IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2020.Doi Number

Efficient Video Fire Detection Exploiting Motion-Flicker-based Dynamic Features and Deep Static Features (April 2020)

YAKUN XIE, JUN ZHU, YUNGANG CAO, YUNHAO ZHANG, DEJUN FENG, YUCHUN ZHANG, AND MIN CHEN

Faculty of Geosciences and Environmental Engineering, Southwest Jiaotong University, Chengdu 611756, China

Corresponding authors: JUN ZHU (e-mail: zhujun@swjtu.edu.cn) and YUNGANG CAO (e-mail: yungang@swjtu.cn)

This work was supported by the National Natural Science Foundation of China (Grant Nos. 41871289 and 41771451), and the Sichuan Youth Science and Technology Innovation Team (20CXTD0102).

ABSTRACT Since fire is one of the most serious types of accidents that can occur, there is always a need for improvement in fire detection capabilities. Convolutional neural networks (CNNs) have been used for a variety of high-performance computer vision tasks. The use of CNNs to extract deep static features of fire has greatly improved the accuracy of fire detection. However, the implementation of CNNs in the real world is limited by their high computational cost. In addition, fire detection methods based on the classification of images alone using CNNs cannot account for the dynamic features of fire. Therefore, in this paper, a method that exploits both motion-flicker-based dynamic features and deep static features is proposed for video fire detection. First, dynamic features are extracted by analyzing the differences in motion and flicker features between fire and other objects in videos. Second, an adaptive lightweight convolutional neural network (AL-CNN) is proposed to extract the deep static features of fire. Finally, the dynamic and static features of fire are combined to establish a video fire detection method with improved operational efficiency in terms of accuracy and run time. To prove the validity of our method, its accuracy and run time are evaluated on three test datasets, and the results reveal that our method exhibits better performance than state-of-the-art methods. Moreover, our method is shown to be feasible in complex video scenarios and for devices with resource constraints.

INDEX TERMS Fire detection, motion-flicker-based dynamic features, deep static features, background subtraction, flicker detection, adaptive lightweight convolutional neural network.

I. INTRODUCTION

Fire is one of the most dangerous types of disasters, threatening human life and property, the ecological environment, and infrastructure. Reducing the damage caused by fire has important theoretical and practical significance [1], [2]. With the increasing popularity of video surveillance equipment and the development of computer vision techniques, video fire detection methods based on fire features have attracted widespread attention from researchers [3]-[5].

The features of fire can be divided into static features and dynamic features. Static features include spectral information and spatial structure information, such as brightness, color, texture, and edges. Dynamic features include the overall motion features and random motion features, such as motion and flicker features [6]. Early methods of fire detection usually identify fire on the basis of one or more of these features, such as methods based on the construction of color models using static features, including RGB, HIS, and YCbCr features [7]-[9]. In addition, methods involving the combination of multiple color models have been applied for fire detection. Zaidi et al. performed video fire detection based on RGB and YCbCr features by setting thresholds [10]. However, color-based fire detection methods are often susceptible to a variety of environmental factors, such as sunlight, other light sources, and red or orange objects, which can lead to high false alarm rates.

To overcome this susceptibility, researchers have conducted further investigations by combining color features, shape features, dynamic features, etc. Seebamrungsat et al. performed fire detection by combining multiple feature rules,

considering the HSV, YCbCr, and interframe features of fire [11]. Lascio et al. combined color and motion features in an expert system for fire detection based on the analysis of surveillance videos [12]. Marbach et al. analyzed video frame sequences. The features of video sequences were extracted and used to determine whether a fire had occurred [13]. Chen et al. used a background detection method to obtain the moving areas associated with fire and smoke in videos and then determined the color features of these moving areas to identify the presence of a fire [8]. Foggia et al. used an expert system to build a rule set based on fire color, shape, and motion features, which offered improved accuracy but also suffered from a high false alarm rate [6]. Yan et al. extracted multiple features for forest fire recognition, including color, texture, area, and shape features [14]. Kosmas et al. built an SVM classifier for fire detection based on motion features, texture features, flicker features, and color probability features [15]. Toreyin et al. carried out a series of studies on fire discrimination and successfully used a hidden Markov model to realize the real-time detection of fire in videos [16]. The researchers whose methods are reviewed above built their own extractors to improve the accuracy of fire detection. Such "hand-crafted" dynamic features, for example, motion and flicker features, have promoted the development of video fire detection. However, motion detection or flicker frequency analysis alone is insufficient to effectively extract dynamic features. In addition, because of the high complexity of fire scenes in videos, artificially designed static features are highly redundant. The intelligent extraction of as many deep static features as possible is impossible. However, a deep neural network can effectively extract the deep static features of an image through automatic learning, which can help to improve performance.

Hinton et al. proposed the theory of deep learning in 2006 [17]. Deep learning involves extracting high-level abstract features of data through nonlinear expressions and building mathematical models to achieve improved classification and detection accuracy; hence, it has become a popular area of research in the artificial intelligence community. In recent years, a large number of neural network models have been proposed, such as convolutional neural networks (CNNs) [18], recurrent neural networks (RNNs) [19], and deep belief networks (DBNs) [20]. These networks have been used for a variety of high-performance computer vision tasks, such as image processing [18], [21], object detection [22], natural language processing [23], speech recognition and other applications [24]-[27]. Among them, CNNs have achieved superior results in image classification.

More recently, many methods using neural network algorithms to extract the static features of fire have been applied for fire detection. Frizzi et al. proposed a CNN-based method for fire and smoke detection and tested it on video sequences [28]. Sharma et al. used higher-performing network models, i.e., VGG16 and ResNet50, for fire detection [29]. Shen et al. used the popular YOLO network framework for fire detection and compared the results with those of shallow learning methods to prove the effectiveness of deep learning [30]. Hu et al. proposed a long-period neural network model and an optical flow method for the real-time detection of fire and smoke [31]. Zhang et al. jointly trained a CNN on complete images and image blocks for the detection and localization of fire in an image [32]. Muhammad et al. proposed a CNN-based early fire recognition method for early real-time fire detection in surveillance videos and established a more efficient CNN-based fire detection framework based on SqueezeNet [33]. In addition, Muhammad et al. conducted further research and established a fire detection framework combined with 5G network transmission to achieve fire detection in uncertain environments [34].

Although these works using CNNs are notable, they do not take advantage of the dynamic features of fire. In addition, CNN models face challenges in terms of popularization because of their high memory consumption. Furthermore, the accuracy of fire detection still requires improvement due to its critical importance for disaster management. Moreover, achieving high robustness of deep learning models for video fire detection in complex video scenarios remains challenging. The main contributions of our work are summarized below.

1) An efficient video fire detection method is proposed that exploits both motion-flicker-based dynamic features and deep static features to achieve improved performance in terms of its accuracy and false alarm rate. In addition, experiments prove that our method can be applied to a variety of complex video scenarios.
2) Our method considers both motion and flicker features, which is helpful for more effectively extracting dynamic features while reducing time consumption.
3) Our method uses an adaptive lightweight CNN to extract the deep static features of fire, which can reduce the computational burden while avoiding the loss of image features caused by fixed-size image input.

The remainder of this article is structured as follows. The proposed method is presented in Section II, including a detailed introduction to the dynamic and static feature acquisition methods. In Section III, the hyperparameter settings and experimental dataset, the evaluation metrics, and the experimental results are described in detail. The results of this paper are discussed in Section IV. Finally, the conclusion and plans for future work are presented in Section V.

II. THE PROPOSED METHOD

A. ACQUISITION OF DYNAMIC FEATURES

As a nonrigid moving object, a fire has obvious dynamic features in a video [35]. To make full use of these dynamic features, a motion-flicker-based algorithm that considers both motion and flicker features is proposed for the acquisition of dynamic features, inspired by the work of Chen et al. [36]. This algorithm includes background subtraction and flicker detection. First, background subtraction is applied to extract

motion features, which are often used for the extraction of moving areas in videos [37]. Research has shown that the KNN-based approach offers desirable performance in outdoor scenarios [38], [39]. Similarly, this approach is suitable for the background subtraction of fire. Video stream processing and background subtraction are implemented through OpenCV, which is an open-source algorithm library [40]. A moving area in a video extracted through background detection is called a suspected region of interest in our method. The second step is flicker detection. A fire will produce a disordered continuous high-frequency time series of changes relative to ordinary objects due to the combustion process [41], [42]. These changes manifest as flicker or pulsations, which are called the flicker features of fire. This step can determine whether a suspected region of interest exhibits flicker features; if so, the moving area is considered to have the dynamic features of fire and is called a region of interest. The overall algorithm is explained as follows.

