achieved good results in accuracy and detection rate, among which the detection rate and accuracy are higher than those in [14] and [18], but slightly lower than that in [21]. Li et al.[14] classified the fire image through the classic structure of the convolutional neural network AlexNet, but did not locate the fire position. Because of the larger convolution kernel of the AlexNet model and the greater number of layers than the algorithm in this paper, its detection accuracy and detection rate are significantly lower than the algorithm in this paper. Saeed et al.[18] used a two-step faster RCNN target detection structure to classify and locate fire images. The faster RCNN model uses the region proposal network (RPN) instead of the selective search method to generate a candidate target box, which improves the algorithm′s detection accuracy and detection rate. But, it is still difficult to meet the requirements of real-time detection. Shen et al.[21] improved the detection accuracy and detection rate of the fire by improving the clustering method of YOLOv2′s network structure and the fusion of shallow and deep features. But, the algorithm in this paper does not change the network structure. By combining the classification model with the location model to achieve the classification and location function of the fire, it not only improves the detection accuracy, but also increases the detection rate.

5 Conclusions

Fire prevention and real-time detection are of great significance for protecting people′s property, forest vegetation, chemical equipment, etc. Many experts and scholars continue to improve and innovate the fire detection algorithm to meet the detection requirements of the real

(a) Classification (b) Fire location

Fig. 9 Test result

Table 1 Comparison of the detection results of different fire data sets

Data sets	Accuracy	False detection rate
No time series relationship	94.0%	6%
Time series relationship	98.0%	2%

Table 2 Comparison of the detection results of CNN and DS-CNN

Network structure	Accuracy	Detection rate
CNN	97.8%	20 fps
DS-CNN	98.0%	50 fps

Table 3 Comparison of the detection algorithms of related literature and our algorithm

Related literature	Accuracy	Detection rate
This paper	98%	38 fps
Literature [14]	90%	20 fps
Literature [18]	96.93%	–
Literature [21]	98.8%	40 fps

environment. In recent years, research on fire detection methods has gradually expanded from the traditional feature extraction algorithm to the field of deep learning, and has achieved certain results in this process. Therefore, this paper uses a 10-layer depthwise separable convolutional neural network as a classification model for fire images, which greatly reduces the amount of calculation and parameters. Then, on the basis of the classification, it only uses the target regression function of YOLOv3 to locate the fire location. The method proposed in this paper can not only ensure the accuracy of the algorithm, but also meet the real-time requirements of fire detection. The detection accuracy and rate are 98% and 38 fps, respectively. And it has good applicability to the detection of different scenes. Further work will focus on further simplifying the model structure and embodying the timing information of the video sequence in the network structure.

Acknowledgements

This work was supported by Liaoning Provincial Science Public Welfare Research Fund Project (No. 2016 002006), and Liaoning Provincial Department of Education Scientific Research Service Local Project (No. L201708).

References

[1] F. Saeed, A. Paul, W. H. Hong, H. Seo. Machine learning based approach for multimedia surveillance during fire emergencies. *Multimedia Tools and Applications*, vol. 79, no. 23, pp. 16201–16217, 2020. DOI: 10.1007/s11042-019-7548-x.

[2] F. Saeed, A. Paul, A. Rehman, W. H. Hong, H. Seo. IoT-based intelligent modeling of smart home environment for fire prevention and safety. *Journal of Sensor and Actuator Networks*, vol. 7, no. 1, Article number 11, 2018. DOI: 10.3390/jsan7010011.

[3] M. J. Park, B. C. Ko. Two-step real-time night-time fire detection in an urban environment using static ELASTIC-YOLOv3 and temporal fire-tube. *Sensors*, vol. 20, no. 8, Article number 2202, 2020. DOI: 10.3390/s20082202.

[4] J. H. Li, R. X. Fan, Z. B. Chen. Forest fire recognition based on color and texture features. *Journal of South China University of Technology* (*Natural Science Edition*), vol. 48, no. 1, pp. 70–83, 2020. DOI: 10.12141/j.issn.1000-565X.190181. (in Chinese)

[5] N. M. Dung, B. Choi, S. Ro. A study on the fire detection algorithm using surveillance camera systems. *The Journal of Korean Institute of Communications and Information Sciences*, vol. 43, no. 6, pp. 921–929, 2018. DOI: 10.7840/kics.2018.43.6.921.

