Extraction and Classification of Image Features for Fire Recognition Based on Convolutional Neural Network

Ruiyang Qi, Zhiqiang Liu*

College of Information Engineering, Inner Mongolia University of Technology, Hohhot 010051, China

Corresponding Author Email: liuzq@imut.edu.cn

ABSTRACT

Fire image monitoring systems are being applied to more and more fields, owing to their large monitoring area. However, the existing image processing-based fire detection technology cannot effectively make real-time fire warning in actual scenes, and the relevant fire recognition algorithms are not robust enough. To solve the problems, this paper tries to extract and classify image features for fire recognition based on convolutional neural network (CNN). Specifically, the authors set up the framework of a fire recognition system based on fire video images (FVIFRS), and extracted both static and dynamic features of flame. To improve the efficiency of image analysis, a Gaussian mixture model was established to extract the features from the fire smoke movement areas. Finally, the CNN was improved to process and classify the fire feature maps of the CNN. The proposed algorithm and model were proved to be feasible and effective through experiments.

1. INTRODUCTION

Fire, as a frequent and highly damaging disaster, is very difficult to prevent. The early detection of fire has long been a focus among domestic and foreign scholars [1-5]. Depending on the detection method, fire detector can be categorized into two types: traditional detectors using sensors, and fire image monitoring systems based on image processing and pattern recognition [6, 7]. With the development of security and monitoring equipment, fire image monitoring systems are being applied to more and more fields, owing to their large monitoring area.

Scholars at home and abroad have achieved a lot in the recent research of fire image monitoring systems [8-10]. For example, Du and Fan [11] extracted several dynamic and static features of flames (e.g., color, shape, and moving speed), and constructed a forest fire recognition system based on support vector machine (SVM), which improves the ability of real-time fire recognition.

In fire recognition, the traditional image processing techniques have difficulty in feature extraction, and face a low accuracy [12-16]. Kim and Kim [17] converted the color space format of fire video images, trained the deformation convolution network (DCN) on the image set, and acquired the flame features that adapt to geometric changes, thereby improving the fire recognition effect. Considering the difference between fire source and interference source, Singh et al. [18] updated the detection algorithm for the aircraft fire detection system, established a recurrent neural network (RNN) based on long short-term memory (LSTM) and real-time dynamic information, and connected the measured signals into a time series of features to train the established network. Sayyed et al. [19] extracted multiple features of the flame, including chroma, area change rate, circularity, number of sharp corners, and centroid displacement; After segmenting the fire images, they created a fast fire recognition model

based on time smoothing and logarithmic regression, and tested the model on self-made flame videos. Drawing on YCbCr color space (YCbCr: light intensity, blue component intensity, and red component intensity) and local phase quantization (LPQ) histogram, Mahmoud and Ren [20] detected the unique colors and textures of flames, extracted the spatial and frequency features, and completed robust and accurate detection and recognition of flames based on the SVM.

In summary, the existing image processing-based fire detection techniques have been primarily applied to identify suspected regions and fit the flame foreground contours. Despite their high recognition accuracy, these techniques cannot realize real-time warning of fires in actual scenes, and their recognition algorithms are not robust enough [21-27]. To solve the problems, this paper tries to extract and classify image features for fire recognition based on the CNN. Section 2 sets up the framework of a fire recognition system based on fire video images (FVIFRS), and extracts both static and dynamic features of flame. The static features include color, morphology, and texture, while the dynamic features include frequency and centroid displacement. To improve the efficiency of image analysis, Section 3 extracts the features from the fire smoke movement areas, with the aid of Gaussian mixture model. Section 4 processes and classifies the fire feature maps based on the CNN. Finally, experimental results verify the feasibility and effectiveness of our model and algorithm.

2. EXTRACTION OF STATIC AND DYNAMIC FLAME FEATURES FROM FIRE IMAGES

Figure 1 illustrates the framework of FVIFRS. The key of fire recognition is to extract the flame features from video images. High-quality flame features help to differentiate the

flame from non-fire objects effectively. The features of flame images can be roughly divided into static features like color, morphology, and texture, and dynamic features like frequency and centroid displacement.

This paper calculates the flame color in the RGB color space (RGB: red, green, and blue). Let I_R, I_G and I_B be the components of the three channels in the RGB color space. Then, we have:

$$\begin{cases} 0.25 \le \dfrac{I_G(a,b)}{I_R(a,b)+1} \le 0.65 \\[2mm] 0.05 \le \dfrac{I_B(a,b)}{I_R(a,b)+1} \le 0.45 \\[2mm] 0.20 \le \dfrac{I_B(a,b)}{I_G(a,b)+1} \le 0.60 \end{cases} \tag{1}$$
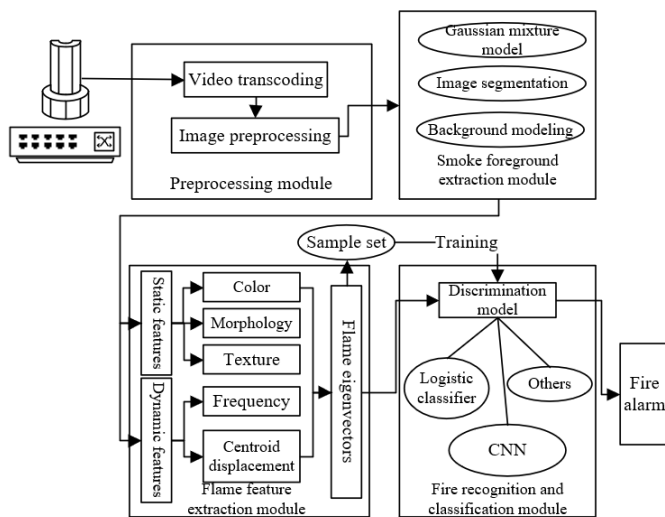

Figure 1. Framework of FVIFRS

In a fire image, the flame is dark blue in the inner layer, dark red or light yellow in the middle layer, and colorless or yellow in the outer layer. Therefore, the three channel components satisfy the inequality $I_R>I_G>I_B$.

