Computer Science and Information Systems 20(4):1819–1843 https://doi.org/10.2298/CSIS221218032H

A Novel Single Shot-multibox Detector Based on Multiple Gaussian Mixture Model for Urban Fire Smoke Detection

Hao Han

Guizhou Fire and Rescue Brigade
Department of Natural Resources of Guizhou Province
hanhaovip0@163.com

Abstract. Under complex scenes, the traditional smoke detection methods cannot satisfy the real-time and accuracy requirements. Therefore, this paper proposes a novel single shot-multibox detector based on a multiple Gaussian mixture model for urban fire smoke detection. Multiple Gaussian models are used to represent the features of each pixel in the moving object image. The Gaussian mixture model is updated based on the principle that each pixel in the image is regarded as a background point if it matches the Gaussian mixture model. Otherwise, if it matches the Gaussian mixture model, it is regarded as the foreground point. By updating the foreground model and calculating the short-term stability index, the detection effect of moving objects is improved. By determining the relationship between Gaussian distribution and pixel, a new parameter is set to construct the background model to eliminate the influence caused by illumination mutation. Aiming at the problems of smoke detection efficiency and network over-fitting, we present an InceptionV3-feature fusion single shot-multibox detector. The new neural network is trained and tested by smoke positive and negative sample images. At the same time, Multibox Loss function is replaced by the Focal Loss function, which reduces the detector misdetection caused by the imbalance of positive and negative samples. Experimental results show that the proposed method is feasible and effective. The average accuracy of smoke detection is 97.5%, and the average response time of the smoke alarm is 4.57s, which can meet the requirements of real-time smoke detection in complex scenes.

Keywords: urban fire smoke detection, multiple Gaussian mixture model, single shot-multibox detector, InceptionV3-feature fusion.

1. Introduction

Smoke is an important symbol of the early stage of fire. Accurate detection and recognition of smoke are helpful for early warning of fire. Traditional smoke detection methods mainly use smoke detectors, the detection range is small, and the detection accuracy is susceptible to temperature, humidity, airflow, and other factors [1,2]. In recent years, smoke detection technology has been widely used in fire warning, fire detection and other fields because of its advantages of wide monitoring range, sensitive response and low environmental requirements [3].

In recent years, the research on smoke detection mainly focuses on the static characteristics of smoke, such as color, shape and texture, and the dynamic characteristics of smoke, such as motion and diffusion. Generally, the flow of smoke detection algorithm

can be divided into three stages: extraction of smoke proposal area, extraction of smoke features, and smoke recognition. Millan-garcia et al. [4] used the color space features of smoke to process video images and exclude non-smoke areas. The shortcoming of this method was that the color information was sensitive to threshold setting. Favorskaya et al. [5] regarded the smoke as dynamic texture, and used different local binary mode (LBP) histograms to classify and identify dense smoke, transparent smoke and non-smoke areas. For remote fire smoke images, Zhou et al. [6] realized smoke recognition based on the local extremal region segmentation method, but this method had a high false positive rate for the areas with thick fog. Dimitropoulos et al. [7] proposed a high-order linear dynamic system (H-LDS) to describe the feature operator and analyzed the dynamic texture of smoke, thus improving the recognition rate of smoke features. Jia et al. [8] first performed enhanced color transformation on the smoke image, then segmented the smoke proposal area, and finally detected the smoke area by establishing static and dynamic criteria of smoke. Vijayalakshmi et al. [9] established a saliency smoke detection model based on the color and motion characteristics of smoke and realized the segmentation of smoke area. Starting from the tone of the image, Cruz et al. [10] compared the pixel tonal distribution of the area containing smoke and the area without smoke, and proposed the concept of fire detection index, which was used to extract the smoke area. Ye et al. [11] used the motion characteristics of smoke to extract the smoke proposal movement area in the video, and further realized the identification of smoke based on space-time wavelet transform, Weber contrast analysis and color space segmentation.

After the above video smoke detection methods extract the features of the smoke proposal area, it is generally necessary to set certain thresholds for the relevant features, and then form a rule criterion to identify the smoke. In this process, most of the smoke feature extraction operators are manually designed, which may not reflect the essential characteristics of the smoke. The selection of threshold values mostly depends on personal experience, and the rationality of threshold value greatly affects the effect of smoke identification [12]. Therefore, based on the above traditional video smoke detection, some scholars have studied smoke recognition based on support vector machine (SVM), AdaBoost algorithm and other methods. Kim et al. [13] used SVM to identify the smoke in the video based on the optical flow characteristics of the smoke by analyzing its diffusivity, color and opacity. Prema et al. [14] extracted the suspected smoke area by YUV color model (Y is the brightness value of black and white, U and V are the chroma value), and realized the smoke identification based on the extracted space-time, contrast and other multi-features and SVM. Zhao et al. [15] identified smoke based on the Adaboost algorithm by taking advantage of the color and other characteristics of smoke, which could also effectively distinguish fog from smoke. Yuan et al. [16] proposed a smoke recognition method based on dual-threshold Adaboost to recognize black smoke and white smoke with bimodal distribution characteristics. However, SVM, AdaBoost and other traditional classifiers still have some limitations. When the amount of smoke image features is small, these classifiers perform well. When the amount of smoke image features is large, the classification accuracy of this classifier needs to be improved.

At present, deep learning [17,18] has been successfully applied in image classification, pattern recognition and other fields. Wei et al. [19] integrated static and dynamic smoke texture information and proposed a smoke texture recognition framework based on the cascaded convolutional neural network (CNN)[20,21], which could improve the

accuracy of smoke recognition. However, the static and dynamic texture information was processed separately in this method, which increased the complexity of the algorithm and affected the real-time performance of smoke detection. Xu et al. [22] used synthetic smoke images and real smoke images to train the CNN model based on the domain adaptability method [23]. Although this method could reduce the false detection rate of smoke recognition, the use of the synthetic smoke images would affect the performance of the training model in the actual scene. On the whole, the current video smoke detection methods have strong scene pertinence. In a fixed scene, these video smoke detection methods can achieve a high recognition accuracy, but in the face of the change of weather, light and other interference factors, these smoke detection false alarm rate is high.

Aiming at the above problems, on the basis of summarizing the current smoke detection methods, this paper proposes an urban fire smoke detection method combining multiple Gaussian mixture model and a modified single shot-multibox detector (MSSD) to satisfy the requirements of anti-interference, real-time smoke detection.

This paper is organized as follows. Section 2 is the related works for the smoke recognition. We detailed show the proposed smoke recognition method in section 3. The experiments are displayed in section 4. Section 5 concluded this paper.

2. Related Works

As an important means of fire detection, smoke detection has been widely used in fire and explosion detection and early warning. The traditional detection technology based on smoke sensor has a small monitoring range, high cost of laying in factories, warehouses, forests and other large areas, and such sensors are easy to age and reduce sensitivity. In recent years, video smoke detection technology has attracted much attention from researchers at home and abroad because of its advantages such as short response time, high sensitivity and large coverage area.

Current video smoke detection methods mainly rely on visual features such as motion, color, shape, transparency and texture. Reference [24] proposed a detection method using smoke color and motion characteristics. Firstly, background extraction and color filtering were used to obtain candidate smoke regions. The light flow was then characterized by the mean and variance of its speed and direction. Finally, BP neural network was used to complete the classification and recognition. The dimension of the obtained eigenvector was too low to effectively describe the different manifestations of smoke in complex environment. Reference [25] proposed a cumulative motion model and used integral graph to rapidly estimate the direction of smoke motion. This method assumed that smoke moved upward, and its application scope was limited. Subsequently, reference [26] proposed a smoke detection method combining the dual mapping frame feature and AdaBoost. The first layer mapped each frame into blocks and extracted the edge direction histogram, edge intensity histogram, LBP histogram, edge intensity density, color and saturation density and other features of each image block. The second layer mapped the image into partitions and calculated the mean value, variance, peak state and skewness of each block feature. These statistics were eventually used to train and classify the AdaBoost model. Reference [27] proposed a smoke detection method based on contour and wavelet transform for fixed camera video. The Hidden Markov model (HMM) was used to analyze the periodic change of smoke profile in time domain. Smoke usually had a certain transparency,

and its visual features were affected by background. If background interference could be overcome, the difficulty of smoke identification could be effectively improved. Aiming at this problem, reference [28] analyzed the mixing mechanism of smoke and background, built a set of smoke foreground extraction method, and solved the mixing coefficient by using sparse expression, local smoothing and other constraints. This method could reduce background interference to some extent and improve the accuracy of smoke recognition.