[6] V. K. Ha, J. C. Ren, X. Y. Xu, S. Zhao, G. Xie, V. Masero, A. Hussain. Deep learning based single image super-resolution: A survey. *International Journal of Automation and Computing*, vol. 16, no. 4, pp. 413–426, 2019. DOI: 10.1007/s11633-019-1183-x.

[7] G. F. Shidik, F. N. Adnan, C. Supriyanto, R. A. Pramunendar, P. N. Andono. Multi color feature, background subtraction and time frame selection for fire detection. In *Proceedings of International Conference on Robotics, Biomimetics, Intelligent Computational Systems*, IEEE, Jogjakarta, Indonesia, pp. 115–120, 2013. DOI: 10.1109/ROBIONETICS.2013.6743589.

[8] X. F. Han, J. S. Jin, M. J. Wang, W. Jiang, L. Gao, L. P. Xiao. Video fire detection based on Gaussian mixture model and multi-color features. *Signal, Image and Video Processing*, vol. 11, no. 8, pp. 1419–1425, 2017. DOI: 10.1007/s11760-017-1102-y.

[9] S. T. Zeng, H. B. Wu, P. H. Shen. Video fire detection based on fusion of multiple features. *Journal of Graphics*, vol. 38, no. 4, pp. 549–557, 2017. DOI: 10.11996/JG.j.2095-302X.2017040549. (in Chinese)

[10] C. E. Prema, S. S. Vinsley, S. Suresh. Multi feature analysis of smoke in YUV color space for early forest fire detection. *Fire Technology*, vol. 52, no. 5, pp. 1319–1342, 2016. DOI: 10.1007/s10694-016-0580-8.

[11] C. E. Prema, S. S. Vinsley, S. Suresh. Efficient flame detection based on static and dynamic texture analysis in forest fire detection. *Fire Technology*, vol. 54, no. 1, pp. 255–288, 2018. DOI: 10.1007/s10694-017-0683-x.

[12] L. Shi, F. F. Shi, T. Wang, L. P. Bu, X. G. Hou. A new fire detection method based on the centroid variety of consecutive frames. In *Proceedings of the 2nd International Conference on Image, Vision and Computing*, IEEE, Chengdu, China, pp. 437–442, 2017. DOI: 10.1109/ICIVC.2017.7984594.

[13] P. Foggia, A. Saggese, M. Vento. Real-time fire detection for video-surveillance applications using a combination of experts based on color, shape, and motion. *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 9, pp. 1545–1556, 2015. DOI: 10.1109/TCSVT.2015.2392531.

[14] S. Q. Li, W. Liu, H. D. Ma, H. Y. Fu. Multi-attribute based fire detection in diverse surveillance videos. In *Proceedings of the 23rd International Conference on Multimedia Modeling*, Springer, Reykjavik, Iceland, pp. 238–250, 2017. DOI: 10.1007/978-3-319-51811-4_20.

[15] S. Frizzi, R. Kaabi, M. Bouchouicha, J. M. Ginoux, E. Moreau, F. Fnaiech. Convolutional neural network for video fire and smoke detection. In *Proceedings of the 42nd Annual Conference of the IEEE Industrial Electronics Society*, Florence, Italy, pp. 877–882, 2016. DOI: 10.1109/IECON.2016.7793196.

[16] K. Muhammad, J. Ahmad, I. Mehmood, S. Rho, S. W. Baik. Convolutional neural networks based fire detection in surveillance videos. *IEEE Access*, vol. 6, pp. 18174–18183, 2018. DOI: 10.1109/ACCESS.2018.2812835.

[17] K. Muhammad, J. Ahmad, Z. H. Lv, P. Bellavista, P. Yang, S. W. Baik. Efficient deep CNN-based fire detection and localization in video surveillance applications. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, no. 7, pp. 1419–1434, 2019. DOI: 10.1109/

TSMC.2018.2830099.

[18] F. Saeed, A. Paul, P. Karthigaikumar, A. Nayyar. Convolutional neural network based early fire detection. *Multimedia Tools and Applications*, vol. 79, no. 13, pp. 9083–9099, 2020.

[19] B. Kim, J. Lee. A video-based fire detection using deep learning models. *Applied Science*, vol. 9, no. 14, Article number 2862, 2019. DOI: 10.3390/app9142862.

[20] H. Liau, N. Yamini, Y. L. Wong. Fire SSD: Wide fire modules based single shot detector on edge device. [Online], Available: https://arxiv.org/abs/1806.05363, 2018.