1) Obtain the coordinate position *(x, y, w, h)* of each suspected region of interest in the current video frame based on background subtraction, where *(x, y)* represents the coordinates of the upper-left corner of the suspected region of interest and *(w, h)* represents the width and height.

2) Create a pixel frequency matrix *SUM* of the same size as each suspected region of interest, which will be used to analyze the brightness changes of each pixel, with coordinates *(x, y)*, in the moving area. The brightness calculation method is shown in equation (1). The equal-weighted average of the three channels is used to avoid floating-point calculations to reduce the number of calculations required [36].

$$I_t(x,\ y) = \frac{1}{3}\left[R_t(x,\ y) + G_t(x,\ y) + B_t(x,\ y)\right] \quad (1)$$

where I_t represents the pixel brightness value at time t; R_t, G_t, and B_t represent the pixel value in each band at time t; and *(x, y)* represents the coordinates of the pixel in the image.

3) If the brightness value of the pixel at *(x, y)* changes between time t and time t-1, the value of the corresponding element in the frequency matrix, $SUM_t(x, y)$, is increased by 1, whereas otherwise, it is increased by 0, as shown in equation (2).

$$SUM_t = \begin{cases} SUM_{t-1}(x,\ y) + 1 & if(\Delta I(x,\ y) \geq T_I) \\ SUM_{t-1}(x,\ y) + 0 & if(\Delta I(x,\ y) < T_I) \end{cases} \quad (2)$$

where

$$\Delta I(x,\ y) = \left| I_t(x,\ y) - I_{t-1}(x,\ y) \right| \quad (3)$$

where $\Delta I(x, y)$ represents the change in brightness at *(x, y)* between time t and time t-1 and T_I is a positive real number that represents the global change threshold.

4) If the oscillation count for a pixel within a certain time exceeds a set threshold, that pixel is considered to have a fire flicker feature, as shown in equation (4).

$$\left| SUM_t(x,\ y) - SUM_{t-n}(x,\ y) \right| \geq SUM_T \quad (4)$$

where n is the specified counting period, the length of which is set to 3, and the interval between counting periods is set to 1. SUM_T is the dynamic flicker threshold. With these settings, if there is at least one above-threshold brightness difference between three consecutive frames of video at the same pixel coordinates, this pixel is considered to have a flicker feature.

5) The final regions of interest are determined on the basis of a threshold λ, as shown in equation (5).

$$T_f \ / \ T_{rect} \geq \lambda \quad (5)$$

where T_f is the number of pixels satisfying equation (4) in the candidate fire region and T_{rect} is the total number of pixels in the candidate fire region. λ is an experimental threshold. Finally, any area that satisfies equation (5) is identified as a region of interest.

B. ACQUISITION OF DEEP STATIC FEATURES

To extract the deep static features of fire, we propose an adaptive lightweight convolutional neural network (AL-CNN), as shown in Fig. 1. The core of the lightweight network is a deep separable convolution structure, which realizes the separate mapping of channels and regions and reduces the required number of parameters and memory consumption. The AL-CNN consists of three parts: a network initialization stage, an inverted residual block stage and a spatial pyramid pooling stage.

The first part is the network initialization stage, as shown in Fig. 1 (A), which consists of three modules: a convolutional layer, a batch normalization (BN) layer and a hard version of the swish (h-swish) activation function. BN is a neural network training optimization method proposed by Google [43]. It has been widely used in neural networks to accelerate network convergence and improve the stability of training. The h-swish activation function draws on the latest achievements of MobileNetV3; it offers increased accuracy while ensuring a low computational cost [44], [45]. During training, the network initialization stage can enhance the ability of the network to learn sparse features and improve the robustness of the extraction of deep static features.

The second part is the inverted residual block stage, which is inspired by MobileNetV2 [46] and consists of two types of components: inverted residual blocks and downsampling blocks. Each inverted residual block consists of three steps, as shown in Fig. 1 (B1). First, the dimensionality is expanded by a 1 × 1 convolution, as a deep convolution itself does not have the ability to change the number of channels. Then, image features are extracted through depthwise separable convolutions. Finally, multiple features are obtained through shortcut connections. The structure of a downsampling block is shown in Fig. 1 (B2). The purpose of downsampling is achieved by setting the stride to 2. The inverted residual block and the downsampling block have the same structure, except for the shortcut connection and the stride. The inverted residual block stage enables us to extract the deep static features of fire.

The third part is the spatial pyramid pooling (SPP) stage, as shown in the red dotted box in Fig. 1. SPP was proposed by He et al. in 2015 [47]. It can enable a CNN to process images of any scale while avoiding the loss of static features caused by cropping and warping operations; this is why we call our network an adaptive network. In addition, the maximum pooling function is used to suppress local noise and improve the accuracy of target recognition. Adding the SPP structure at the end of the network avoids the need for fixed-size image input and improves the ability of the network to detect fire.

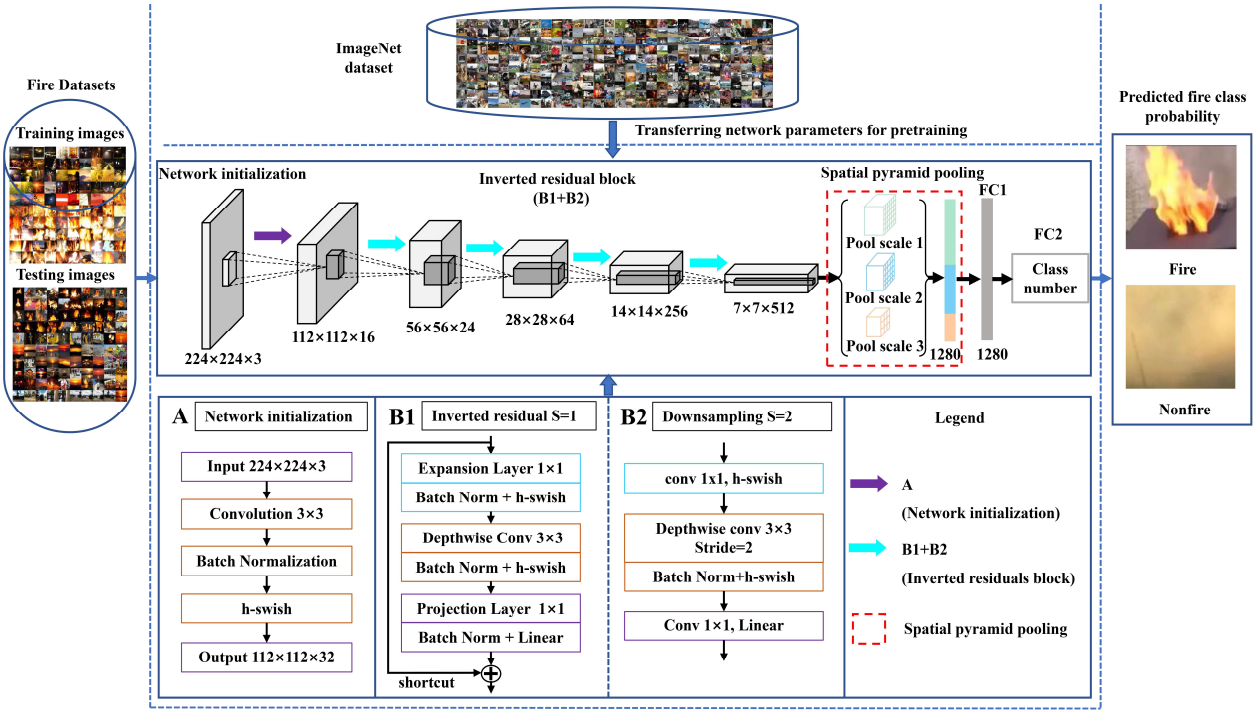

FIGURE 1. Deep static feature extraction framework based on an AL-CNN.