During combustion, the flame jitters and extends/retracts randomly, and its tip becomes a sharp corner. This paper quantifies the contours of the flame with a sharp tip with circularity, which reflects how similar a pattern is to a standard circle. Let CIR be the circularity of the flame; P be the size of the foreground flame; C be the circumference of the foreground flame. Then, we have:

$$CIR = \frac{4\pi P}{C^2} \tag{2}$$

Formula 2 shows, CIR satisfies the inequality $1>CIR>0$. The closer CIR is to 1, the smoother the flame contours; the closer CIR is to 0, the coarser the flame contours, and the more complex the flame shape.

To better capture the flame structure in the image or the periodic changes of the flame structure, this paper computes every element (i, j) in the gray-level co-occurrence matrix (GLCM), where i and j are the gray-levels of a pixel and its adjacent pixel, respectively. In this way, the flame texture was quantified as $O(i,j)$. Further, the flame eigenvectors can be derived. Entropy S_{ENT}, energy S_{ENE}, and contrast S_{CON} are three common statistics of flame features:

$$S_{ENT} = -\sum_i \sum_j O(i,j)\log O(i,j) \tag{3}$$

$$S_{ENE} = \sum_i \sum_j O(i,j)^2 \tag{4}$$

$$Con = \sum_i \sum_j (i-j)^2 O(i,j) \tag{5}$$

Next, the two-dimensional (2D) Mallat algorithm was adopted for wavelet decomposition of video fire images, aiming to preserve more valuable image information, while ensuring the computing efficiency and data compression effect. Figure 2 explains the process of wavelet decomposition of a fire image by 2D Mallat algorithm. As can be seen from the figure, the fire image is firstly decomposed by rows to obtain the low-frequency component DF and high-frequency component GF in the horizontal direction. Next, DF and GF are further decomposed by columns to obtain the low-frequency component DD and high-frequency component GG in the vertical and horizontal directions, the horizontal high-frequency and vertical low-frequency component GD, and the horizontal low-frequency and vertical high-frequency component DG.
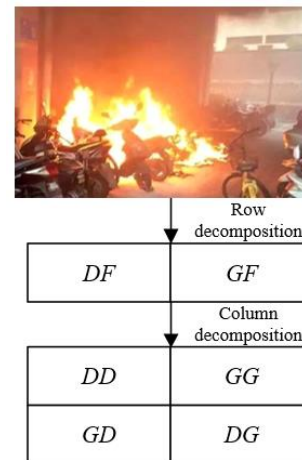

Figure 2. Wavelet decomposition of a fire image by 2D Mallat algorithm

DD, DG, GD, and GG were used to represent the low-frequency component, the high-frequency component of horizontal details, the high-frequency component of vertical details, and the high-frequency component of diagonal details of the original image, respectively. Among them, the three high-frequency components of the original fire image, namely, DG, GD, and GG, carry lots of information about flame texture. Therefore, the maps of the three high-frequency components were adopted to describe flame texture.

Let $T(i, j)$ be the eigenvalue of pixel (i, j) in the original fire image; $q(l, k)$ be the (l, k)-th wavelet coefficient in the window centering on pixel (i, j). From each high-frequency component map, a $(2m+1)*(2m+1)$ window was used to extract the macro-feature T of energy for flame texture:

$$T(i, j) = \frac{1}{(2m+1)^2} \sum_{l=i-m}^{i+m} \sum_{k=j-m}^{j+m} |q(l, k)| \tag{6}$$

Let $A=\{1, 2, 3\}$ denote the three channels of the RGB color space; $n=\{1, 2, 3\}$ denote the three high-frequency component maps in the three channels. Then, the wavelet energy T_n^A of the n-th map in channel A can be calculated by:

$$T_n^A = \int \left(T(\vec{e}) \right)^2 d\vec{e}, \vec{e} \in R^2 \tag{7}$$

After 2D wavelet transform, the signal energy was calculated for each high-frequency component map. Through normalization, the flame texture eigenvector $\{T_n^A\}$ could be obtained for a series of fire images. Let (CP_m, CP_m) be the coordinates of the center pixel; $GRAY_m$ be the grayscale of the center pixel; BRI_l be the brightness of the eight neighborhood pixels. Then, the texture can be calculated based on the local binary pattern (LBP) operator:

$$LBP(CP_m, CP_m) = \frac{\sum_{l=0}^7 2^l \cdot sign(BRI_l - GRAY_m)}{225} \tag{8}$$

If m=1, then $GRAY_m$ should be compared one by one with the eight neighborhood pixels of the center pixel in the 3×3 window. After converting decimal numbers through binary mapping to $\{0, 1\}$, the LBP value of the center pixel texture could be obtained.