In the aspect of smoke texture feature extraction, GLCM, LBP, Wavelet are the most widely used. Reference [29] implemented a set of real-time flame and smoke detection system based on GLCM analysis of smoke texture. Reference [30] proposed a smoke detection method based on pyramid histogram sequence. Firstly, the pyramid sampling was a three-layer multi-scale structure, and the LBP and LBPV features of different modes were extracted from each layer. Finally, the LBP and LBPV feature sequences were spliced together as smoke texture features, and the BP neural network was used for classification. However, there are many false detections in the existing methods in practical application, mainly for the following reasons: 1) Smoke presents various states under different environments. The data set selected in the existing references is small, so it is difficult to train a stable and reliable classifier to fit its complex manifestations. 2) Smoke visual feature extraction has always been a difficulty in video smoke detection. Relying only on static features is insufficient to distinguish smoke from some smoke-like objects (such as clouds, fountains, etc.). How to construct a stable and efficient feature extraction algorithm to integrate static and dynamic information in video becomes the key to reduce false smoke detection.

Traditional classifiers such as SVM and decision tree perform well in small data sets, but it is difficult to improve the classification accuracy when the amount of data is large. In recent years, deep neural network (DNN) has been successfully applied in the field of computer vision. By establishing a hierarchical network model structure similar to human brain, DNN extracts features from the input data step by step from the bottom level to the top level, so as to better obtain the mapping relationship from the bottom level signal to the top level semantic. Convolutional neural networks (CNN), as one of the most important network models, have made breakthroughs in face recognition and image classification, driven by big data and high-performance computing. In 2012, reference [31] used deeper CNN on the famous ImageNet image data set to obtain the best results in the world, reducing the recognition error rate from 26% to 15%, and greatly improving the accuracy of large-scale image recognition. Deep convolutional neural networks can take the original image as input to learn features from the bottom pixel level to the top representation level, transforming the mode of manual feature extraction to the mode of automatic feature learning from data. Moreover, the model is more effective on big data.

In this paper, CNN is introduced into smoke texture feature extraction and a cascaded CNN smoke texture recognition framework is proposed to integrate static and dynamic texture information. The original image is used as input in the static texture and the optical flow sequence of the original image is used as input in the dynamic texture. The final experimental results show that the proposed method achieves better performance in the accuracy and false detection rate of smoke recognition.

3. Proposed Smoke Detection Method

The urban smoke detection algorithm flow is shown in figure 1. Firstly, the image of urban smoke scene is obtained by the camera. Secondly, the background subtraction method is used to process the collected sequence images, and the foreground image of moving object is extracted preliminarily. Further, the noise in the foreground image is removed by morphology operation. Finally, the trained SSD model [32] is used to classify and identify the moving object area. If it is judged as smoke, smoke alarm will be issued.

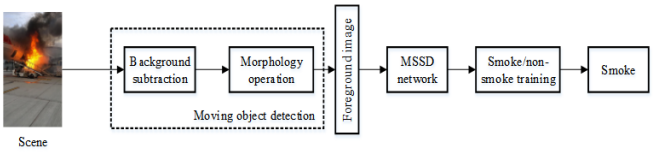

Fig. 1. Flowchart of the smoke detection algorithm

3.1. Moving Object Detection

In complex scenes, clouds, fogs and other objects similar to smoke will interfere with smoke detection. In video stream sequence images, moving object extraction can effectively filter out the interference of static objects, and then reduce the false positive rate of CNN recognition. In addition, the filtering of non-smoke areas in video images can reduce the running time of subsequent smoke detection algorithms and further improve the efficiency of smoke detection.

A. Gaussian mixture model (GMM) The features of each pixel in the moving object image are represented by N Gaussian models. The Gaussian mixture model is updated with each frame of the image [33]. The Gaussian mixture model is matched with each pixel in the current image. If there is no match, it determines that the pixel is the foreground point. If a match can be made, the pixel is the background point. Set the gray value of a pixel in the image as g, and the pixel gray value from time 1 to time t is expressed as $(a_1, a_2, \cdots, a_i, \cdots, a_t)$. A detailed description of pixel gray values requires N Gaussian distributions. In the description, N Gaussian distributions need to be mixed by weighting, so as to obtain the probability density function:

$$g(a_t) = \sum_{i=1}^{t} \lambda_{t,i} \times \gamma(a_i, v_{t,i}\Sigma_{t,i}) \tag{1}$$

Where $\gamma(a_i, v_{t,i}\Sigma_{t,i})$ represents the probability density function of Gaussian distribution. $v_{t,i}$ is the mean value of the Gaussian distribution. $\lambda_{t,i}$ is the weight. $\Sigma_{t,i}$ is the covariance matrix of the Gaussian distribution. The probability density function of Gaussian distribution is calculated as follows:

$$\gamma(a_i, v_{t,i}\Sigma_{t,i}) = \frac{1}{2\pi(d/2)|\Sigma_{t,i}|^{0.5}} s^{-0.5(a_t - v_{t,i})\sum_{i=1}^{t}(a_t - v_{t,i})} \tag{2}$$

Where, the dimension of a_t is d. When the observation point is updated as a_{t+1}, the pixel values and the mean value $\tau_{t,i}$ of N Gaussian function distributions are compared respectively. Formula (3) is used as the judgment rule to select Gaussian function.

$$|a_{t+1} - \tau_{t,i}| < o \times \varepsilon_{t,i}^2, i = 1, 2, \cdots, N \tag{3}$$

In the formula, o represents a user-defined parameter, whose value is usually 2.5. When equation (3) is satisfied, the $i-th$ Gaussian matches a_{t+1}.

In the absence of a Gaussian match, the variance and mean remain the same. Equations (4) to (6) represent the generated parameters after the Gaussian function is matched with a_t, they are updated by:

$$\tau_{t,i} = (1 - \vartheta) \times \tau_{t-1,i} + \vartheta \times a_t \tag{4}$$

$$\varepsilon_{t,i}^2 = (1 - \vartheta) \times \varepsilon_{t-1,i}^2 + \vartheta \times (a_t - \tau_{t,i})^2 \tag{5}$$

$$\vartheta = \delta \times \lambda_{t,i} \tag{6}$$

In the formula, ϑ is the parameter learning rate. N Gaussian distribution weights are updated through equation (7):

$$\lambda_{t,i} = (1 - \delta) \times \lambda_{t-1,i} + \vartheta \times E \tag{7}$$

In the formula, the updating rate of the Gaussian distribution weight is expressed as $\delta \in [0, 1]$. When $E = 1$, the matching distribution is satisfied. When $E = 0$, it satisfies $N - 1$ distribution.

In order to improve the reliability of the background model, the weight is normalized. After normalization, we can get $\bar{\lambda}_{t,i} = \lambda_{t,i} / \sum_{i=1}^{n} \lambda_{t,i}$. The descending order of each distribution should follow the ratio $\bar{\lambda}_{t,i}/\varepsilon_{t,i}$. The reliable background part is selected as the first x distribution, and the number of background distribution needs to be controlled by H_1, where $x = \underset{x'}{argmin}(\sum_{i=1}^{x'} \bar{\lambda}_{t,i} > H_1)$. The reliability of the background distribution needs to be expressed by the ratio $\bar{\lambda}_{t,i}/\varepsilon_{t,i}$. When the weight is inversely proportional to the variance, it can be concluded that the samples belonging to this distribution are affected by the probability of sample occurrence. The higher probability denotes the more concentrated the samples. The probability is higher of the distribution belonging to the background distribution.