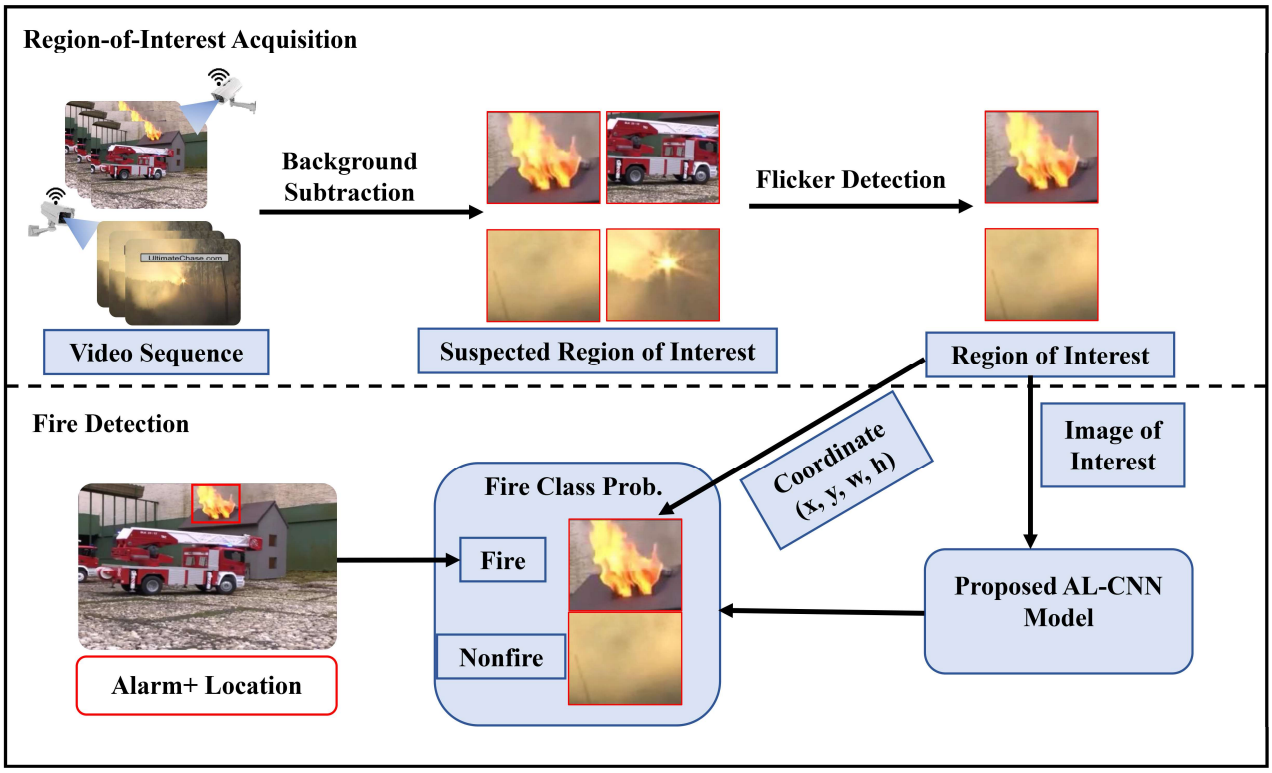

FIGURE 2. The proposed framework for video fire detection exploiting motion-flicker-based dynamic features and deep static features.

C. VIDEO FIRE DETECTION EXPLOITING MOTION-FLICKER-BASED DYNAMIC FEATURES AND DEEP STATIC FEATURES

To improve the accuracy and efficiency of video fire detection, a method that exploits motion-flicker-based dynamic features and deep static features is proposed, as shown in Fig. 2. The proposed framework is divided into two main phases. First, region-of-interest acquisition is carried out based on the dynamic features. This process involves background subtraction and flicker detection. Background subtraction helps to obtain the suspected regions of interest, while flicker detection helps to obtain the regions of interest. In this phase, the images of interest are extracted, and the coordinates of interest in the video frames are recorded. Second, fire detection is carried out based on the static features. This phase involves extracting the deep static features of fire using the AL-CNN, which can fully extract these static features by means of inexpensive computations while avoiding the loss of image features due to fixed-size image input. The AL-CNN is used to identify whether each region of interest identified in the first phase is, in fact, a fire region; if so, an alarm is generated, and the fire coordinates in the video frame are output.

III. EXPERIMENTS AND RESULTS

A. HYPERPARAMETER SETTINGS AND DATASET DESCRIPTIONS

1) HYPERPARAMETER SETTINGS

All training and testing are implemented using TensorFlow and Keras on the Windows 10 platform with an Nvidia GeForce GTX 1060 6 GB graphics card. The values of the hyperparameters are shown in Table 1.

TABLE 1. Values of Hyperparameters Used in the Experiments.

Epochs	Batch size	Learning rate
200	80	0.001

By monitoring the value of the loss function, the learning rate was reduced by 0.9 after every 5 consecutive epochs in which the performance did not improve. In addition, the transfer learning strategy was applied during the training process. First, the AL-CNN was pretrained on the 1000 classes of the ImageNet Dataset to determine the initial weights. Then, to classify fire and nonfire regions, the number of neurons in the last layer of our network was changed from 1000 to 2.

2) DATASET DESCRIPTIONS

Many fire detection datasets have been provided by researchers. Chino et al. provided an image dataset including 119 fire images and 107 nonfire images [48]. Foggia et al. provided a fire video dataset consisting of 31 videos captured by a camera in different scenes, with a total duration of more than 1 hour [6]. In addition, Dimitropoulos et al., Byoung et al., Hüttner et al., and Chenebert et al. have also provided datasets for fire detection [15], [49]-[51]. Although the existing datasets are large, the training datasets mostly consist of video frame images, leading to a large number of repeated images. Considering the features of only a single fire type will result in a feature representation that is insufficiently discriminative. By contrast, the combustion of different substances will produce fires with different representative color features, as shown in Fig. 3. Therefore, to improve the robustness of the neural network model, the fire training dataset was refined by adding different categories of fire images. The final training dataset included 22586 images, of which 9332 were fire images and 13254 were nonfire images. A detailed description of the training and testing datasets is shown in Table 2. Note that the images used during training and testing do not overlap.

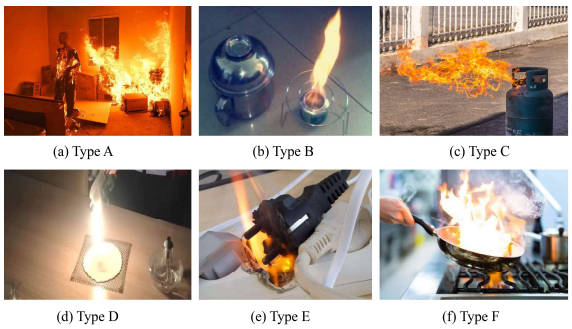

FIGURE 3. Fires with different features observed when different substances are burning. (a) Fire from the burning of solid fuel. (b) Fire from the burning of liquid or liquefiable fuel. (c) Fire from the burning of gaseous fuel. (d) Fire from the burning of combustible metals. (e) Any type A or type B fire that occurs next to an electrical appliance, wire, or living object. (f) Fire from the burning of fat and oil used for cooking.

TABLE 2. Statistics of the Training and Test Datasets.

Dataset	Dataset source	Total images	Fire	Nonfire
Training	-	22586	9332	13254
Test	DS1 (image) [48]	226	119	107
	DS2 (video) [6]	50151	8960	41191
	DS3 (video)	79072	31837	47235

B. EVALUATION METRICS

To quantitatively evaluate the performance of our proposed method and compare it with the results of other researchers, the false positive rate (also referred to as the false alarm rate) (equation (6)), false negative rate (equation (7)) and accuracy (equation (8)) are used as evaluation metrics in this paper [52]. The goals in this paper are to achieve a high accuracy, a low false positive rate and a low false negative rate. In addition, the run time necessary for detection is evaluated in terms of the frame rate (fps), which is the average number of video frames that can be processed per second.

$$False\ positive\ rate = \frac{FP}{FP+TN} \tag{6}$$

$$False\ negative\ rate = \frac{FN}{FN+TP} \tag{7}$$

$$Accuracy = \frac{TP+TN}{TP+FN+FP+TN} \tag{8}$$

TP, FN, FP and TN represent the fire detection results in comparison with the ground truth.

TP: the number of true positives, i.e., the number of correctly detected fire regions.
FN: the number of false negatives, i.e., the number of misclassified fire regions.
FP: the number of false positives, i.e., the number of erroneously detected fire regions.
TN: the number of true negatives, i.e., the number of correctly detected nonfire regions.