Repeated experiments have demonstrated that the flame features of the frequency of 10Hz are independent of the material of the burning object. This paper proposes a method to calculate the flame foreground change rate, which reflects the flame frequency. Let VP_m and VP_{m-1} be the set of pixels in the flame foreground in the current frame and the previous frame, respectively; ○ be the subtraction between the two sets. Then, flame frequency can be characterized by:

$$FC_m = \left| \frac{(VP_m \cup VP_{m-1}) \circ (VP_m \cap VP_{m-1})}{VP_m} \right| \tag{9}$$

Formula 9 shows that the flame movement can be measured by the difference between adjacent frames in flame foreground area, divided by the flame foreground area in the current frame. If the flame changes obviously at the current moment, then the ratio FC_m will be greater than the threshold HV. Considering the stochasticity of flame jitter and interference movement, the flame frequency of the current frame is equivalent to the mean change rate of foreground region of the current frame and the previous (M-1) frames:

$$FC_m' = \frac{\sum_{i=0}^{M-1} FC_{m-i}}{M} \tag{10}$$

During combustion, the flame moves differently from other objects. The centroid movement is far less active than the flame. In fact, the centroid jumps rather stably throughout the combustion. Figure 3 details the calculation flow of centroid displacement. The algorithm of centroid displacement can be calculated by:

$$\begin{cases} a' = \frac{1}{N} \iint_D a\lambda(a, b) d\tau \\ b' = \frac{1}{N} \iint_D b\lambda(a, b) d\tau \end{cases} \tag{11}$$

If the collected flame images are digital images, the foreground area could be assumed as P, and the weights of all pixels could be deemed as equal. Then, the centroid displacement can be obtained through the following discretized method:

$$\begin{cases} a' = \frac{\sum n_i a_i}{\sum n_i} = \frac{\sum a_i}{P} \\ b' = \frac{\sum n_i b_i}{\sum n_i} = \frac{\sum b_i}{P} \end{cases} \tag{12}$$
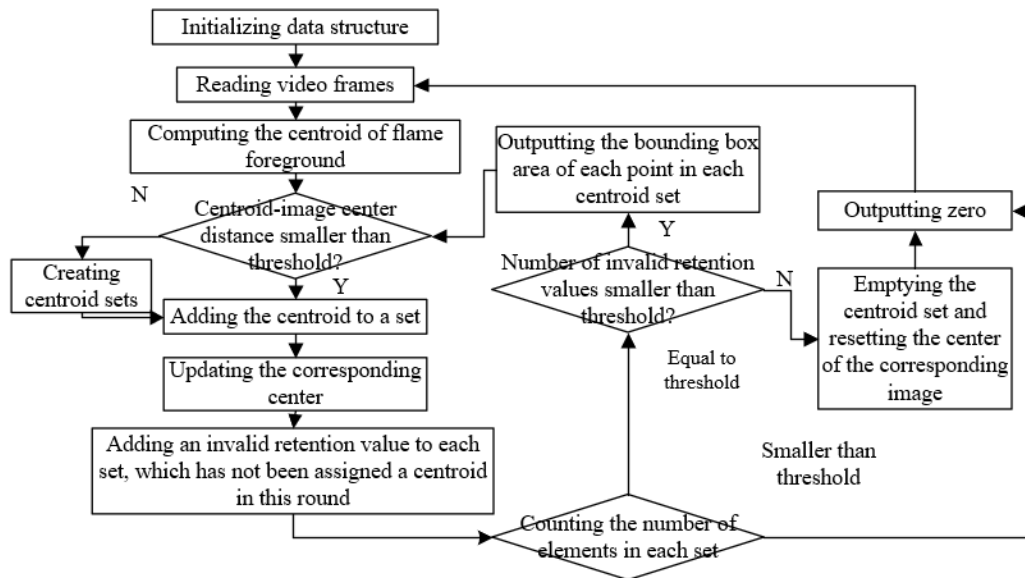

Figure 3. Calculation flow of centroid displacement

3. FEATURE EXTRACTION FROM SMOKE MOVEMENT AREAS

If the fire video images are coherent and continuous, the movement areas must be extracted to improve the efficiency of image analysis. The extraction of smoke movement areas is an important link in fire detection and recognition. However, most moving object extraction methods rely too much on threshold selection. To overcome the defect, this paper adopts Gaussian mixture model to extract smoke movement areas from fire images.

Let a_h be the value of a constantly moving pixel in fire video images at time h; $\{a_1, a_2, a_3, ..., a_h\}$ be the movement of the pixel in a period of time. Different types of images (e.g., grayscale image, and color image) have different meanings of a_h in terms of grayscale or vector. Let N be the number of Gaussian distributions related to computing speed; ω_{i-h} be the weight of the i-th Gaussian distribution allocated at time h in the model, that is, the probability of a pixel to satisfy that Gaussian distribution; v_{i-h} be the mean of the i-th Gaussian distribution at time h; Γ_{i-h} be the variance matrix of the i-th Gaussian distribution at time h. Then, the pixel a_h in the Gaussian mixture model at time h can be defined as:

$$\Phi(a_h) = \sum_{i-1}^{N} \omega_{i-h} \times \delta(a_h, v_{i-h}, \Gamma_{i-h}) \qquad (13)$$