B. Improved GMM In common moving object detection and tracking methods, foreground model is seldom used, it is only used as auxiliary. However, in the modeling of improved Gaussian mixture model, the generated foreground model when background matching fails is used to make a comprehensive judgment of foreground combined with

short-term stability index [34]. Based on the Gaussian mixture model, if all the corresponding N background models fail to match the current pixel value, the variance and mean of the model with the minimum weight will be replaced by the larger value and the current pixel value. The generated model at this time is the foreground model $\gamma(\tau_f, \varepsilon_f)$. If the threshold value H_f is greater than the difference between the mean value of the prospect model and the subsequent points, formula (8) is needed to update the prospect model and Formula (9) is used to calculate the short-term stability:

$$R = \frac{P\sum_{i=0}^{P} a_{t+1}^2 - \sum_{i=0}^{P} a_{t+i}}{P(P-1)} \tag{8}$$

$$\tau_{f,t+1} = (1 - v_f)\tau_{f,t} + v_f a_{t+1} \tag{9}$$

Where $v_f \in [0, 1]$ represents the learning rate of the prospect model. P represents the range of sliding window frames. When matching the current pixel, the foreground model is preferred, which can reduce the decision risk caused by the error of matching background model and foreground point.

The shapes of moving objects are different, and the calculation window length P value of short-term stability of moving objects with uniform colors is set in the range of 2-5. If the color of the moving object is rich, the change time of pixel value is too short, which will cause the object to be mistaken for the background [35]. At this point, P value should be controlled within 5-20, and there is a positive correlation between P value and detection effect. However, if the P value is too large, the response speed of the index will be slow. After the stability is obtained through formula (9), the threshold R_{th} of judgment can be obtained as:

$$R_{th} = R_{min} + \frac{R_{max} - R_{min}}{L} \tag{10}$$

The maximum stability of the current L frame is R_{max} and the minimum stability is R_{min}. L is a constant. The condition that the current pixel is judged to be a foreground point is that all the current pixels in the range of consecutive L frames exceed the short-term stability threshold.

The change of stability is positively correlated with the relation of pixel value. Therefore, the stability can fully describe the emergence and persistence of prospects. If the pixels in the moving object area change in a short time, it is easy to detect the background incorrectly. This situation can be effectively avoided by using the short-term stability index to improve the Gaussian mixture function. When the traditional Gaussian model detects the object's motion speed is too slow, it cannot be detected. In this paper, the method of combining short-term stability index and prospect model is adopted to solve the common problems of Gaussian mixture model and improve the detection effect of object's motion.

C. Elimination of light mutation The improved Gaussian mixture model can improve the background extraction due to the slow motion of the detected object. However, pixels will be wrongly detected as foreground pixels, once it is disturbed by light mutation.

The movement of the object and the change of illumination will lead to the mutation of pixel gray scale, and the mutation caused by the change of illumination is larger than that caused by the movement of the object [36]. Therefore, in order to achieve better detection and tracking effect, it is necessary to eliminate light mutation. The specific methods are as follows.

In order to determine whether the Gaussian distribution in the current frame matches the pixel, a new parameter w needs to be set for each pixel in the image. The value of w is set to improve the background estimation. If N Gaussian functions and pixels can be matched with each other, then $w = 0$. If not, the value of w is 1. If the pixel mutation area is large, it is caused by illumination mutation. The number of $w = 1$ in the image is counted. If formula (11) is satisfied, it indicates that the background illumination mutation occurs in this time.

$$\frac{\sum_{i=1}^{K} w_i}{K} > H_1 \tag{11}$$

When illumination mutation occurs, the proportion of pixel area with gray mutation occurs in the image is the threshold value, which is represented by H_1. In the actual experiment, $H_1 = 0.66$. If the image can conform to formula (11), the background pixel is determined to be a pixel block with $w = 1$. In this case, the background model is updated by improving the background model, and the one with the smallest weight among the first N distributions is replaced by the Gaussian distribution based on the gray value of pixel blocks as the mean value, and it becomes the background model.

3.2. Morphological Processing

Isolated noises, small gaps and holes still exist in foreground images extracted by improved GMM background subtraction method. In order to realize the complete extraction of moving object, the foreground image is further processed by morphological method. Let O be the object to be processed, S be the structural element. It uses S to perform open and close operations on O respectively, i.e,

$$OPEN(O, S) = (O \Theta S) \oplus S \tag{12}$$

$$CLOSE(O, S) = (O \oplus S) \Theta S \tag{13}$$

Where, Θ is the corrosion operation on the image. $\oplus$ is the expansion operation of the image. Firstly, the open operation is performed on each image, then the gaps and holes in the image are filled by closed operation after the noise is eliminated. Finally, the processed results are obtained.

3.3. Network Design for Smoke Detection

A. InceptionV3 network InceptionV3 is one of the Google Inception series networks. Based on InceptionV2, this network proposed a new Network in Network (NIN) structure

[37]. InceptionV3 is a new network constructed according to the NIN architecture of the Inception module. The Inception module for InceptionV3 is shown in figure 2.

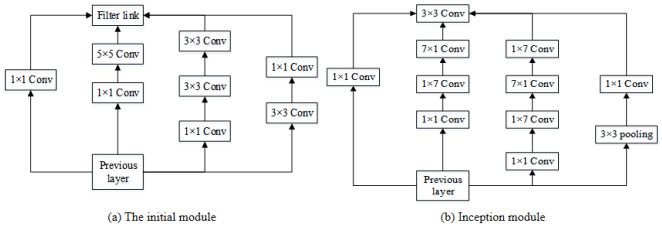

Fig. 2. Structure of InceptionV3 Inception module

As shown in figure 2, InceptionV3 module reduces the parameter number of the model by extracting features of different scales [38]. At the same time, multi-scale features can improve the recognition ability of the model. Studies have proved that the convolution layer with 1×1 convolution kernel can realize feature conversion, improve network recognition ability and change the number of channels output by the convolution module through a small amount of computation. The fifth ninth modules of InceptionV3 use a larger convolution kernel to capture more abstract features. For the convolution layer with 7×7 convolution kernel and prone to generate a large number of parameters, InceptionV3 uses 1×7 convolution layer and 7×1 convolution layer to improve the efficiency and reduce the over-fitting. Experiments show that this asymmetric convolution structure can deal with more and more spatial features and increase feature diversity.

B. SSD and MSSD InceptionV3 network model is mainly used to classify smoke. There is a limitation that there can only be one smoke in each image and the background cannot be too complicated. To solve this problem, many object detection methods have been proposed, such as using color and texture features to detect smoke. The accuracy of smoke detection is greatly reduced because of overlapping and false detection.

At present, there are many deep learning methods for object detection that are widely used in urban areas, such as towards real-time object detection with region proposal networks (Faster-RCNN) [39], Object-detection via region-based fully convolutional networks (RFCN) [40], Single shotmultibox detector (SSD), You only look once (YOLO) [41] and RetinaNet [42]. In addition, smoke detection has higher requirements on the real-time performance of the algorithm, so it is necessary to choose a more efficient algorithm as far as possible under the condition that the accuracy meets the requirements.

SSD is an algorithm that uses the same deep neural network to detect and identify detected objects in images. SSD generates a series of candidate boxes of different sizes. The offset value of the labeled box and the candidate box is calculated to match them. Typically, each annotation box will match multiple candidate boxes. SSD considers candidate boxes with IoU greater than 0.5 as positive samples, and sets other boxes as negative samples.

SSD uses VGG16 as the backbone network to form multi-scale detection by extracting feature maps with different sizes from Conv4_3, FC7, Conv7_2, Conv6_2, Conv8_2 and Pool6. The network structure diagram of SSD is shown in figure 3.