C. EXPERIMENTS ON DYNAMIC AND STATIC FEATURE EXTRACTION

1) RESULTS OF DYNAMIC FEATURE EXTRACTION

To prove the performance of the dynamic feature extraction method, several videos in the test dataset were used for experiments. Fig. 4 (a) shows the original video frames, including images of a fire and other moving objects. Fig. 4 (b) shows the suspected regions of interest obtained after background subtraction, which include many sources of interference, such as pedestrians, car lights, and sunlight. Fig. 4 (c) shows the resulting regions of interest obtained after background subtraction and flicker detection. As shown, most of the fire-like sources of interference are eliminated in these results. However, this extraction method is based on hand-crafted features, and some items may be missed during detection in complex video scenarios, such as the moving fog in the last row of Fig. 4 (c). These sources of interference are avoided through the use of the deep neural network in the next step.

(a) (b) (c)

FIGURE 4. Results of dynamic feature extraction. (a) The original video frames. (b) The resulting suspected regions of interest obtained via background subtraction. (c) The resulting regions of interest obtained via background subtraction and flicker detection.

The run time needed for the acquisition of dynamic features is also considered in this paper. Background subtraction can eliminate static frames and avoid the need for time-consuming flicker detection over an entire image. In Table 3, we compare the run times for dynamic feature acquisition using our method (background subtraction and flicker detection) and using only flicker detection without background subtraction. Our method achieves a frame rate of 67 fps, whereas flicker detection without background subtraction has a frame rate of 43 fps, thus proving that the strategy for dynamic feature acquisition presented in this paper can result in a faster run time.

TABLE 3. Run-time comparison of dynamic feature acquisition using different strategies.

Method	Frame rate (fps)
Our method	67
Flicker detection without background subtraction	43

2) RESULTS OF DEEP STATIC FEATURE EXTRACTION

To verify the performance of static feature extraction based on the AL-CNN, a small image dataset (DS1) was used to test the model in a separate experiment. In addition, several excellent existing lightweight networks were selected for comparison, namely, SqueezeNet, ShuffleNet, ShuffleNetV2, MobileNet, and MobileNetV2 [53]-[56], [46]. The results are shown in Table 4. Compared with the other methods, our method achieves a false positive rate that is lower by 0.94-9.21%, a false negative rate that is lower by 2.6-6.73%, and an accuracy rate that is higher by 1.83-7.48%. In addition, the average time needed to process an image using our method is 0.014 s. Nevertheless, although the overall performance of our method is better than that of the existing lightweight network methods, its accuracy is still limited. The false positive and false negative rates are still undesirably high, reaching 14.15% and 6.72%, respectively. Similarly, although the accuracy is also improved, it is only 89.78%.

TABLE 4. Comparison of our method with other lightweight CNNs on DS1.

Method	False positive rate (%)	False negative rate (%)	Accuracy (%)
Our method	14.15	6.72	89.78
SqueezeNet	19.63	13.45	83.63
ShuffleNet	21.50	10.92	84.07
ShuffleNetV2	16.82	9.24	87.17
MobileNet	23.36	12.61	82.30
MobileNetV2	15.09	9.32	87.95

The test results were further analyzed to identify the specific shortcomings of our method. Example of correct detection results are shown in Fig. 5 (a). Examples of misclassification are shown in Fig. 5 (b); these cases mostly correspond to small fires at long distances or with occlusions. Fig. 5 (c) shows examples of erroneous detection, which is mostly caused by fire-like sunlight or other light sources. The region-of-interest acquisition process can be used to identify small moving objects and recognize whether such moving objects exhibit flicker features, which helps our method to avoid misclassification and erroneous detection.

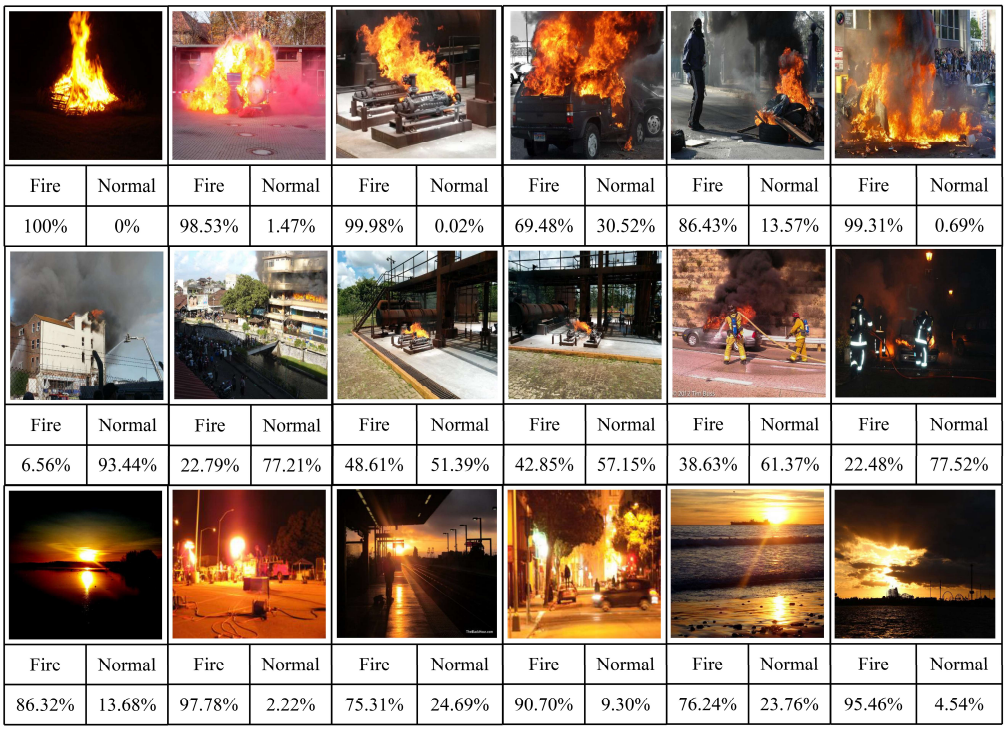

FIGURE 5. Fire detection results of the AL-CNN. (a) Correct detection of fire. (b) Misclassification of fire. (c) Erroneous detection of fire.

D. EXPERIMENTS ON VIDEO DATASET 2

DS2 was provided by Foggia et al. and includes 31 videos captured by a camera in different scenes, of which 14 videos contain fire and the remaining 17 videos are ordinary videos. As done by Foggia et al., 80% of the videos were used for testing in these experiments [6]. Examples of images extracted from DS2 are shown in Fig. 6. Videos 1-3 show small fires from a long distance, videos 4 and 5 contain fire-like objects, and videos 6-8 contain forest fires, thus constituting a good test of our method's robustness. In addition, the dataset contains a large number of nonfire interference videos. Videos 9 and 10 contain red fire-like objects, videos 11 and 12 contain sunlight and fire-like objects, videos 13 and 14 show common outdoor scenes that contain interference from fire-like objects and smoke, and videos 15 and 16 contain videos recorded in the mountains with moving fog. Hence, this dataset is challenging for both color-based and motion-based fire detection methods. In addition, this dataset is often used in fire detection studies, which makes it easier to compare the method proposed in this paper with other existing methods.

We selected 8 related algorithms that yield excellent results for comparison with our method, including methods based on both deep learning and hand-crafted features for fire detection. The accuracy comparison results are shown in Table 5. Among the compared methods, the algorithms of Foggia, Lascio, and Celik have the best false negative rates, but among them, the highest false positive rate is 29.41%, and the lowest is 6.67%. These algorithms have high false positive rates, and their accuracy is not optimal. From the perspective of the false positive rate, the best result is achieved by the method of Habioğglu, i.e., 5.88%, but the corresponding false negative rate is 14.29%, which is the worst among all of the detection methods. The method proposed by Muhammad et al. has the best accuracy, but its false positive rate is 8.87%, and its false negative rate is also still somewhat high. By contrast, the false positive rate of our method is 2.33%, representing a reduction of 3.55%-38.85% relative to the other methods. The accuracy of our method is 97.94%, corresponding to an increase of 3.44%-23.74%, and the false negative rate is 0.84%. A further analysis of the experimental results showed that the false negatives correspond to frames in which the fires are about to burn out, whereas the fires can be detected effectively in the early stage. In summary, our method shows the best performance.

TABLE 5. Comparison with different fire detection methods on DS2.