The probability density function $\delta(a_h, v_{i-h}, \sum_{i-h})$ of the i-th Gaussian distribution can be defined as:

$$\delta(a_h, v_{i-h}, \Gamma_{i-h}) = \frac{1}{\sqrt{2\pi\Gamma_{i-h}}} \phi^{\frac{(a_h - v_{i-h})^2}{2\Gamma_{i-h}}} \qquad (14)$$

For a color image in fire video, the variance of independent color components can be denoted as $\Gamma_{i-h} = \tau^2_N \cdot II$, where II represents the integral image of the said color image. The Gaussian mixture model can be regarded as a dynamic adaptive movement detector. When the Gaussian mixture model is applied to process the current frame of the fire video, whether the Gaussian model of a pixel needs to be updated can be determined in real time according to the parameters of that pixel. In this way, it is possible to track the pixel changes dynamically and adaptively in the image. However, it is quite laborious to implement calculation based on the Gaussian model of each pixel. Thus, this paper chooses the k-means algorithm to update the models. The new pixel value a_h was substituted in turn to N Gaussian distributions, and subject to position matching. The constraint can be defined as:

$$\left| a_h - v_{i,h} \right| \leq 2.5\tau_N \qquad (15)$$

Let β be the time constant of the speed for updating the model of a Gaussian distribution. Then, the weight, mean, and variance of the n-th Gaussian distribution ensuring a_h to obey Gaussian distribution can be calculated by:

$$\begin{cases} \omega_{n,h} = (1-\beta)\omega_{n,h-1} + \beta \\ v_{n,h} = (1-\varepsilon)v_{n,h-1} + \varepsilon a_h \\ \tau_{n,h} = (1-\varepsilon)\tau_{n,h-1} + \varepsilon(a_h - v_{n,h})^\psi (a_h - v_{n,h}) \end{cases} \qquad (16)$$

Based on β, the learning rate ε can be calculated by:

$$\varepsilon = \beta\delta(a_h, v_N, \tau^2_N) \qquad (17)$$

If a_h does not satisfy the n-th Gaussian distribution, there is no need to update the mean or variance of Gaussian distribution. Then, the weight can be updated by:

$$\omega(i,h) = (1-\beta)\omega_{i,N-1} \qquad (18)$$

The next step is to choose the flame background model from the N reconstructed Gaussian distribution models. Taking GA Gaussian distributions as the background, we have:

$$GA = argmin_r\left(\sum_{l=1}^{r} \omega_l > \psi \right) \qquad (19)$$

If a_h is a background pixel, then a_h satisfies any of the GA Gaussian distribution models; If a_h is a flame foreground pixel, then a_h does not satisfy any of the GA Gaussian distribution models.

4. CNN-BASED PROCESSING AND CLASSIFICATION OF FIRE FEATURE MAPS

This paper constructs a depth wise separable CNN, which is less complex in computation than standard CNN. Let $CR_W \times CR_W \times N$ and $CR_W \times CR_W \times M$ be the size of the input and output fire feature maps, respectively, where CR_W is the width and height of the feature map; N_A be the number of channels for an input image; $CR_L \times CR_L$ be the size of a standard convolution. Then, the computing load of the standard convolution equals $CR_L \times CR_L \times N \times M \times CR_W \times CR_W$. Figure 4 shows the principles of depth convolution and point-by-point convolution in depth wise separable convolution. The total computing load of in depth wise separable convolution is the sum of the total computing loads of the two convolution patterns: $CR_L \times CR_L \times CR_W \times CR_W + N \times M \times CR_W \times CR_W$.
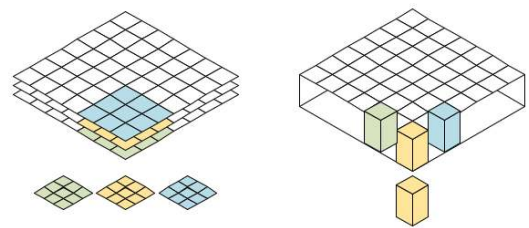

Figure 4. Principles of depth convolution and point-by-point convolution

The computing load ratio of depth wise separable convolution to standard convolution can be expressed as:

$$\frac{CR_L \times CR_L \times N \times CR_W \times CR_W + N \times M \times CR_W \times CR_W}{CR_L \times CR_L \times N \times M \times CR_W \times CR_W} = \frac{1}{M} + \frac{1}{CR_l^2} \qquad (20)$$

Since M takes a large value, if the convolution kernel is of the size 3×3, the depth wise separable convolution can reduce the computing load of standard convolution by 9 times. The network adopts rectified linear unit (ReLU) as the activation function of the network:

$$g(a) = \begin{cases} a & a \geq 0 \\ 0 & a < 0 \end{cases} \qquad (21)$$

Let $\{a_1, a_2, ..., a_m\}$ be the feature data of a batch of images in network training. Then, $\{a_1, a_2, ..., a_m\}$ could be the output of a layer in the network and the input of the next layer. Thus, the batch normalization, which normalizes the output of a previous layer by subtracting the batch mean and dividing by the batch standard deviation, can be completed in three steps:

$$\lambda \leftarrow \frac{1}{n}\sum_{i=1}^{n} a_i \qquad (22)$$

$$\tau^2 = \frac{1}{n}\sum_{i=1}^{n}(a_i - \lambda) \qquad (23)$$

$$a_i^* \leftarrow \frac{a_i - \lambda}{\sqrt{\tau^2 + \theta}} \qquad (24)$$

Sometimes, the image feature data are asymmetric, or the selected activation function performs poorly on image features with a variance of 1. To give full play to nonlinear transform, batch normalization can be backed up by the following operation:

$$b_i^* \leftarrow \eta a_i^* + \alpha \qquad (25)$$

where, η and α are two parameters that need to be updated in network training.

