Outdoor fire detection on the video frames including fire zones close to the fire-like objects recorded by a fixed camera

Mahmood Talebiyan Mashhadi [1]; Farahnaz Mohanna[2*]

1- Department of Communication Engineering, University of Sistan and Baluchestan, Zahedan, Iran, e-mail: mahmood22805.talebiyan@gmail.com
2- Department of Communication Engineering, University of Sistan and Baluchestan, Zahedan, Iran, e-mail: f_mohanna@ece.usb.ac.ir
[*]Corresponding author

Received: 2021-12-19
Accepted: 2022-02-15

Abstract
In this paper, an automatic outdoor fire detection method is proposed for the fire videos recorded by a fixed camera. First, a new set of color rules is introduced to eliminate the non-fire pixels as much as possible while detecting the fire zone pixels completely. Next, the texture and flicker effect features are extracted from the detected fire zone, to remove the remainder of non-fire pixels if still any non-fire pixel exists. The texture feature is extracted by using the angular second moment. To extract the flicker effect feature, the time prehistory signal of color components of each fire zone pixel is obtained and passed through a half band high pass filter. Finally, the Ward classifier clusters the fire features to separate the fire zone pixels from the non-fire. At the various steps of the proposed method, the morphology process is also used to improve the accuracy of fire detection. The proposed method is applied to the 200 different fire videos including the fire-like objects. Simulation results indicate 6% to 56% improvement on performance of the proposed method in comparison to the similar ones.

Keywords
Fire detection, Outdoor fire video frames, Fixed camera, Feature extraction, Flicker effect feature, Clustering.

1. Introduction

Fire detection is one of the most important categories of the modern world, in which life of living beings, especially humans, is threatened. It is necessary that fire is detected through remote monitoring, without human intervention due to fire damages including fatal, financial, and environmental. In this respect, machine vision systems are superior. In these systems, a wide area is monitored under supervision of at least a camera fitted at an appropriate height. So, by applying image processing methods, to the fire video captured by the camera, information such as area, shape, and boundary of the fire zone can be found, that are useful in subsequent analysis, including fire volume estimation, spread rate, spread frontier. There are also factors affect the fire video images, which are categorized as factors that are selected by user such as, camera installing place, camera mode, imaging rate, and imaging quality, and factors that are out of the user control, such as, fire occurrence time, kind of the burned materials, and environmental conditions. In addition, daytime fires are different from fires at night based on color and texture. Fire areas in the daytime usually have roughness, while fire areas at night are brighter and their surfaces are almost uniform. Also, images at night are darker than daytime [1]. Furthermore, in fire detection using image processing and machine vision, two steps should be considered, 1) correct identification of the fire presence or absence in video frames, and 2) specifying the fire zone precisely in the presence of fire. Output information of item 1 is used for alert system, and output information of item 2 is used to obtain the area, shape, and boundary of the fire zone and to obtain time variation of these parameters for subsequent analysis [2]. So far, researches which have carried out, have good performance in 1, but their performance is not satisfactory with regard to 2. Because in these methods, in a wide range of fires types, a significant part of the fire pixels is eliminated, and in many cases, inconsistencies are created in the fire zone.

Fire detection research so far, has used a variety of fire features, such as color, motion and flicker effect, which are applied at the pixel level, and other features, like texture, that can be implemented at the block level [3]. Ultimately features such as boundary roughness, and shape should be implemented on the fire spot surface, and only indicate that the fire area exists or not, which is not our goal [3]. Our goal is to determine exact area of the fire in an image. On the other hand, it is known that fire has both motion and flicker effect. The flicker feature is under the set of motion characteristics. In fact, each dynamic zone such as fire which has the flicker effect, has also movement. But the vice versa is not necessarily true. It can be said that when the flicker feature is used to identify the fire areas, no longer is needed to consider the fire movement [4]. Therefore, use of the motion detection techniques only causes computational complexity. Based on these issues, to identify the outdoor fire zone, only considering the fire color, texture, and flicker effect is enough.

In this paper, a method is proposed to solve the above problems using fire color, texture and flicker effect while in removing the non-fire zones, its optimal performance is maintained. In order to test the proposed method, 200 variety of outdoor fire videos are collected from various bases. According to the proposed method results, areas of fire-like objects removed and 80% of the fire along frames, maintained. Proposed method also uses method in [2] for fire detection on fire videos at night. The rest of the paper is organized as follow. Past work is presented in section 2. Proposed method is described in section 3. Simulation results are shown in section 4 and section 5 concludes the paper.