Method	False positive rate (%)	False negative rate (%)	Accuracy (%)
Our method	2.33	0.84	97.94
Muhammad et al. [57]	8.87	2.12	94.50
Foggia et al. [6]	11.76	0	93.55
Lascio et al. [12]	6.67	0	92.59
Habiboğlu et al. [60]	5.88	14.29	90.32
Rafiee et al. (RGB) [7]	41.18	7.14	74.20
Rafiee et al. (YUV) [7]	17.65	7.14	87.10
Celik et al. [9]	29.41	0	83.87
Chen et al. [8]	11.76	14.29	87.10

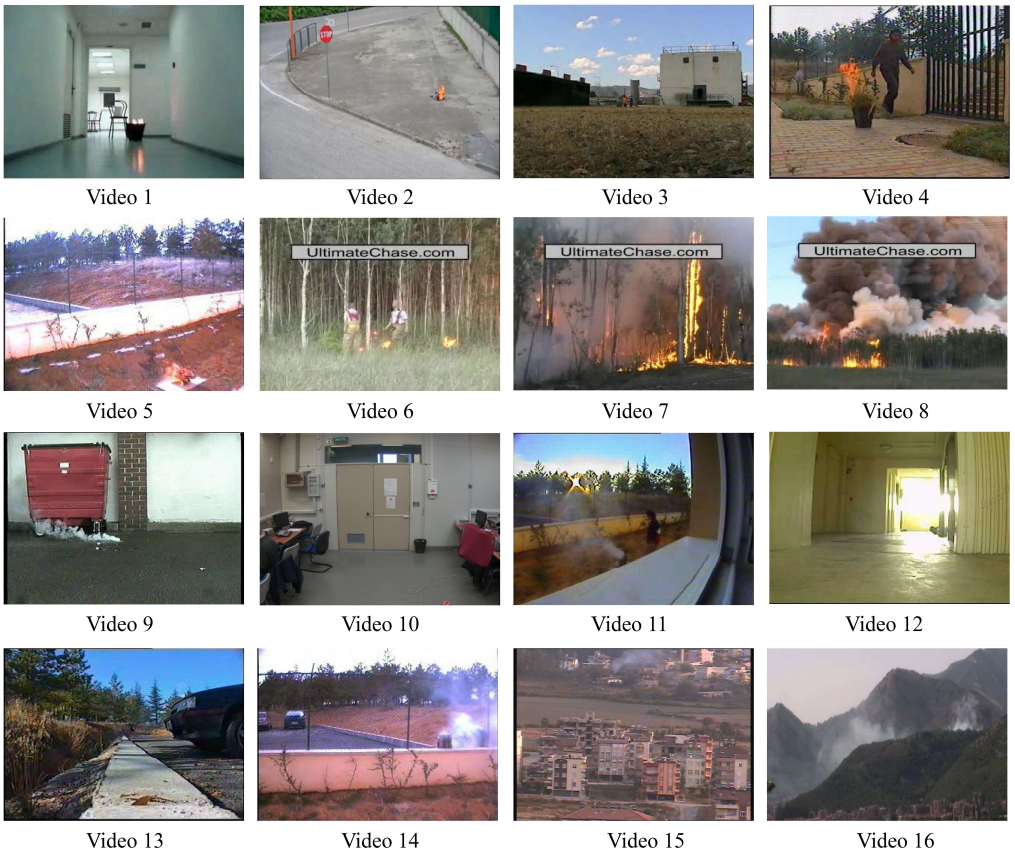

FIGURE 6. Examples of images extracted from DS2. The top two rows show frames extracted from fire videos, while the bottom two rows show frames extracted from normal videos.

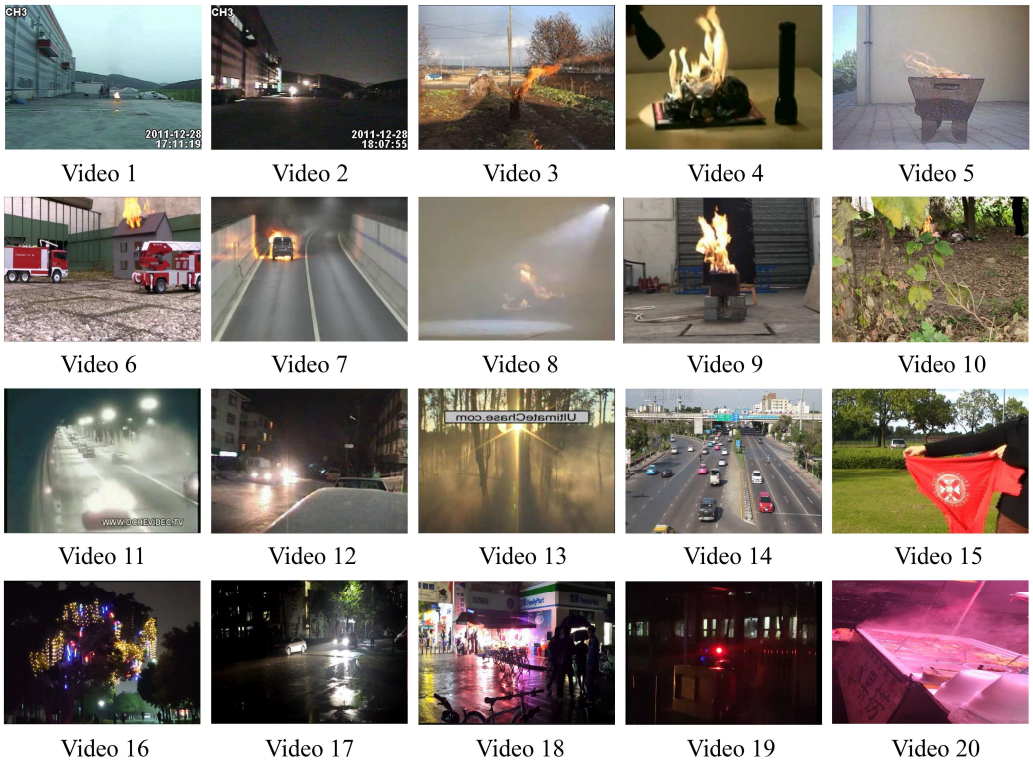

FIGURE 7. Sample video frames from DS3 that were used to test the proposed method.

E. EXPERIMENTS ON VIDEO DATASET 3

We built a new dataset from complex video scenarios to further demonstrate the performance of our method. Sample frames from the videos in DS3 are shown in Fig. 7. The dataset includes 20 videos: 10 fire videos (videos 1-10) and 10 nonfire videos (videos 11-20). The total time of the videos in DS3 is approximately 43 minutes. In addition, videos 8-10 and 16-20 were obtained experimentally, whereas the other videos were found on the Internet.

The example frames show that DS3 contains many challenges, such as fires viewed from far away, an occluded fire, a tunnel fire, and fires obscured by smoke and light. The nonfire videos include complex video scenarios with many types of interferences, such as artificial lights, sunlight, red objects, and bad weather. A detailed explanation of each video in DS3 is given in Table 6.

TABLE 6. Detailed descriptions of the videos in DS3.

Video name	Modality	Notes
Video 1	Fire	Fire at a 30 m distance in daytime
Video 2	Fire	Fire at a 30 m distance at night
Video 3	Fire	Fire in the wild
Video 4	Fire	Fire under a light
Video 5	Fire	Fire outside in wind
Video 6	Fire	Fire and fire truck
Video 7	Fire	Fire in a tunnel
Video 8	Fire	Fire in fog and under a light
Video 9	Fire	Fire in a building
Video 10	Fire	Fire in the wild with occlusion
Video 11	Normal	Tunnel lights
Video 12	Normal	Nighttime lights
Video 13	Normal	Smoldering ground in a smoky forest
Video 14	Normal	Road monitoring
Video 15	Normal	Red moving object
Video 16	Normal	Flashing lights at night
Video 17	Normal	Nighttime lights in the rain
Video 18	Normal	Shop lights at night
Video 19	Normal	Flashing warning light
Video 20	Normal	Fire-like objects and smoke

In addition to our proposed method, four commonly used deep neural networks were selected for use in place of the network proposed in this paper for comparison. Among them, ShuffleNetV2 [55] and MobileNetV2 [46] are excellent lightweight neural networks that have been recently proposed, and VGG16 [58] and ResNet50 [59] are ordinary CNNs with many applications. In Table 7, we compare five models in terms of run time and accuracy. Clearly, our proposed method is superior to ShuffleNetV2 and MobileNetV2 based on its false positive rate, false negative rate, and accuracy. The false positive rate is lower by 1.41%-2.30. The false negative rate is lower by 0.78%-1.5%. The accuracy is higher by 1.45%-1.69%. However, the frame rate is somewhat lower than that of MobileNetV2, which is far higher than the general video frame rate requirements. The resulting accuracy is not very different from that of VGG16 or ResNet50, but our method achieves a shorter run time than these methods do. Thus, our proposed method can better balance accuracy and run time for fire detection.