This paper chooses gradient descent to update the kernel parameters and bias in CNN training, and to calculate the value of loss function. The gradient ∇LO_A of individual sample and the global gradient ∇LO of the image feature data could be calculated based on a few randomly selected samples of input images. The image data are assumed to be sufficiently large. In this case, the estimated gradient is basically equal to the actual gradient. Then, we have:

$$\frac{\sum_{j=1}^{n} \nabla LO_{A_j}}{n} \approx \frac{\sum_a \nabla LO_a}{m} = \nabla LO \qquad (26)$$

That is:

$$\nabla LO \approx \frac{1}{n}\sum_{j=1}^{n} \nabla LO_{A_j} \qquad (27)$$

The weight and bias of the CNN could be updated by computing the partial derivative of the loss function relative to weight layer by layer. Finding the partial derivative of the loss function LO relative to e_K:

$$\xi_K^{(e)} = \frac{\partial LO}{\partial e_L} = -(h - e_K) \qquad (28)$$

When the error of the loss function propagates from e in the k+1-th layer to c, we have:

$$\frac{\partial e^{(k+1)}}{\partial c^{k+1}} = e^{(k+1)}\left(1 - e^{(k+1)}\right) \qquad (29)$$

When the error of the loss function propagations from the k+1-th layer to the k-th layer, we have:

$$\frac{\partial c^{(k+1)}}{\partial e^{(k)}} = \omega^{(k)}, \frac{\partial c^{(k+1)}}{\partial \omega^{(k)}} = e^{(k)}, \frac{\partial c^{(k+1)}}{\partial r^{(k)}} = J \qquad (30)$$

Next, finding the partial derivatives of LO on the k-th layer to e and c:

$$\xi_k^{(e)}\frac{\partial LO}{\partial e^{(k)}} = \begin{cases} -\left(b - e^{(k)}\right), & if \quad k = K \\ \dfrac{\partial LO}{\partial c^{(k+1)}}\dfrac{\partial c^{(k+1)}}{\partial e^{(k)}} = \left(\omega^{(k)}\right)^{\psi}\xi_{k+1}^{(c)}, & otherwise \end{cases} \qquad (31)$$

$$\xi_k^{(e)} = \frac{\partial LO}{\partial c^{(k)}} = \frac{\partial LO}{\partial e^{(k)}}\frac{\partial e^{(k)}}{\partial c^{(k)}} = \xi_k^e e^{(k)}\left(1 - e^{(k)}\right) \qquad (32)$$

Finding the partial derivatives of LO on the k-th layer to weight ω and bias r:

$$\nabla_{\omega^{(k)}} LO(\omega, r, a, b) = \frac{\partial}{\partial \omega^{(k)}} LO = \frac{\partial LO}{\partial c^{(k+1)}}\frac{\partial c^{(k+1)}}{\partial \omega^k} = \xi_{k+1}^{(c)}\left(e^{(k)}\right)^{\psi} \qquad (33)$$

$$\nabla_{r^{(k)}} LO(\omega, r, a, b) = \frac{\partial}{\partial r^{(k)}} LO = \frac{\partial LO}{\partial c^{(k+1)}}\frac{\partial c^{(k+1)}}{\partial r^{(l)}} = \xi_{k+1}^{(c)} \qquad (34)$$

Let χ be the learning rate of stochastic gradient descent. Then, the weight ω and bias r can be respectively updated by:

$$\omega' = \omega - \chi \nabla_{\omega^{(k)}} LO(\omega, r, a, b) \qquad (35)$$

$$r' = r - \chi \nabla_{r^{(k)}} LO(\omega, r, a, b) \qquad (36)$$

Fire image classification is a binary classification problem. Therefore, logistic classifier was adopted to classify fire images. Let $\omega^T a + r$ be the affine transform. Then, the classifier can be defined as:

$$g(a) = \frac{1}{1 + e^{-\left(\omega^T a + r\right)}} \qquad (37)$$

The Logistic classifier is a rotation symmetric function. Suppose $v = \omega^T a + r$. If the value of formula 37 is smaller than 0.5, then v must be negative; if the value is greater than 0.5, then v must be positive. During the classification of fire images, the probability of an image to be a fire image is [0.5, 1], and that of an image to be a non-fire image is [0, 0.5].