[21] D. Q. Shen, X. Chen, M. Nguyen, W. Q. Yan. Flame detection using deep learning. In *Proceedings of the 4th International Conference on Control, Automation and Robotics*, IEEE, Auckland, New Zealand, pp. 416–420, 2018. DOI: 10.1109/ICCAR.2018.8384711.

[22] C. X. Du, Y. Y. Yan, Y. A. Liu, S. B. Gao. Video fire detection method based on YOLOv2. *Computer Science*, vol. 46, no. 6, pp. 301–304, 2019. DOI: 10.11896/j.issn.1002-137x.2019.06.045. (in Chinese)

[23] J. F. Ren, W. H. Xiong, Z. H. Wu, M. Jiang. Fire detection and identification based on improved YOLOv3. *Computer Systems and Applications*, vol. 28, no. 12, pp. 171–176, 2019. (in Chinese)

[24] L. Ma, M. C. Li, X. X. Ma, L. Cheng, P. J. Du, Y. X. Liu. A review of supervised object-based land-cover image classification. *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 130, pp. 277–293, 2017. DOI: 10.1016/j.isprsjprs.2017.06.001.

[25] H. Wu, Z. W. Chen, G. H. Tian, Q. Ma, M. L. Jiao. Item ownership relationship semantic learning strategy for personalized service robot. *International Journal of Automation and Computing*, vol. 17, no. 3, pp. 390–402, 2020. DOI: 10.1007/s11633-019-1206-7.

[26] Q. F. Liu, H. L. Zhang, Y. L. Wang. Real-time pixel-wise classification of agricultural images based on depthwise separable convolution. *Scientia Agricultura Sinica*, vol. 51, no. 19, pp. 3673–3682, 2018. DOI: 10.3864/j.issn.0578-1752.2018.19.005. (in Chinese)

[27] B. Liu, S. Z. Wang, J. S. Zhao, M. F. Li. Ship tracking and recognition based on Darknet network and YOLOv3 algorithm. *Journal of Computer Applications*, vol. 39, no. 6, pp. 1663–1668, 2019. DOI: 10.11772/j.issn.1001-9081.2018102190. (in Chinese)

[28] M. R. Ju, H. B. Luo, Z. B. Wang, M. He, Z. Chang, B. Hui. Improved YOLOv3 algorithm and its application in small target detection. *Acta Optica Sinica*, vol. 39, no. 7, Article number 0715004, 2019. DOI: 10.3788/AOS201939.0715004. (in Chinese)

[29] W. J. Chai, L. M. Wang. Recognition of Chinese characters using deep convolutional neural network. *Journal of Image and Graphics*, vol. 23, no. 3, pp. 410–417, 2018. DOI: 10.11834/jig.170399. (in Chinese)

Yue-Yan Qin is a master student in control theory and control engineering at Liaoning Shihua University, China.

Her research interests include image processing and intelligent video analysis.

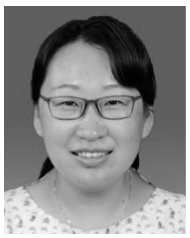
E-mail: 18341318515@163.com

ORCID iD: 0000-0002-6225-3519

Jiang-Tao Cao received the Ph. D. degree in intelligent control from University of Portsmouth, China in 2009. Now, he is a professor and M. Sc. supervisor at Liaoning Shihua University, China.

His research interests include intelligent method and its application, and video analysis.

E-mail: cigroup@126.com



Xiao-Fei Ji received the M. Sc. in control theory and control engineering from Liaoning Shihua University, China in 2003, and the Ph. D. degree in pattern recognition and intelligent systems from University of Portsmouth, UK in 2010. From 2003 to 2012, she was a lecturer with School of Automation, Shenyang Aerospace University, China. Since 2013, she has been an associate professor with Shenyang Aerospace University, China. She has published over 40 technical research papers and 3 books. She is the leader of National Natural Science Foundation Project (61103123) and six national and local government projects.

Her research interests include vision analysis and pattern recognition, information processing and fusion.

E-mail: jixiaofei7804@126.com (Corresponding author)

ORCID iD: 0000-0001-8279-7727

International Journal of Automation & Computing

Citation: Y. Y. Qin, J. T. Cao, 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. https://doi.org/10.1007/s11633–020–1269–5