TABLE 7. Comparison with different deep neural network models on DS3.

Method	False positive rate (%)	False negative rate (%)	Accuracy (%)	Frame rate (fps)
Our method	2.67	1.19	97.93	39
ShuffleNetV2	4.97	1.97	96.24	35
MobileNetV2	4.08	2.69	96.48	42
VGG16	2.54	1.70	97.80	23
ResNet50	2.41	1.04	98.14	18

IV. DISCUSSION

A. COMPARISON OF FIRE DETECTION RESULTS WITH RECENT RESEARCH

We tested our method with three different device configurations: an Intel Core i7-8750H CPU with 8 GB of RAM, and an Nvidia GeForce GTX 1060 with 6 GB of onboard memory, an Intel(R) Core(TM) i7-4810MQ CPU with 8 GB of RAM, and an Intel(R) Core(TM) i7-5500U CPU with 8 GB of RAM. We compared our method with the most advanced methods reported to date in terms of the frame rate, accuracy, and false positive rate on DS2. Our method can balance time and accuracy better than the other methods, as shown in Table 8. Real-time detection can be achieved in all three device configurations, with frame rates of 41 fps, 36 fps, and 27 fps. In addition, the table shows that the methods of Foggia et al. and Lascio et al. have faster run times, reaching frame rates of 60 fps and 70 fps, respectively. However, we achieve an accuracy increase of 4.39%-5.35% and a false positive rate decrease of 4.34%-9.43% by comparison.

B. APPLICABILITY OF OUR METHOD IN SPECIAL SCENARIOS

The ultimate goal of fire detection is to increase the accuracy while reducing the rates of false positives and false negatives. However, the situations depicted in real videos are complex, including interfering factors such as fire viewed from far away, artificial lights, red objects and other moving objects, as shown in Fig. 8. A high false positive rate still occurs when deep learning models alone are used for fire detection. Our method of exploiting both motion-flicker-based dynamic features and deep static features can solve these problems. First, region-of-interest acquisition is performed based on dynamic features. Second, fire detection is performed based on static features.

In the first phase, the acquisition of regions of interest based on dynamic features can enable the extraction of fires or other moving objects viewed from a long distance, as shown in Fig. 8 (a), enabling us to focus on the spectral and textural features of such fire regions. Furthermore, the introduction of flicker

detection can eliminate some fire-like interferences from among the candidate moving objects, such as artificial lights and red objects, as shown in Fig. 8 (b). However, some items may be still missed during this first phase of detection in complex video scenarios, for example, moving fog or a car.

In the second phase, static features are used to further identify fire, which can eliminate these interferences, as shown in Fig. 8 (c) and (d). To improve the robustness of the proposed AL-CNN, different categories of fire images were used in the training process, thereby improving the accuracy and reducing the false positive rate for fire detection.

TABLE 8. Detailed comparison of the accuracy and run time of our method with those of other state-of-the-art methods on DS2.

Method	Frame rate (fps)	Accuracy (%)	False positive rate (%)	Remarks
Our method	41	97.94	2.33	Nvidia GeForce GTX 1060 with 6 GB of onboard memory
Our method	36	97.94	2.33	Intel(R) Core(TM) i7-4810MQ with 8 GB of RAM
Our method	27	97.94	2.33	Intel(R) Core(TM) i7-5500U with 8 GB of RAM
Muhammad et al. [57]	20	94.50	8.87	Nvidia TITAN X (Pascal) with 12 GB of onboard memory
Foggia et al. [6]	60	93.55	11.76	Intel dual core T7300 with 4 GB of RAM
Lascio et al. [12]	70	92.59	6.67	-
Habiboğlu et al. [60]	20	90.32	5.88	Dual core 2.2 GHz

FIGURE 8. Applicability of our method in special scenarios.

C. ANALYSIS OF DYNAMIC FEATURE EXTRACTION

During dynamic feature extraction, the recall should be as high as possible to ensure that the AL-CNN model can subsequently be effectively applied for fire detection. Notably, the recall during dynamic feature extraction depends on the value chosen for λ (equation (5)). To investigate the impact of different thresholds on the recall, multiple types of fire videos, including videos of fires from the burning of solid fuel, liquid fuel, and gaseous fuel, were used in experiments to determine the optimal threshold. The receiver operating characteristic (ROC) curve for region-of-interest acquisition with different thresholds for the extracted dynamic features are shown in Fig. 9. The preliminary threshold range is set to 0-1, and the step size is 0.05. It can be seen from the figure that the true positive rate remains the same when the value of λ is in the range of 0-0.5 and decreases when the value of λ is increasing in the range of 0.5-1. By contrast, the false positive rate continuously decreases when the value of λ is increasing in the range of 0-1. To obtain a high true positive rate while balancing the true positive rate and false positive rate, the value of λ is set to 0.5.

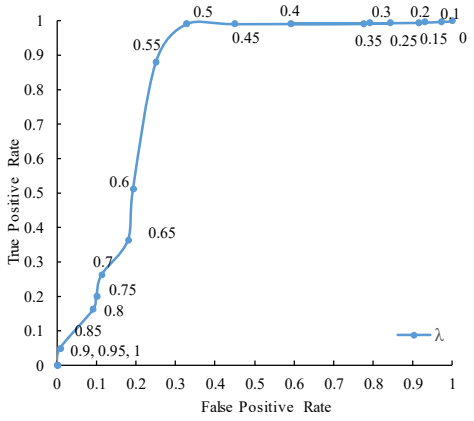

FIGURE 9. ROC curve for different thresholds in dynamic feature extraction

To further prove the effectiveness of the dynamic feature extraction procedure, we performed 4 experiments on DS2 and DS3 without considering dynamic features (with only the AL-CNN) and with the consideration of dynamic features (with background subtraction, flicker detection and the AL-CNN). The results are shown in Table 9. On both datasets, better results are achieved by considering dynamic features. Without dynamic feature extraction, the false positive rate is increased by 9.1%-11.85%, the false negative rate is increased by 6.69%-8.79%, and the accuracy is reduced by 8.67%-10.62%. As seen from the above analysis, considering both the dynamic and static features of fire can effectively improve the accuracy of fire detection and reduce the rates of false positives and false negatives.

TABLE 9. Comparison of detection results with and without the consideration of dynamic features

Test dataset	Method	False positive rate (%)	False negative rate (%)	Accuracy (%)
Test dataset 2	Without dynamic features	11.43	7.53	89.27
	With dynamic features	2.33	0.84	97.94
Test dataset 3	Without dynamic features	14.52	9.98	87.31
	With dynamic features	2.67	1.19	97.93

D. EFFECTIVENESS OF OUR METHOD

Early in the development of a fire is the best time to extinguish it, and the time elapsed between the ignition of a fire and its detection is an important factor to consider when evaluating the ability to achieve early fire detection. Therefore, five fire videos were divided into five stages in accordance with the changes in the fire characteristics during the burning process, i.e., ignition, development, fierce burning, decay, and extinction, as shown in Fig. 10. These five fire videos represent different scenarios, including indoor and outdoor scenes.

FIGURE 10. Examples of the whole evolution of a fire event. (a) Ignition. (b) Development. (c) Fierce burning. (d) Decay. (e) Extinction.

In addition to the commonly used evaluation criteria of accuracy and run time, the time from ignition to detection was recorded. In addition, we recorded the detection accuracy during the fire evolution process for each video separately, as shown in Fig. 11. The amounts of elapsed time until fire detection is achieved for the 5 videos are 2.5 s, 2 s, 1 s, 1.5 s, and 2 s. The average accuracies are 97.97%, 98.53%, 98.3%, 97.80%, and 98.11%. For all videos, it is possible to detect the fire within 3 s with high accuracy.

To analyze the detection accuracy of the method for different fire combustion stages, we analyzed the fire detection accuracy in each of the five stages; the results are shown in Fig. 12. The average accuracy across all five videos is 97.25%, 99.93%, 100%, 98.93%, and 94.63% for the first through the fifth stages, respectively. For the ignition stage, the accuracy is 97.25%, which can meet the needs of early fire detection. For the development, fierce burning, and decay stages, the average accuracy results are 99.93%, 100%, and 98.93%,

respectively, which are also suitable for achieving accurate fire detection. During the extinction stage, the accuracy is 94.63%. This analysis reveals that the method proposed in this paper can achieve more accurate detection in the early stages of a fire.