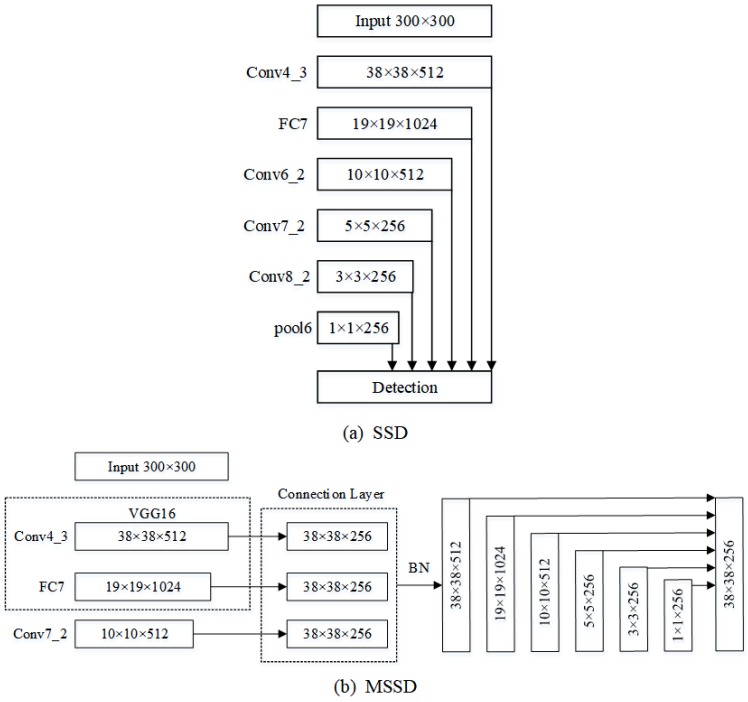

Fig. 3. Model structure of SSD and MSSD

The sizes of feature map are 5×5 pixels, 3×3 pixels and 1×1 pixels, respectively. Low-level feature maps are beneficial to small object detection, while advanced feature maps with the size of 38×38 pixels, 19×19 pixels and 10×10 pixels are beneficial to large object detection. SSD uses 3×3 convolution layer and 1×1 convolution layer with 1024 channels to replace full connection layer and lose layer to reduce model parameters, improve computational efficiency and effectively prevent over-fitting. The loss function of SSD is:

$$L(x, c, l, g) = \frac{1}{N}(L_{conf}(x, c) + \alpha L_{loc}(x, l, g)) \tag{14}$$

Where $L(x, c, l, g)$ is the total loss value. x is the convolution eigenvalue. c is the real class. l is the predicted box position coordinate value. g is the real box position coordinate value. L_{loc} is the smoothing loss between the prediction box and the real box. N is the number of candidate boxes. L_{conf} is the softmax loss under multi-class confidence. α is the weight of L_{loc}. Multibox Loss is composed of two parts. The former uses Softmax to calculate the classification loss, and the latter predicts the position through local loss.

SSD provides a solid foundation for smoke detection tasks, with the primary benefit of fast detection. However, experiments have shown that SSD has the following problems: small objects are difficult to identify; Some background regions are identified as object objects, and objects are repeatedly identified. To solve these problems, the SSD model is modified to the MSSD model without increasing the number of model parameters and reducing the detection speed.

The relationship between each layer used for prediction in the SSD model is independent of each other. The MSSD model fuses these layers with feature maps of different proportions so that they communicate with each other and improve accuracy. However, this approach is not suitable for fusing feature maps less than 10×10 in the MSSD model, because there is little information to merge. Feature graphs of three larger layers are combined to generate feature graphs of 3838 pixels. Feature pyramid network (FPN) is generated [43]. Finally, the FSSD model extracts features from FPN.

The structure of the MSSD model is shown in figure 3b. The MSSD model uses VGG16 as the main backbone network. The feature map of Conv7_2 becomes 10×10. FC7 and Conv7_2 use bilinear interpolation to adjust the feature maps to 38×38 pixels and then connect them to Conv4_3, in which the number of channels in the fusion layer is 768(256+256+256). BN acts on the fusion layer, the number of channels in this layer is reduced to 512, and 5 convolution layers are used to reduce the size of the feature graph. Finally, six feature maps of different sizes (38×38 pixels, 19×19 pixels, 10×10 pixels, 5×5 pixels, 3×3 pixels and 1×1 pixel) are obtained and used for multi-scale prediction. This method can combine shallow detail features with high-level semantic features to identify small objects better than SSD models and reduce false detection rates. However, the speed of MSSD model is slightly lower than that of SSD model.

The loss function partially satisfies the following equations:

$$L = L_{cls} + L_{reg} \tag{15}$$

$$L_{cls} = \lambda_1 L_{tr} + \lambda_2 L_{tcl} \tag{16}$$

$$L_{reg} = \lambda_3 L_r + \lambda_4 L_{sin} + \lambda_5 L_{cos} \tag{17}$$

Where L_{cls} represents the classification loss of TR and TCL. L_{reg} represents the regression loss of r, $cos\theta$ and $sin\theta$. L_{tr} and L_{tcl} are the cross drop losses of tr and tcl respectively. L_r, L_{sin} and L_{cos} are smooth-L1 losses. Set λ_1, λ_2, λ_3, λ_4 and λ_5 to 1, satisfying the following formula:

$$[L_r, L_{cos}, L_{sin}]^T = SmoothedL_1[(\hat{r} - r)/r, \hat{cos\theta} - cos\theta, \hat{sin\theta} - sin\theta]^T \tag{18}$$

C. Optimized MSSD In this paper, it is only necessary to determine whether the detection object is smoke or not, without multi-category classification. Optimized MSSD is simplified as IFSSD. The IFSSD model structure is shown in figure 4.

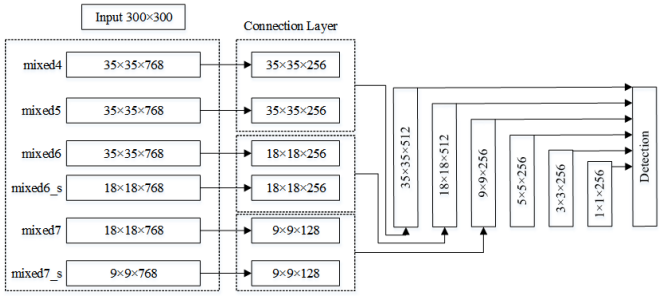

Fig. 4. IFFSD-based smoke detection structure

In order to reduce the parameter number and improve the detection effect of small objects, InceptionV3 is adjusted in this paper to achieve multi-scale detection based on different receptive fields. The mixed7 layer has been removed in this article. IFSSD model adds 1×1 convolution after mixed6 layer with step size of 2 and channel number of 768, and then conducts batch normalization. The feature map of the newly added layer mix6_s is 18×18 pixels. The feature map size of mixed7 layer has been changed from 35×35 pixels to 18×18 pixels. After mixed7, the convolution layer with convolution kernel 1×1, step size 2 and channel number 768 is added. The feature map of mixed7_s is 99 pixels.

The IFSSD model is similar to the FSSD model in that large feature maps are combined. In InceptionV3 of IFSSD model, four layer modules with different feature graph sizes are selected respectively, namely mixed4, mixed5, mixed6_s and mixed7_s. In the first connection layer, the convolution layer with convolution kernel of 1×1, step size of 1 and channel number of 256 is applied to mixed4 and mixed5. The purpose of this method is to reduce the model parameters without changing the size of the feature graph, so as to reduce the number of parameters in the model and enhance the nonlinear recognition ability of the model. The size of the feature graph generated after convolution is

35×35 pixels, and then the feature graph generated by convolution of mixed4 and mixed5 is fused together. In the second connection layer, a convolution layer with 11 convolution kernel, 2 step size and 256 channels are added after mixed5 module, and the size of the generated feature map is 18×18 pixels. After mixed6_s, it adds a convolution layer with 1×1 convolution kernel, 1 step size and 256 channels. The generated feature map is 1818 pixels.

Then, the two feature maps are combined. In the third connection layer, a convolutional layer with 1×1 convolution kernel, step size 2 and channel number 128 is added to mixed6_s to generate a feature graph of $9 \times 9 \times 128$. Finally, the convolution layer with convolution kernel of 1×1, step size of 1 and channel number of 128 are added to mixed7_s to generate a feature graph of $9 \times 9 \times 128$, and the two feature graphs are combined.

In the IFSSD model, $35 \times 35 \times 512$, $18 \times 18 \times 512$ and $9 \times 9 \times 256$ connection layers are used to generate FPN. After the connection layer of $9 \times 9 \times 256$, it adds a SAME convolution layer with a convolution kernel of 3×3, step size of 2 and channel number of 256, and generates a feature map with a size of 5×5 pixels. Then, it adds a convolution kernel of 3×3, step size of 1×1, channel number of 256 and uses VALID convolution layer to generate feature graph size of 3×3 pixels. Finally, after layer $3 \times 3 \times 256$, it adds a VALID convolution layer with convolution kernel 3×3, step size 1 and channel number 256, and generates feature graph size 1×1 pixel.