5. EXPERIMENTS AND RESULTS ANALYSIS

Flame contour extraction experiments were carried out on fire video frames with complex backgrounds, using the opensource database OpenCV. During the experiments, the fire video frames were judged preliminarily based on attributes like color, circularity, and eccentricity (Figure 5). On the left of Figure 5 is a screenshot of the original video, which contains interfering objects similar to fire in color (coal furnace and red cupboard). The suspected flame area is marked by blue curves. It can be seen from Figure 5 that the

proposed flame feature extraction algorithm can correctly extract the flame area and its contours. The extracted flame area is well connected, and rich in contour information. The subjective feature extraction effect is better than that of the existing algorithms.
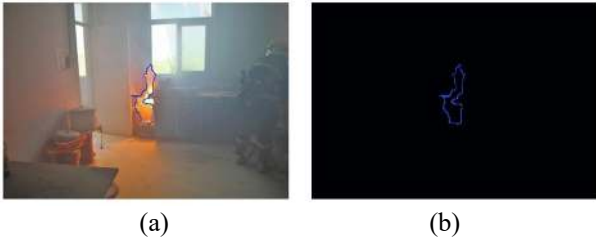

(a)   (b)

Figure 5. Flame contour extraction and recognition results
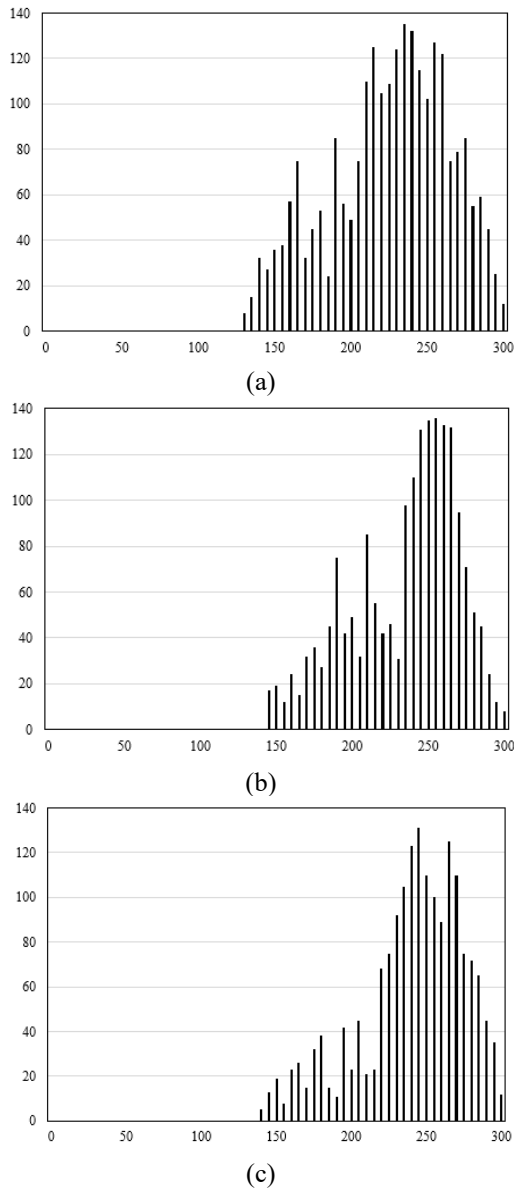

(a)

(b)

(c)

Figure 6. Statistics of I_R, I_G and I_B in smoke movement area

Figure 6 shows the statistics of I_R, I_G and I_B in smoke movement areas. Our previous analysis shows a small difference between I_R, I_G and I_B in the RGB color space of fire smoke image. The difference was greater than 4 and smaller than 25. The grayscales of the three components all belonged to the interval of [150, 210], and changed in the same direction

(increasing or decreasing). By extracting the color of smoke, the smoke movement area could be effectively identified in the fire image.

Table 1 lists the settings of the proposed CNN. It can be seen that the proposed neural network is less complex in computation than other commonly used CNNs. The computing load was 1/13 of that of lightweight MobileNet, and 1/356 of VGG-16. Under the same computer hardware, our CNN consumed less computing time than the other networks. The data processing effect could be ensured when it is applied to mobile terminals or embedded terminals.

Table 1. CNN settings

| Layer	Kernel size	Stride	Output size
Standard convolution 1	16×3×3×3	2	112×112×32
Depth wise separable convolution 1	16×1×3×3	1	56×56×32
Standard convolution 2	32×1×1×1	1	56×56×32
Depth wise separable convolution 2	32×1×3×3	2	28×28×32
Standard convolution 3	64×32×1×1	1	28×28×64
Depth wise separable convolution 3	64×1×3×3	2	14×14×64
Standard convolution 4	64×64×1×1	1	14×14×64
Depth wise separable convolution 4	64×1×3×3	2	7×7×64
Standard convolution 5	128×64×1×1	1	7×7×128
Depth wise separable convolution 5	128×1×3×3	1	7×7×128
Standard convolution 6	512×128×1×1	2	7×7×512
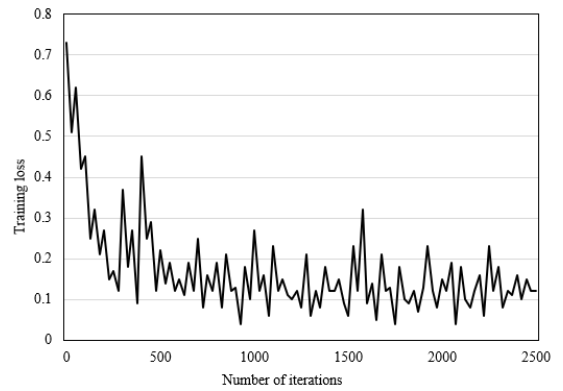
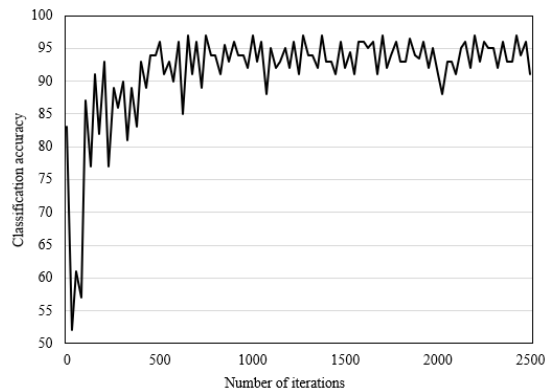**Figure 7.** Change curve of training loss