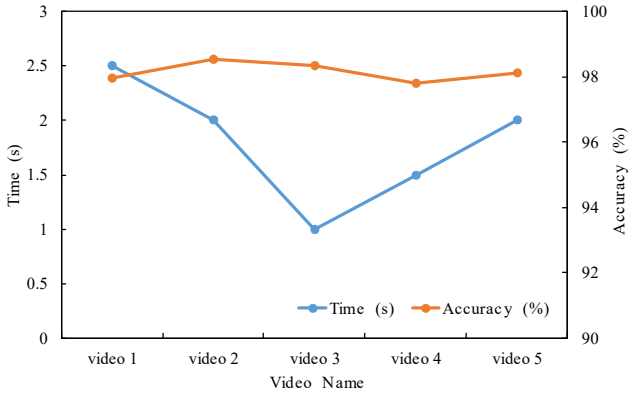

FIGURE 11. Time required for and average accuracy of fire detection for each video.

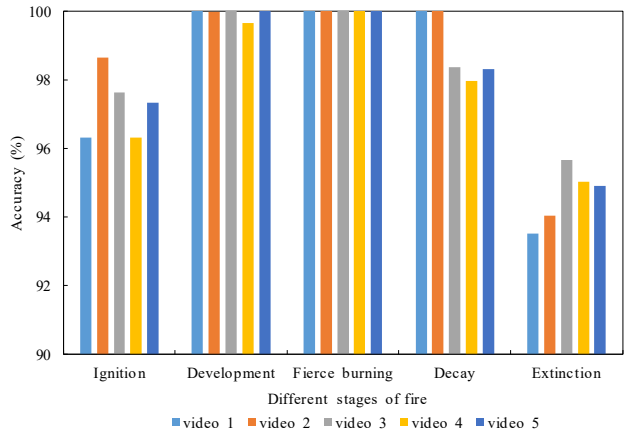

FIGURE 12. Fire detection accuracy in different stages of fire evolution.

V. CONCLUSIONS AND FUTURE WORK

In recent years, with the development of computer vision technology, deep learning has been applied for fire detection by many researchers. Although such applications are feasible under certain conditions, their efficiency needs to be improved, and complex video scenarios must be considered. Motivated by these considerations, an efficient fire detection method is proposed in this paper. Our method offers several advantages over other recent fire detection methods. First, both motion and flicker features are considered, which enables us to more effectively extract dynamic features. In addition, our method relies on an adaptive lightweight neural network, which can effectively extract deep static features with a low computational cost. Finally, experimental results prove that our method achieves state-of-the-art performance in terms of its accuracy and false alarm rate and that our method is applicable to complex video scenarios. In general, our proposed method has better application prospects than other state-of-the-art methods and is suitable for use in public safety management systems.

In this study, we concentrated solely on fire detection. In future work, we will conduct in-depth research on fire spread prediction and spatial positioning based on existing research. We hope that our research can support the intelligent suppression of fire in its early stages and provide improved fire detection and fire suppression methods for fire management in the area of public safety.

REFERENCES

[1]. K. Muhammad, J. Ahmad, I. Mehmood, S. Rho, and S. W. Baik, “Convolutional neural networks based fire detection in surveillance videos.” *IEEE Access*, vol. 6, pp. 18174-18183. Mar. 2018

[2]. Y. Luo, L. Zhao, P Liu, and D. Huang, “Fire smoke detection algorithm based on motion characteristic and convolutional neural networks.” *Multimed. Tools Appl.,* vol.77, pp. 15075-15092. Aug. 2017

[3]. L. Shi, F. Long, C. Lin, and Y. Zhao, “Video-Based Fire Detection with Saliency Detection and Convolutional Neural Networks.” *in Proc. 14th Int. Symp. Neural Networks (ISNN),* 21-26 Jun. 2017, pp. 299-309.

[4]. B. Kim, and J. Lee, “A Video-Based Fire Detection Using Deep Learning Models.” *Appl. Sci.-Basel*, vol. 9, pp. 2862. Jul. 2019

[5]. S. Khan, K. Muhammad, S. Mumtaz, S. W. Baik and V. H. C. de Albuquerque, "Energy-Efficient Deep CNN for Smoke Detection in Foggy IoT Environment." *IEEE Internet Things J.,* vol. 6, pp. 9237-9245, Dec. 2019.

[6]. P. Foggia, A. Saggese, and M. Vento, “Real-time fire detection for video-surveillance applications using a combination of experts based on color, shape, and motion.” *IEEE Trans. Circuits Syst. Video Technol.,* vol. 25, pp. 1545-1556. Jan. 2015

[7]. A. Rafiee, R. Dianat, M. Jamshidi, R. Tavakoli, and S. Abbaspour, “Fire and smoke detection using wavelet analysis and disorder features.” *in Proc. 3rd Int. Conf. Comput. Res. and Dev.*, Mar.2011, pp. 262–265.

[8]. T. Chen, P. Wu, and Y. Chiou, “An early fire-detection method based on image processing.” *in Proc. Int. Conf. on Image Process.*, 24-27 Oct. 2004, pp. 1707-1710.

[9]. T. Celik, and H. Demirel, “Fire detection in video sequences using a generic color model.” *Fire Saf. J.,* vol. 44, pp. 147-158. Feb. 2009

[10]. N. I. binti Zaidi, N. A. A. binti Lokman, M. R. bin Daud, H. Achmad, and K. A. Chia, “Fire recognition using RGB and YCBCR color space.” *ARPN J. Eng. Appl. Sci*., 2015, vol. 10, pp. 9786-9790. Nov. 2015

[11]. J. Seebamrungsat, S. Praising, and P. Riyamongkol, “Fire detection in the buildings using image processing.” *in Proc. 3rd ICT Int. Student Proj. Conf. (lSPC)*, 26-27 Mar. 2014, pp. 95-98.

[12]. D. R. Lascio, Greco, A. Saggese, and M. Vento, “Improving fire detection reliability by a combination of video analytics.” *in Proc. 11th Int. Conf. Image Anal. Recognit. (ICIAR)*, 22-24 Oct. 2014, pp. 477-484.

[13]. G. Marbach, M. Loepfe, and T. Brupbacher, “An image processing technique for fire detection in video images.” *Fire Saf. J.* vol. 41, pp. 285-289. Jun. 2006

[14]. Y. Qiang, B. Pei, and J. Zhao, "Forest Fire Image Intelligent Recognition based on the Neural Network." *J. Multimed*, vol. 9, pp. 449-455. Mar. 2014

[15]. K. Dimitropoulos, P. Barmpoutis, and N. Grammalidis, "Spatio-temporal flame modeling and dynamic texture analysis for automatic video-based fire detection." *IEEE Trans. Circuits Syst. Video Technol.,* vol. *25*, pp. 339-351. Jul. 2014

[16]. B. U. Toreyin, Y. Dedeoglu, and A. E. Cetin, "Flame detection in video using hidden markov models." *in Proc. IEEE Int. Conf. Image Process*., 11-14 Sept. 2005, pp. 2457-2460.

[17]. G. E. Hinton, and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks." *Science*, vol. 313, pp. 504-507. Jul. 2006

[18]. A. Krizhevsky, I. Sutskever, and G.E. Hinton, "ImageNet classification with deep convolutional neural networks." *in Proc. 25th Int. Conf. Neural Inf. Process. Syst*., 3-6 Dec.2012 pp. 1097-1105.

[19]. A. Graves, M. Liwicki, S. Fernández, R. Bertolami, H. Bunke, and J. Schmidhuber, "A novel connectionist system for unconstrained handwriting recognition." *IEEE Trans. Pattern Anal. Mach. Intell.,* vol. 31, pp. 855-868. May. 2008.

[20]. G. E. Hinton, S. Osindero, Y. and W. Teh, "A fast learning algorithm for deep belief nets." *Neural Comput.,* vol. *18,* pp. 1527-1554. May. 2006.

[21]. J. Yang, B. Jiang, B. Li, K. Tian, and Z. Lv, "A fast image retrieval method designed for network big data." *IEEE Trans. Ind. Inform.,* vol. 13, pp. 2350-2359. Jan. 2017.

[22]. R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Region-based convolutional networks for accurate object detection and segmentation." *IEEE Trans. Pattern Anal. Mach. Intell.,* vol. *38*, pp. 142-158. May. 2015.

[23]. T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space." [Online]. Available: https://arxiv.org/abs/1301.3781

[24]. G. E. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition." *IEEE Trans. Audio Speech Lang. Process.,* vol. 20, pp. 30-42. Apr. 2011.