It can be seen that, the MSSD model uses bilinear interpolation to get the same dimensions from three different layers in VGG16 and fuses them together to get an output layer and generate FPN. IFSSD model uses 1×1 convolution to modify the channel number and image size of mixed4, mixed5, mixed6_s, and mixed7_s layers in InceptionV3, and fuses the above modified convolution layers in pairs to obtain three different layers, and then generates FPN based on these three layers.

The difference in performance between one-stage and two-stage algorithms is mainly caused by the imbalance of a large number of foreground background categories. In the two-stage algorithm, in the candidate box stage, score and non-maximum suppression (NMS) [44] are adopted to filter out a large number of negative samples. Then it fixes the proportion of positive and negative samples in the classification and regression stage to make foreground and background relatively balanced. The one-stage algorithm needs to generate about 100kb candidate locations. Despite repeated sampling, training is still dominated by a large number of negative samples. The IFSSD model used in this paper belongs to the one-stage model, and Focal Loss is used to replace the original loss function.

Focal Loss is mainly to solve the problem of serious imbalance of positive and negative sample ratio in one-stage object detection. The loss function is reduced, and the weight of a large number of simple negative samples in training can also be understood as a kind of difficult sample mining. Focal Loss is modified based on the cross entropy Loss function. The cross entropy loss of dichotomies can be calculated as:

$$L = \begin{cases} -lg(y') & y = 1 \\ -lg(1 - y') & y = 0 \end{cases} \tag{19}$$

In the formula, L is the loss value of cross entropy. y is the true classification. y' is the prediction classification. y' is the output of the activation function, and the value is between 0 and 1. It can be seen that for the positive cross entropy, the larger output probability denotes the smaller loss. For negative samples, the smaller output probability denotes the smaller loss. At this time, the loss function is slow and may not be optimized in the iterative process of a large number of simple samples.

Focal Loss adds a constant factor γ to the original basis to reduce the Loss and focus more on difficult and misclassified samples. In addition, the balance factor a_t is added to balance the uneven proportion of positive and negative samples. The specific formula is as follows:

$$F_L(p_i) = -a_t(1-p_i)^{\gamma} lg(p_i) \tag{20}$$

p_i is the probability of different categories. a_t is the ratio of positive samples to negative samples. If the foreground category is a_t, the corresponding background category is $1 - a_t$.

For the data set of single target detection, Anchor can be reasonably set according to the size of target object identified in the data set and the width to height ratio of real frame, which cannot only better detect the location of target, but also improve the detection speed and detection accuracy of network model. In order to obtain the specific aspect ratio of smoke in the experimental sample, 50 images are randomly selected from the sample data set, and then about 200 single fire images are manually intercepted, and the width and height of single smoke images are recorded at the same time.

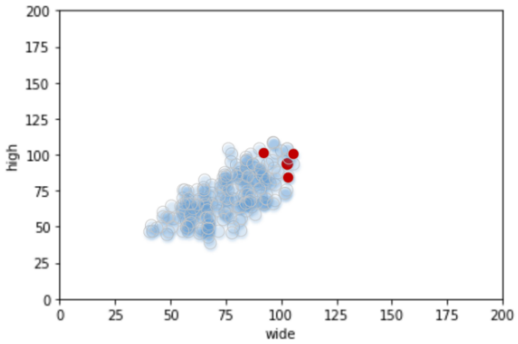

Fig. 5. Wide and high scatter diagram

It can be seen from figure 5 that the smoke scatter plot of width and height is relatively concentrated. It generally presents a linear distribution. This is because in the wild, smoke varies little in aspect ratio, so it is more concentrated. The aspect ratio of the Anchor of the initial lightweight SSD is set as 1/3, 1/2, 1, 2 and 3. By drawing these five straight lines in the aspect scatter plot, the setting of the aspect ratio of Anchor can be more intuitively understood. In figure 6, 1/3, 1/2, 1, 2 and 3 correspond to the red, black, pink, blue and green lines in the figure respectively.

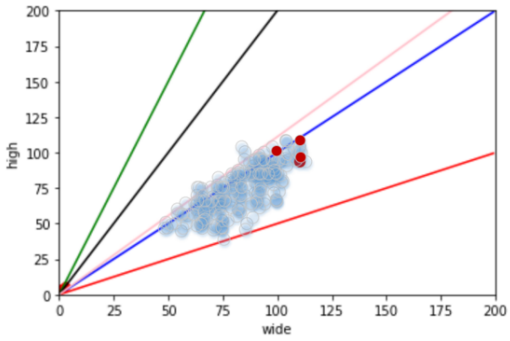

Fig. 6. Anchor setup for the original lightweight SSD

It can be seen that the original aspect/height ratio setting of Anchor is not reasonable, so this paper resets the aspect/height ratio parameters of Anchor. Firstly, the aspect ratio of all single smoke images obtained from the sample database is calculated, and then the new aspect ratio parameters are analyzed.

D. Training stage The Blending strategy in the training stage consists of two prediction parts: base learner and meta learner. Firstly, the data sets are divided (training set, validation set, test set) and input into each base learner in the first layer of the model. Each base learner will produce a prediction result in the validation set and the test set. Secondly, the historical load values corresponding to the prediction results, validation, set and samples in the test set of the first layer are recombined into a new sample set, which is used as the input of the meta-learner model in the second layer. Finally, the meta-learner fits the prediction results of the validation set, predicts the prediction results of the test set, and obtains the final prediction results. Blending models can make full use of the differences of prediction principles of different models, so as to achieve complementary advantages among them. The specific training method is:

- Step 1. For a data set $S = (y_n, x_n), n = 1, 2, \cdots, N$, where x_n is the feature vector of n-th sample. y_n is the load value corresponding to the $n - th$ sample. n is the number of features, that is, each feature vector is $[x_1, x_2, \cdots, x_n]$. Firstly, the data is divided into training set $S_1 = (y_v, x_v), v = 1, 2, \cdots, V$, and validation set $S_2 = (y_i, x_i), i = 1, 2, \cdots, I$. The size of the data set after partitioning is $S_1 < S_2 < S$. The test set to be predicted in this paper is $S_3 = (y_t, x_t), t = 1, 2, \cdots, T$. It selects K different models as the base learner, fits the K base learners on S_1, makes prediction on S_2 and S_3, and gets the predicted value of S_2 as A_{ik}. The predicted value on S_3 is B_{tk}. A_{ik} and B_{tk} are combined with target values corresponding to S_2 and S_3 respectively to constitute the new training set $D_1 = (A_{ik}, y_i), k = 1, 2, \cdots, K$, testing set $D_2 = (A_{tk}, y_t), k = 1, 2, \cdots, K$.
- Step 2. The newly generated data sets D1 and D2 are taken as the input data of the second layer, and the meta-learner model of the second layer is used to fit on D1 and predict on D2 to obtain the final prediction result.

XGBoost algorithm adds penalty term and second-order Taylor expansion on the basis of gradient lifting tree (GBDT). The objective f_k is obtained by reducing the loss function. The loss function is shown in Equation (21):

$$L^{(t)} = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k) \tag{21}$$

Where, the first part is the sum of training errors, where $\hat{y}_i$ is the predicted value. y_i is the target value, t is the iteration number. The second part is the sum of regularization coefficients, that is, the sum of tree depth T and leaf node weight w is:

$$\Omega(f) = \gamma T + 0.5\lambda ||w||^2 \tag{22}$$

Where, γ and λ represent the penalty coefficient of the model. Second order Taylor expansion is carried out for Equation (22):

$$\begin{aligned} L^{(t)} &\cong \sum_{i=1}^{n} (g_i f_t(x_i) + 0.5 h_i f_t^2(x_i)) + \Omega(f_t) \\ &\cong \sum_{i=1}^{n} (g_i f_t(x_i) + 0.5 h_i f_t^2(x_i)) + \gamma T + 0.5\lambda \sum_{j=1}^{T} w_j^2 \\ &\cong \sum_{j=1}^{T} [(\sum_{i \in I_j} g_i) w_j + 0.5 (\sum_{i \in I_j} h_i + \lambda)] + \gamma T \end{aligned} \tag{23}$$