Figure 8. Classification accuracy through the test

This paper sets up a fire image library based on the screenshots of fire monitoring video. The library includes

1,400 positive samples (fire images), and 700 suspected negative samples (non-fire images). The 2,100 images were divided into a training set and a test set by the ratio of 5:1. The two sets were adopted to train and test our neural network, respectively. The training loss and test accuracy curves were plotted (Figures 7 and 8). With the growing number of iterations, the network loss gradually declined. After 1,000 iterations, the network tended to be stable and showed a converging trend. The fire classification accuracy increased gradually with the number of iterations. After network learning and parameter update, the classification accuracy stabilized at around 95%.

Figure 9. Fire recognition results

Table 2. Fire recognition probabilities and recognized fire states

Image number	Probability	Fire state	Image number	Probability	Fire state
a	0.752	Yes	d	0.653	Yes
b	0.645	Yes	e	0.786	Yes
c	0.751	Yes	f	0.934	Yes

Table 3. Partial test results

Video number	Type	SVM	Bayesian algorithm	Our algorithm
1	Smoke	0.725	0.816	0.851
2	Smoke	0.756	0.762	0.934
3	Open fire + smoke	0.742	0.804	0.916
4	Open fire + smoke	0.793	0.819	0.882
5	Open fire + smoke	0.755	0.823	0.935
7	Moving pedestrians	0.031	0	0
8	Floating balloons	0.237	0.834	0.075
9	Humidifier	0.623	0.726	0.143
10	Steam engine	0.532	0.543	0.057

Figure 9 shows the recognition and classification results of our neural network on fire video images in different scenes. On six fire images, our algorithm and model could successfully extract the features of the flame and smoke, and further realize fire recognition and positions. Table 2 lists the fire recognition probabilities and recognized fire states.

Finally, logistic classifier was adopted to classify the fires based on fire feature maps. Table 3 presents the classification results on images, including suspected negative samples. It can be seen that our method is much superior to the SVM and the Bayesian algorithm, as evidenced by its higher accuracy on the open fire and smoke in fire video images. The results confirm that our algorithm and model are advantageous in recognizing open fire and smoke amidst similar interfering objects.

6. CONCLUSIONS

This paper attempts to extract and classify image features for fire recognition based on the CNN. Firstly, the authors set up the framework of FVIFRS, and detailed the approach to extract flame features, both dynamic and static. Next, a Gaussian mixture model was established to extract the features from the fire smoke movement areas, thereby enhancing the efficiency of image analysis. Finally, the depth wise separable CNN was adopted to process and classify fire feature maps. The proposed algorithm and model were proved to be feasible and effective through experiments. Specifically, the flame contour extraction and recognition results demonstrate that our algorithm is better than other algorithms in feature extraction. According to the statistics on the I_R, I_G and I_B components in the smoke areas of fire images, the effective extraction of smoke color can lead to the effective recognition of the smoke areas. Further, the training loss and test accuracy curves were plotted for our network, and the recognition and classification results were obtained in different scenes. The results confirm the large superiority of our neural network in recognizing open fire and smoke amidst similar interfering objects.

REFERENCES

[1] Ya'acob, N., Najib, M.S.M., Tajudin, N., Yusof, A.L., Kassim, M. (2021). Image processing based forest fire detection using infrared camera. In Journal of Physics: Conference Series, 1768(1): 012014.

[2] Furtado, H., Gendrin, C., Spoerk, J., Steiner, E., Underwood, T., Kuenzler, T., Birkfellner, W. (2016). FIRE: an open-software suite for real-time 2D/3D image registration for image guided radiotherapy research. In Medical Imaging 2016: Image Processing, International Society for Optics and Photonics, 9784: 978449. https://doi.org/10.1117/12.2216082

[3] Toulouse, T., Rossi, L., Celik, T., Akhloufi, M. (2016). Automatic fire pixel detection using image processing: a comparative analysis of rule-based and machine learning-based methods. Signal, Image and Video Processing, 10(4): 647-654. https://doi.org/10.1007/s11760-015-0789-x

[4] Nemalidinne, S.M., Gupta, D. (2018). Nonsubsampled contourlet domain visible and infrared image fusion framework for fire detection using pulse coupled neural network and spatial fuzzy clustering. Fire Safety Journal, 101: 84-101. https://doi.org/10.1016/j.firesaf.2018.08.012

[5] Madhevan, B., Ramanathan, S., Jha, D.K. (2017). A

novel image intelligent system architecture for fire proof robot. In Artificial Intelligence and Evolutionary Computations in Engineering Systems, 805-817. https://doi.org/10.1007/978-981-10-3174-8_67

[6] Jun, J.H., Kim, M.J., Jang, Y.S., Kim, S.H. (2017). Fire detection using multi-channel information and gray level co-occurrence matrix image features. Journal of Information Processing Systems, 13(3): 590-598. https://doi.org/10.3745/JIPS.02.0062

[7] Hossain, F.A., Zhang, Y.M., Tonima, M.A. (2020). Forest fire flame and smoke detection from UAV-captured images using fire-specific color features and multi-color space local binary pattern. Journal of Unmanned Vehicle Systems, 8(4): 285-309. https://doi.org/10.1139/juvs-2020-0009

[8] Bianco, V., Mazzeo, P.L., Paturzo, M., Distante, C., Ferraro, P. (2020). Deep learning assisted portable IR active imaging sensor spots and identifies live humans through fire. Optics and Lasers in Engineering, 124: 105818. https://doi.org/10.1016/j.optlaseng.2019.105818

[9] Luo, G. (2017). 3. A Method of Forest-Fire Image Recognition Based on Collaborative Filtering Algorithm. Boletín Técnico, 55(9).