[25]. D. Erhan, C. Szegedy, A. Toshev, and D. Anguelov, "Scalable object detection using deep neural networks." *in Proc. IEEE conf. Comput. vision pattern Recognit.*, 23-28 Jun. 2014, pp. 2147-2154.

[26]. G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, and C. I. Sánchez, "A survey on deep learning in medical image analysis." *Image Anal.,* vol. 42, pp. 60-88. Dec. 2017.

[27]. K. Muhammad, S. Khan, V. Palade, I. Mehmood and V. H. C. de Albuquerque, "Edge Intelligence-Assisted Smoke Detection in Foggy Surveillance Environments." *IEEE Trans. Ind. Inform.*, vol. 16, pp. 1067-1075, Feb. 2020.

[28]. S. Frizzi, R. Kaabi, M. Bouchouicha, J. M. Ginoux, E. Moreau, and F. Fnaiech, "Convolutional neural network for video fire and smoke detection." *in Proc. 42nd Annu. Conf. IEEE Ind. Electron. Soc. (IECON)*, 24-27 Oct., 2016, pp. 877-882.

[29]. J. Sharma, O. C. Granmo, M. Goodwin, and J. T. Fidje, "Deep convolutional neural networks for fire detection in images." *in Proc. 18th Int. Conf. Eng. Appl. Neural Networks* (EANN), 25-27 Aug.2017, pp. 183-193.

[30]. D. Shen, X. Chen, M. Nguyen, and W. Q. Yan, "Flame detection using deep learning." *in Proc. Int. Conf. Control, Autom. Rob. (ICCAR)*, 20-23 Apr.2018, pp. 416-420.

[31]. C. Hu, P. Tang, W. Jin, Z. He, and W. Li, "Real-Time Fire Detection Based on Deep Convolutional Long-Recurrent Networks and Optical Flow Method." *in Proc. 37th Chin. Control Conf. (CCC)*, 25-27 Jul. 2018, pp. 9061-9066.

[32]. Q. Zhang, J. Xu, L. Xu, and H. Guo, "Deep convolutional neural networks for forest fire detection." *in Proc. Int. Forum Manage. Educ. Inf. Technol. Appl. (IFMEITA)*, 30-31 January 2016, pp. 568-575.

[33]. K. Muhammad, J. Ahmad, and S. W. Baik, "Early fire detection using convolutional neural networks during surveillance for effective disaster management." *Neurocomputing*, vol. 288, pp. 30-42. May. 2018.

[34]. K. Muhammad, S. Khan, M. Elhoseny, S. H. Ahmed, and S. W. Baik, "Efficient Fire Detection for Uncertain Surveillance Environment." *IEEE Trans. Ind. Inform.,* vol. 15, pp. 3113-3122. Feb. 2019.

[35]. Y. Wu, X. Wu, S. Lu, J. Zhang, and K. Cen, "Novel Methods for Flame Pulsation Frequency Measurement with Image Analysis." *Fire Technol.,* vol. 48, pp. 389-403. May. 2011.

[36]. J. Chen, Y. He, and J. Wang, "Multi-feature fusion based fast video flame detection." *Build. Environ.,* vol. 45, pp. 1113-1122. May. 2010.

[37]. A. Filonenko, D. C. Hernández and K. Jo, "Fast Smoke Detection for Video Surveillance Using CUDA." *IEEE Trans. Ind. Inform.*, vol. 14, pp. 725-733, Feb. 2018.

[38]. P. KaewTraKulPong, and R. Bowden, "An improved adaptive background mixture model for real-time tracking with shadow detection." *in Proc. Video-based surveill. Syst.*, 2002, pp. 135-144.

[39]. Z. Zivkovic, and F. Van Der Heijden, "Efficient adaptive density estimation per image pixel for the task of background subtraction." *Pattern Recognit. Lett.,* vol. 27, pp. 773-780. May. 2006

[40]. H. Wu, D. Wu, and J. Zhao, "An intelligent fire detection approach through cameras based on computer vision methods." *Process Saf. Environ. Protect.,* vol. 127, pp. 245-256. Jul. 2019

[41]. A. Hamins, J. C. Yang, and T. Kashiwagi, "An experimental investigation of the pulsation frequency of flames." *in Symp. (int.) on Combust.,* 1992, pp. 1695-1702.

[42]. Z. An, H. Yuan, and Y. Qu, "Data acquisition application in the measurement research and analysis of flame flicker frequency." *Fire Saf. Sci.,* 2000, vol. 9, pp. 43-47.

[43]. S. Ioffe, and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift." [Online]. Available: https://arxiv.org/abs/1502.03167

[44]. Ramachandran, Prajit, Barret Zoph, and Quoc V. Le. "Searching for activation functions." [Online]. Available: https://arxiv.org/abs/1710.05941

[45]. A. Howard, M. Sandler, G. Chu, L. C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan, Q. V. Le, and H. Adam, "Searching for mobilenetv3." *in Proc. IEEE Int. Conf. Comput. Vision,* 20-26, Oct. 2019, pp. 1314-1324.

[46]. M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks." *in Proc. 31st IEEE/CVF Conf. on Comput. Vision and Pattern Recognit. (CVPR)*, 18-23 Jun., 2018, pp. 4510-4520.

[47]. K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition." *IEEE*

Trans. Pattern Anal. Mach. Intell., vol. 37, pp. 1904-1916. Sept. 2015.

[48]. D. Y. Chino, L. P. Avalhais, J. F. Rodrigues, and A. J. Traina, "Bowfire: detection of fire in still images by integrating pixel color and texture analysis." *in Proc. 28th SIBGRAPI Conf. Graphics, Patterns Images*, 26-29 Aug., 2015, pp. 95-102.

[49]. B. C. Ko, S. J. Ham, and J. Y. Nam, "Modeling and formalization of fuzzy finite automata for detection of irregular fire flames." *IEEE Trans. Circuits Syst. Video Technol.,* vol. 21, pp. 1903-1912. Dec. 2011

[50]. V. Hüttner, C. R. Steffens, and S. S. Da Costa Botelho, "First response fire combat: Deep leaning based visible fire detection." *in Proc. 14th Latin Am. Rob. Symp. (LARS) / 5th Braz. Rob. Symp. (SBR)*, 08-11 Nov.2017, pp. 1-6.

[51]. A. Chenebert, T. P. Breckon, and A. Gaszczak, "A non-temporal texture driven approach to real-time fire detection." *in Proc. IEEE Int. Conf. Image Process*., 11-14 Sept. 2011 pp. 1741-1744.

[52]. C. R. Steffens, R. N. Rodrigues, and S. S. Da Costa Botelho, "Non-stationary VFD Evaluation Kit: Dataset and Metrics to Fuel Video-Based Fire Detection Development." *in Proc. 12th Latin Am. Rob. Symp. (LARS) / 3rd Braz. Rob. Symp. (SBR)*, 28 Oct. - 01 Nov., 2015, pp. 135-151.

[53]. F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and< 0.5 MB model size." [Online]. Available: https://arxiv.xilesou.top/abs/1602.07360

[54]. X. Zhang, X. Zhou, M. Lin, and J. Sun, "Shufflenet: An extremely efficient convolutional neural network for mobile devices." *in Proc. IEEE Conf. Comput. Vision Pattern Recognit*., 18-23 Jun. 2018, pp. 6848-6856.

[55]. N. Ma, X. Zhang, H. T. Zheng, and J. Sun, "Shufflenet v2: Practical guidelines for efficient cnn architecture design." *in Proc. Eur. Conf. on Comput. Vision (ECCV)*, 08-14 Sept. 2018, pp. 116-131.

[56]. A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications." [Online]. Available: https://arxiv.xilesou.top/abs/1704.04861

[57]. K. Muhammad, J. Ahmad, Z. Lv, P. Bellavista, P. Yang, and S. W. Baik, "Efficient deep CNN-based fire detection and localization in video surveillance applications." *IEEE Trans. Syst. Man Cybern. Syst.*, vol. 49, pp. 1419-1434. Jul. 2019.

[58]. K. Simonyan, and A. Zisserman, "Very deep convolutional networks for large-scale image recognition."[Online]. Available: https://arxiv.org/abs/1512.03385

[59]. K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition." [Online]. Available: https://arxiv.org/abs/1512.03385

[60]. Y. H. Habiboğlu, O. Günay, and A. E. Çetin, "Covariance matrix-based fire and flame detection method in video." *Mach. Vis. Appl.,* vol. 23, pp. 1103-1113. Sept. 2011.