Where $g_i = \partial_{\hat{y}_i^{(i-1)}} l(y_i, \hat{y}_i^{(t-1)})$, $h_i = \partial^2_{\hat{y}_i^{(i-1)}} l(y_i, \hat{y}_i^{(t-1)})$ are first and second derivatives of the loss function. Define $G_j = \sum_{i \in I_j} g_i$, $H_j = \sum_{i \in I_j} h_i$, so formula (23) can be written as:

$$L^{(t)} \cong \sum_{j=1}^{T} [G_j w_j + 0.5 (H_j + \lambda) w_j^2] + \gamma T \tag{24}$$

Taking the partial derivative with respect to w and substituting it into the objective function, we can obtain:

$$L^{(t)} \cong -0.5 \sum_{j=1}^{T} \frac{G_j^2}{H_j + \lambda} + \gamma T \tag{25}$$

In XGBoost model, the smaller loss function denotes the better model effect. Through the greedy algorithm, the model traverses all the different tree structures, splits nodes, and calculates the gain of each split node. The gain L_{Gain} is:

$$L_{Gain} \cong 0.5 (\frac{G_L^2}{H_L + \lambda} + \frac{G_R^2}{H_R + \lambda} - \frac{G_L + G_R}{H_L + H_R + \lambda}) - \gamma \tag{26}$$

Where, the first two terms are the gains after left and right segmentation, and the third term is the overall segmentation gain.
The training pseudocode based on Bagging-blending multiple models fusion method is shown in **Algorithm 1**.

Algorithm 1 Bagging-blending fusion method

Input: Image set $M = (x_n, y_n), n = 1, 2, \cdots, N$;
Output: Bagging-blending fusion model;

Step 1. The data is divided into L identical training sets $M_{1L} = (x_v, y_v), v = 1, 2, \cdots, V$ validation set $M_{2L} = (x_i, y_i), i = 1, 2, \cdots, I$, the testing set $M_{3L} = (x_t, y_t), i = t, \cdots, T$.
Step 2. L single models are trained and predicted.
for all $l = 1$ to L **do**
    The prediction error ∂_l is obtained by fitting M_{1L} prediction on M_{2L}.
end for
Step 3. Pearson correlation coefficient algorithm is used to analyze the correlation r_{xy} between model prediction error ∂_l, and L_1 single models with large differences are selected, where $L_1 \leq L$.
Step 4. Embed L_1 in the Bagging ensemble algorithm.
for all $l = 1$ to L_1 **do**
    Using l_1 as the base learner of Bagging ensemble model, the integrated model L_2 is obtained.
end for
Step 5. The grid search G is used to optimize L_2.
for all $l_2 = 1$ to L_2 **do**
    G_{l_2} is used to search for the optimal hyperparameter of the model.
end for
Step 6. The L_2 model with adjusted superparameters is trained as the base learner at layer 1.
for all $l = 1$ to L_2 **do**
    s is used to fit on the training set M_{ls}, and the prediction is made on the validation set M_{2s} and training set M_{3s} respectively, and the prediction result is P_{is} and Q_{ts}.
end for
Step 7. Constitute a new training set $M_{1new} = (P_{is}, y_i), s = 1, 2, \cdots, L_2$, testing set $M_{2new} = (Q_{ts}, y_t), s = 1, 2, \cdots, L_2$.
Step 8. The XGBoost model of Blending Layer 2 meta-learner is used to fit on the new training set and predict on the new test set.

4. Experiments and Analysis

The CPU used in the experiment is Intel Core I7-6700, main frequency 3.4GHz, 16G memory, NVIDIA graphics TX1060, video memory is 64GB under Windows10 operating system. The smoke detection network model is built and trained by TensorFlow.

4.1. Training and Testing Data Sets

The data set used in the experiment contains 15000 images with 128×128pixels containing positive and negative samples. Combined with the common moving object in-

terference items in the actual smoke scene, the negative samples are divided into four categories: people, cars, motorcycles and buses, with about 3000 images in each category. Smoke sample data are from the Bilkent, Visor and KLFS. Smoke area images are extracted from different videos. It can guarantee the diversity of smoke images to prevent over-fitting of training network. Part of positive smoke samples are shown in figure 7(a). Negative samples come from ImageNet data set and network search, and some negative samples are shown in figure 7(b).

Fig. 7. Data set. (a) Positive samples; (b) Negative samples

In this paper, 70% of the positive and negative samples are used as the training set and 30% as the test set. GPU is used to train and test the proposed network model in figure 8.

In this paper, different parameters are set for the model, and the optimal parameters are selected through cross-validation. The initial learning rate is 0.001. After several iterations, it is fine-tuned to 10^{-4}. The minimum normalized boundary is set to 0.0001, the redecay is 5×10^{-4}, and the image batch is 16. The detection rate of the model in this paper for 128×128 pixel images is 29f/s. The change curve of loss value in the training process is shown in figure 8. As the number of iterations increases, the loss value of IFFSD model decreases continuously. After 100 iterations, the decline tends to flatten and it is stable after 200 iterations.

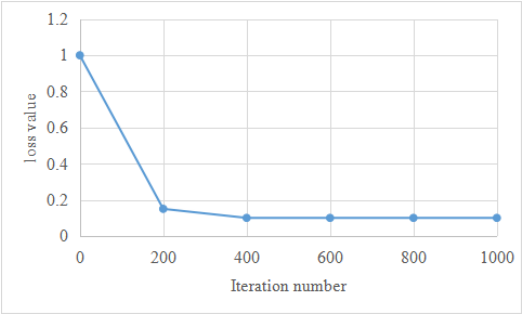

Fig. 8. Change of loss value during IFSSD training

4.2. Smoke Recognition Results

Based on the moving object extraction method and the image processing function of OpenCV, the contour of each moving area detected in the smoke scene is extracted by using the minimum rectangle box. Furthermore, the moving areas extracted from each rectangular frame label are uniformly converted into images with 128×128 fixed size. The smoke discrimination of each moving object is carried out by the network model after training and testing.

In this paper, IFFSD, SSD and MSSD models are used to detect smoke data sets respectively. Table 1 shows the average detection results, in which detected objects are regarded as positive samples and undetected objects as negative samples. The IoU (Intersection over Union) threshold is 0.5. Figure 9 shows the intuitive result. There are four test results, namely, IoU value is less than or equal to 0.5 (FP), IoU value is greater than 0.5 (TP), and without real object FN and TN. TN samples are not counted in this paper. The accuracy rate can be calculated by:

$$P = \frac{T_P}{T_P + F_P} \times 100\% \tag{27}$$

Where T_P is the detected positive sample. F_P is the detected negative sample.

Table 1. Detection results with different models

Model	Accuracy/%	Model size/MB	Detection time per image/s
SSD	5.4	28.1	36
MSSD	92.6	28.1	39
IFFSD	94.8	19.5	21

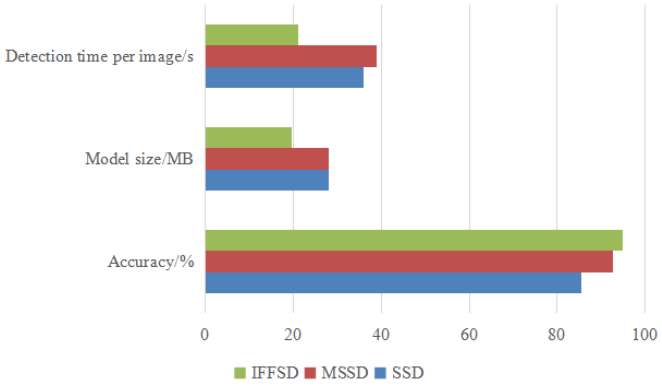

Fig. 9. Detection results with different models

In terms of detection accuracy, the accuracy of IFFSD model is 94.8%, which is 9.4% higher than SSD model and 2.2% higher than FSSD model. In terms of image detection rate, the detection time of IFFSD model is 21s, which is lower than 36s of SSD model and 39s of MSSD model.