[10] Gwak, D.G., Kim, D.H. (2016). Image matching algorithm for thermal panorama image construction adaptable for fire disasters. Journal of Institute of Control, Robotics and Systems, 22(11): 895-903.

[11] Du, Y., Fan, Y. (2020). Distributed image fire detection and alarm system based on machine learning. In International Conference on Application of Intelligent Systems in Multi-modal Information Analytics, pp. 362-371. https://doi.org/10.1007/978-3-030-51556-0_53

[12] Mahmoud, M.A., Ren, H. (2018). Forest fire detection using a rule-based image processing algorithm and temporal variation. Mathematical Problems in Engineering. https://doi.org/10.1155/2018/7612487

[13] Han, B., Wu, Y.Q., Kang, N.I. (2018). Segmentation of early fire image of mine using improved cv model. Journal of China University of Mining & Technology, 47(2): 429-435.

[14] Li, P., Yang, Y., Zhao, W., Zhang, M. (2021). Evaluation of image fire detection algorithms based on image complexity. Fire Safety Journal, 121: 103306. https://doi.org/10.1016/j.firesaf.2021.103306

[15] Doutsi, E., Fillatre, L., Antonini, M., Tsakalides, P. (2021). Dynamic image quantization using Leaky Integrate-and-Fire neurons. IEEE Transactions on Image Processing, 30: 4305-4315. https://doi.org/10.1109/TIP.2021.3070193

[16] Dutta, S., Ghosh, S. (2021). Forest fire detection using combined architecture of separable convolution and image processing. In 2021 1st International Conference on Artificial Intelligence and Data Analytics (CAIDA),

pp. 36-41. https://doi.org/10.1109/CAIDA51941.2021.9425170

[17] Kim, H., Kim, G. (2020). Single image super-resolution using fire modules with asymmetric configuration. IEEE Signal Processing Letters, 27: 516-519. https://doi.org/10.1109/LSP.2020.2980172

[18] Singh, L., Kumar, A., Priyadarshi, R. (2018). Performance and comparison analysis of image processing based forest fire detection. In International Conference on Nanoelectronics, Circuits and Communication Systems, pp. 473-479.

[19] Sayyed, A.S., Hasan, M.T., Mahmood, S., Hossain, A.R. (2019). Autonomous fire fighter robot based on image processing. In 2019 IEEE Region 10 Symposium (TENSYMP), pp. 503-507. https://doi.org/10.1109/TENSYMP46218.2019.8971157

[20] Mahmoud, M.A.I., Ren, H. (2019). Forest fire detection and identification using image processing and SVM. Journal of Information Processing Systems, 15(1): 159-168. https://doi.org/10.3745/JIPS.010038

[21] Zarkasi, A., Nurmaini, S., Stiawan, D., Amanda, C.D. (2019). Implementation of fire image processing for land fire detection using color filtering method. In Journal of Physics: Conference Series, 1196(1): 012003.

[22] Sadewa, R.P., Irawan, B., Setianingsih, C. (2019). Fire detection using image processing techniques with convolutional neural networks. In 2019 International Seminar on Research of Information Technology and Intelligent Systems (ISRITI), pp. 290-295. https://doi.org/10.1109/ISRITI48646.2019.9034642

[23] Windle, C.I., Anderson, J., Boyd, J., Homan, B., Korivi, V., Ma, L. (2021). In situ imaging of 4D fire events in a ground vehicle testbed using customized fiber-based endoscopes. Combustion and Flame, 224: 225-232. https://doi.org/10.1016/j.combustflame.2020.11.022

[24] Chen, S., Su, C., Kuang, Z., Ouyang, Y., Gong, X. (2021). Insulator fault detection in aerial images based on the mixed-grouped fire single-shot multibox detector. Journal of Imaging Science and Technology. https://doi.org/10.2352/J.ImagingSci.Technol.2021.65.3.030402

[25] Jeong, S.Y., Kim, W.H. (2020). Thermal imaging fire detection algorithm with minimal false detection. KSII Transactions on Internet and Information Systems (TIIS), 14(5): 2156-2170. https://doi.org/10.3837/tiis.2020.05.016

[26] Sharma, A., Singh, P.K., Kumar, Y. (2020). An integrated fire detection system using IoT and image processing technique for smart cities. Sustainable Cities and Society, 61: 102332. https://doi.org/10.1016/j.scs.2020.102332

[27] Pranamurti, H., Murti, A., Setianingsih, C. (2019). Fire detection use CCTV with image processing based Raspberry PI. In Journal of Physics: Conference Series, 1201(1): 012015.