As shown in table 2 and figure 10, the accuracy of IFFSD model is higher than that of Histogram of Oriented gradient (HOG)+Support Vector Machine (SVM) and Deformable Parts Model (DPM)+SVM [45], it is increased by 14.1% and 16.4% respectively. This is because the traditional object detection algorithm is only suitable for the image with obvious features and simple background. However, in practical application, the background is complex and changeable, and the object to be detected is complex and changeable, so it is difficult to complete the object detection through general abstract features. However, deep learning can extract rich features of the same object to complete object detection. However, IFFSD model is larger than HOG+SVM and DPM+SVM in model size, because IFFSD model has more parameters. In terms of efficiency, the IFFSD model is close to the traditional method.

Table 2. Comparison of IFSSD and other detection methods

Model	Accuracy/%	Model size/MB	Detection time per image/s
HOG+SVM	80.7	20.8	28
DPM+SVM	78.4	3.6	26
Faster RCNN	87.0	53.1	29
YOLOv2-DarkNet-19	85.4	24.6	20
IFFSD	94.8	20.8	28

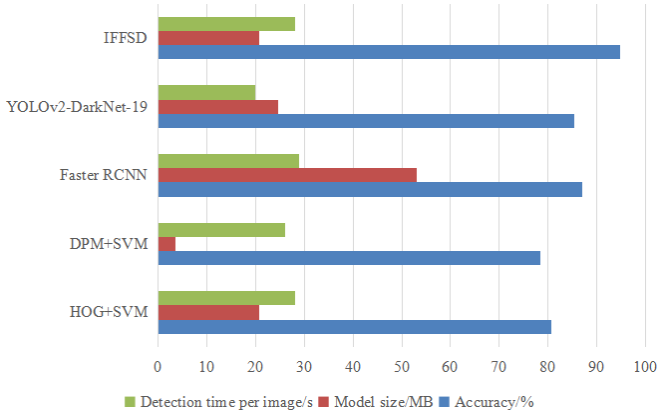

Fig. 10. Comparison of IFSSD and other detection methods

Compared with Faster RCNN and YOLOv2-Darknet19, the accuracy of IFFSD model is increased by 7.8% and 9.4% respectively due to the addition of FPN structure and

Inception backbone network. In terms of detection time, IFFSD model is close to Faster RCNN model.

For all input foreground images, the IFFSD model should be able to identify smoke and non-smoke moving objects. However, in practice, due to the limited training sample data, it is difficult for the negative sample to cover all non-smoke motion interference items, which makes some out-of-sample moving objects cause false positives to smoke detection. Figure 11 is the detection result.

Fig. 11. The detection result with proposed method.

In order to further verify the effectiveness of the urban smoke detection algorithm proposed in this paper, 120 videos of different scenes are obtained through self-shooting and Internet downloading, in which 72 images are smoke videos as shown in figure 12(a). There are 48 smoke-free videos as shown in figure 12(b).

Fig. 12. Video set. (a) Examples of smoke videos; (b) Examples of non-smoke videos

The response time from outbreak of fire to fire alarm signal should not exceed 20s. On this basis, in the video with smoke, if the smoke alarm is issued within 20s after the smoke breaking, the proposed method in this paper is considered to meet the requirements of smoke detection. If there is no smoke alarm in 20s, it is judged to be missed detection. In the smoke-free video, if the non-smoke area is identified as the smoke area, it will be judged as a false positive. Some experimental results are shown in table 3.

In the 72 videos with smoke, only No. 18 and No. 43 have the smoke alarm response more than 20s, with a missed alarm rate of 2.7%. In the 48 smoke-free videos, only the

Table 3. Model performance of smoke detection

Video sequence	Video type	Response time/s	Type of alarm
1	smoke	4.31	true
2	smoke	7.42	true
11	smoke	0.17	true
18	smoke	23.83	missed
43	smoke	25.60	missed
52	smoke	8.52	true
71	smoke	5.68	true
72	smoke	1.83	true
73	Non-smoke	–	–
74	Non-smoke	–	–
93	Non-smoke	–	false
119	Non-smoke	–	–
120	Non-smoke	–	–

No. 93 video has false positive, with a false rate of 2.0%. Overall, the smoke detection accuracy of 120 videos is 97.5%. In the smoke alarm video with smoke (except No.18 and 43), the longest response time is 8.52s of No. 52 video, and the shortest response time is 0.17s of No. 11 video. According to statistics, the average response time of smoke alarm is 4.58s, which is far lower than the specified 20s. The above experimental results show that under the current experimental conditions, the urban video smoke detection algorithm proposed in this paper can meet the real-time and accuracy requirements of most actual smoke detection.

The main factors affecting the accuracy of smoke detection include: 1) the number of positive samples used for SSD model training is limited, and smoke items not covered by positive samples may cause missing reports; 2) The negative sample data is limited, such as water fog with similar color and movement of smoke, white automobile exhaust and other interference items not covered by the negative sample, will cause certain false positives. By increasing the diversity of positive and negative samples, the accuracy of video smoke detection can be further improved.

5. Conclusions

An urban smoke detection method based on MGMM and improved SSD is proposed. This method does not need to judge and recognize all the objects in the video, but only input the moving objects in the foreground image into the IFFSD model for smoke recognition after filtering the static objects in the video image by the improved Gaussian mixture model. This method can not only reduce the interference of static objects, but also improve the efficiency of smoke detection. Experimental results show that this method is feasible and effective in many complex environments. The early fire smoke is mostly gray due to low temperature, so the positive samples selected in this paper are mainly gray smoke, aiming to solve the problems of real-time and accuracy of video smoke detection in complex scenes. In order to make the smoke detection method more general, the diversity of positive and negative samples can be increased in the subsequent work to meet the smoke

detection requirements with more scenes and further improve the accuracy of smoke detection. In the future, more advanced deep learning method will be adopted to improve the detection rate.

Acknowledgments. The author greatly appreciates the expert review comments.

References

1. Lh, A., Xg, A., Sz, A., et al.: "Efficient attention based deep fusion CNN for smoke detection in fog environment-ScienceDirect," *Neurocomputing*, 434, 224-238. (2021)
2. Saponara, S., Elhanashi, A., Gagliardi, A.: "Real-time video fire/smoke detection based on CNN in antifire surveillance systems," *Journal of Real-Time Image Processing*, 18, 889-900. (2021)
3. Liu, H., Lei, F., Tong, C., et al.: "Visual smoke detection based on ensemble deep cnns," *Displays*, 69, 102020. (2021)
4. Millan-Garcia, L., Sanchez-Perez, G., Nakano, M., et al.: "An early fire detection algorithm using IP cameras," *Sensors*, Vol. 12, No. 5, 5670-5686. (2012)
5. Favorskaya, M., Pyataeva, A., Popov, A.: "Verification of Smoke Detection in Video Sequences Based on Spatio-temporal Local Binary Patterns," *Procedia Computer Science*, Vol. 60, No. 1, 671-680. (2015)
6. Zhou, Z., Shi, Y., Gao, Z., et al.: "Wildfire smoke detection based on local extremal region segmentation and surveillance," *Fire Safety Journal*, 85, 50-58. (2016)
7. Dimitropoulos, K., Barmpoutis, P., Grammalidis, N.: "Higher order linear dynamical systems for smoke detection in video surveillance applications," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 27, No. 5, 1143-1154. (2016)
8. Jia, Y., Chen, W., Yang, M., et al.: "Video Smoke Detection with Domain Knowledge and Transfer Learning from Deep Convolutional Neural Networks," *Optik-International Journal for Light and Electron Optics*, Vol. 240, No. 8, 166947. (2021)
9. Vijayalakshmi, S., Muruganand, S.: "Smoke detection in video images using background subtraction method for early fire alarm system," *2017 2nd International Conference on Communication and Electronics Systems (ICCES). IEEE*, 167-171. (2017)
10. Cruz, H., Eckert, M., Meneses, J., et al.: "Efficient Forest Fire Detection Index for Application in Unmanned Aerial Systems (UASs)," *Sensors*, Vol. 16, No. 6, 893. (2016)
11. Ye, S., Bai, Z., Chen, H., et al.: "An effective algorithm to detect both smoke and flame using color and wavelet analysis," *Pattern Recognition and Image Analysis*, Vol. 27, No. 1, 131-138. (2017)
12. Shi, J., Wang, W., Gao, Y., et al.: "Optimal placement and intelligent smoke detection algorithm for wildfire-monitoring cameras," *IEEE Access*, Vol. 8, 72326-72339. (2020)
13. Kim, J., Bae, S.: "Smoke Detection Method Using Local Binary Pattern Variance in RGB Contrast Imag," *Journal of Korea Multimedia Society*, Vol. 18, No. 10, 1197-1204. (2015)
14. Prema, C., Vinsley, S., Suresh, S.: "Multi Feature Analysis of Smoke in YUV Color Space for Early Forest Fire Detection," *Fire Technology*, Vol. 52, No. 5, 1319-1342. (2016)
15. Zhao, Y., Li, Q., Gu, Z.: "Early smoke detection of forest fire video using CS Adaboost algorithm," *Optik -International Journal for Light and Electron Optics*, Vol. 126, No. 19, 2121-2124. (2015)
16. Yuan, F., Fang, Z., Wu, S., et al.: "Real-time image smoke detection using staircase searching-based dual threshold AdaBoost and dynamic analysis," *IET Image Processing*, Vol. 9, No. 10, 849-856. (2015)
17. Maguire, G., Chen, H., Schnall, R., et al.: "Smoking cessation system for preemptive smoking detection," *IEEE Internet of Things Journal*, Vol. 9, No. 5, 3204-3214. (2021)

18. Yuan, Y., Xu, Z., Lu, G.: "SPEDCCNN: Spatial Pyramid-Oriented Encoder-Decoder Cascade Convolution Neural Network for Crop Disease Leaf Segmentation," *IEEE Access*, vol. 9, 14849-14866. (2021) doi: 10.1109/ACCESS.2021.3052769. (2021)
19. Ye, W., Zhao, J., Zhao, Y., et al.: "Smoke detection based on Surfacelet transform and dynamic texture," *Comput. Eng.*, Vol. 41, No. 2, 203-208. (2015)
20. Almeida, J., Huang, C., Nogueira, F., et al.: "EdgeFireSmoke: A Novel Lightweight CNN Model for Real-Time Video FireCSmoke Detection," *IEEE Transactions on Industrial Informatics*, Vol. 18, No. 11, 7889-7898. (2022)
21. Yang, M., Tjuawinata, I., Lam, K.: "K-Means Clustering With Local d_X-Privacy for Privacy-Preserving Data Analysis," *IEEE Transactions on Information Forensics and Security*, vol. 17, 2524-2537. (2022) doi: 10.1109/TIFS.2022.3189532. (2022)
22. Xu, G., Zhang, Y., Zhang, Q., et al.: "Deep Domain Adaptation Based Video Smoke Detection using Synthetic Smoke Images," *Fire Safety Journal*, Vol. 93, 53-59. (2017)
23. Jang, H., Lee, J.: "Machine learning versus econometric jump models in predictability and domain adaptability of index options," *Physica A: Statistical Mechanics and its Applications*, Vol. 513, 74-86. (2019)
24. Wang, S., Guo, Q., Xu, S., et al.: "A moving target detection and localization strategy based on optical flow and pin-hole imaging methods using monocular vision," *2021 IEEE International Conference on Real-time Computing and Robotics (RCAR). IEEE*, 147-152. (2021)
25. Esfahlani, S.: "Mixed reality and remote sensing application of unmanned aerial vehicle in fire and smoke detection," *Journal of Industrial Information Integration*, Vol. 15, 42-49. (2019)
26. Liu, B., Sun, B., Cheng, P., et al.: "An embedded portable lightweight platform for real-time early smoke detection," *Sensors*, Vol. 22, No. 12, 4655. (2022)
27. Kaabi, R., Bouchouicha, M., Mouelhi, A., et al.: "An efficient smoke detection algorithm based on deep belief network classifier using energy and intensity features," *Electronics*, Vol. 9, No. 9, 1390. (2020)
28. Tao, H., Zheng, P., Xie, C., et al.: "A three-stage framework for smoky vehicle detection in traffic surveillance videos," *Information Sciences*, Vol. 522, 17-34. (2020)
29. Martins, L., Guede-Fernndez, F., Valente de Almeida R, et al.: "Real-Time Integration of Segmentation Techniques for Reduction of False Positive Rates in Fire Plume Detection Systems during Forest Fires," *Remote Sensing*, Vol. 14, No. 11, 2701. (2022)
30. Dong, Y., Wu, H., Li, X., et al.: "Multiscale symmetric dense micro-block difference for texture classification," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 29, No. 12, 3583-3594. (2018)
31. Krizhevsky, A., Sutskever, I., Hinton, G.: "ImageNet Classification with Deep Convolutional Neural Networks," *Advances in neural information processing systems*, Vol. 25, No. 2. (2012)
32. Yang, J., He, W., Zhang, T., et al.: "Research on Subway Pedestrian Detection Algorithms Based on SSD Model," *IET Intelligent Transport Systems*, Vol. 14, 7553. (2020)
33. Yin, S., Zhang, Y., Karim, S.: "Large scale remote sensing image segmentation based on fuzzy region competition and Gaussian mixture model," *IEEE Access*, Vol. 6, 26069-26080. (20181)
34. Zhang, X., Hill, D.: "Load stability index for short-term voltage stability assessment," *2019 IEEE Power & Energy Society General Meeting (PESGM). IEEE*, 1-5. (2019)
35. Zhang, D., Shafiq, M., Wang, L., et al.: "Privacy-preserving remote sensing images recognition based on limited visual cryptography," *CAAI Transactions on Intelligence Technology*, (2023). https://doi.org/10.1049/cit2.12164
36. Yin, S., Wang, L., Shafiq, M., Teng, L., Laghari, A., Khan, F.: "G2Grad-CAMRL: An Object Detection and Interpretation Model Based on Gradient-Weighted Class Activation Mapping and Reinforcement Learning in Remote Sensing Images," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 16, 3583-3598. (2023) doi: 10.1109/JSTARS.2023.3241405

37. Suzuki, S., Shouno, H.: "A study on visual interpretation of network in network," *2017 International Joint Conference on Neural Networks (IJCNN)*, 903-910. (2017) doi: 10.1109/IJCNN.2017.7965948.
38. Zhao, Y., Xie, K., Zou, Z., He, J.: "Intelligent Recognition of Fatigue and Sleepiness Based on InceptionV3-LSTM via Multi-Feature Fusion," *IEEE Access*, vol. 8, 144205-144217. (2020)
39. Yin, S., Li, H.: "Hot Region Selection Based on Selective Search and Modified Fuzzy C-Means in Remote Sensing Images," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 13, 5862-5871. (2020) doi: 10.1109/JSTARS.2020.3025582.
40. Xu, C., Hong, X., Yao, Y., et al.: "Multi-Scale Region-based Fully Convolutional Networks," *Neurocomputing*, 500-505. (2020)
41. V. E.K. and C. Ramachandran.: "Real-time Gender Identification from Face Images using you only look once (yolo)," *2020 4th International Conference on Trends in Electronics and Informatics (ICOEI)(48184)*, 1074-1077. (2020)
42. Zhu, M., et al.: "Arbitrary-Oriented Ship Detection Based on RetinaNet for Remote Sensing Images," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 14, 6694-6706. (2021) doi: 10.1109/JSTARS.2021.3082526.
43. Tan, Y., Wu, P., Zhou, G., et al.: "Combining Residual Neural Networks and Feature Pyramid Networks to Estimate Poverty Using Multisource Remote Sensing Data," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, Vol. 13, No. 99, 553-565. (2020)
44. Tao, H., Lu, X.: "Smoke Vehicle Detection Based on Spatiotemporal Bag-Of-Features and Professional Convolutional Neural Network," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 10, 3301-3316. (2020) doi: 10.1109/TCSVT.2019.2920657.
45. Hebbalaguppe, R., Garg, G., Hassan, E., Ghosh, H., Verma, A.: "Telecom Inventory Management via Object Recognition and Localisation on Google Street View Images," *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 725-733. (2017) doi: 10.1109/WACV.2017.86.

Hao Han is with the Guizhou Fire and Rescue Brigade and Department of Natural Resources of Guizhou Province. He has published several papers and won several awards. His research direction: Image processing, big data analysis.

Received: December 18, 2022; Accepted: May 20, 2023.