2. Related Work

Fire detection algorithms generally divided into two categories: traditional approaches based on computer vision, and AI-based approaches using machine learning and deep learning.

O. Gunay et al. [1] developed a method for video-based detection of wildfire at night that comprised four sub-algorithms: (i) slow moving video object detection, (ii) bright region detection, (iii) detection of objects exhibiting periodic motion, and (iv) a sub-algorithm interpreting the motion of moving regions in video. Individual decisions of the sub-algorithms combined together using a least-mean-square based decision fusion approach, and fire/no-fire decision was reached by an active learning method. J. Zhao et al. [2] proposed an automatic method for forest fire detection from video. Based on the 3D point cloud of collected sample fire pixels, the Gaussian mixture model was built and helped segment some possible flame regions in a single image. Then the specific flame pattern was defined for forest, and three types of the fire colors were labelled accordingly. The static SVM classifier was trained with eleven static features including color distributions, texture parameters and shape roundness, and filtered the segmented results. A. Gupta et al. [3] presented a fire detection alarming system according to a rule-based technique using RGB and HSV color spaces and the spatial analysis based on wavelet transform. For differentiation of fire and fire-like objects, the threshold-based spatial energy methodology was applied. Texture analysis using the local binary pattern was also performed when fire or fire-like candidate having spatial energy near to the threshold of fire pixel. B. Ugur Toreyin et al. [4] proposed a method to detect fire and flames in real-time by processing the video data generated by an ordinary camera monitoring a scene. Different clues such as, fire flicker, Quasi-periodic behaviour in flame boundaries, color variation, flame regions, were detected applying different methods. Finally, all of the clues were combined to reach a final decision for the fire and flame detection. D. H. Lee et al. [5] extracted fire candidate regions in video frames with HSV color model. The motion information of flames and smoke were modelled, and the optical flow in a fire candidate region was estimated. The spatial temporal distribution of the optical flow vectors was analysed. Finally, the probability of fire in successive video frames were accumulated which reduced false-positive errors. F. Gong et al. [6] considered combination of flame motion and color detection as fire pre-processing stage. An algorithm of flame centroid stabilization based on spatiotemporal relation was also presented and the flame region centroid of each frame were calculated and added the temporal information. Next, they extracted features including spatial variability, shape variability, and area variability of the flame to improve the accuracy of recognition. Finally, they used Support Vector Machine (SVM) classifier for training, completed the analysis of candidate fire images, and achieved automatic fire monitoring. F. Karimi Zarchi et al. [7] used color, spatial and temporal information by using three filters. The first filter separated red color area as a primary fire candidate. The second filter worked based on differences between fire candidate areas in the sequence of frames. In the third filter, variation of red channel in the candidate area was computed and compared with the threshold which was updating continuously. A. Mouelhi et al. [8] presented a tracking method for fire regions using an artificial neural network (ANN) combined with a hybrid geometric active contour (GAC) model based on Bayes error energy functional for forest wildfire videos. Firstly, an estimation function was built with local and global information collected from RGB, HIS, and YC_bC_r color spaces using Fisher's linear discriminant analysis and a trained ANN in order to get a preliminary fire pixel classification in each frame. This function controlled the active contour model providing a refined fire segmentation in each processed frame. M. Ajith and M. Martinez-Ramon [9] proposed a vision-base fire and smoke segmentation system which used spatial, temporal and motion information. The fusion of information was done using multiple features such as optical flow, divergence and intensity values. Then the Markov Random Field segmentation algorithm was used to classify the spatial interactions of pixels using these features. A. Gaur et al. [10] discussed the fire sensing technologies for the discrepancies in the development of hardware and algorithm. The review presented an overview of the existing practices in the areas of fire sensing and control system. They focused on the capability to detect fire, reduce the detection of false positives, the ability to notify the occupants, passes the information and the status of the fires to the fire department, and the automatic control capability of the occupants' safety and controlling functions. The major elements of the fire moment, such as surrounding heat, flame, smoke, and gases level, were discussed with their merits, demerits, measurement benchmarks, and measuring the span of the parameters. X. Chen et al. [11] proposed a fire identification algorithm by merging fire segmentation and multi-feature fusion of fire. According to the relationship between R and Y components, the improved YC_bC_r models were established for initial fire segmentation under reflection and non-reflection conditions, and these conditions were judged by comparing the areas obtained by the two improved YC_bC_r models. Second, an improved region growing algorithm was proposed for fine fire segmentation by making use of the relationship between the seed point and its adjacent points.

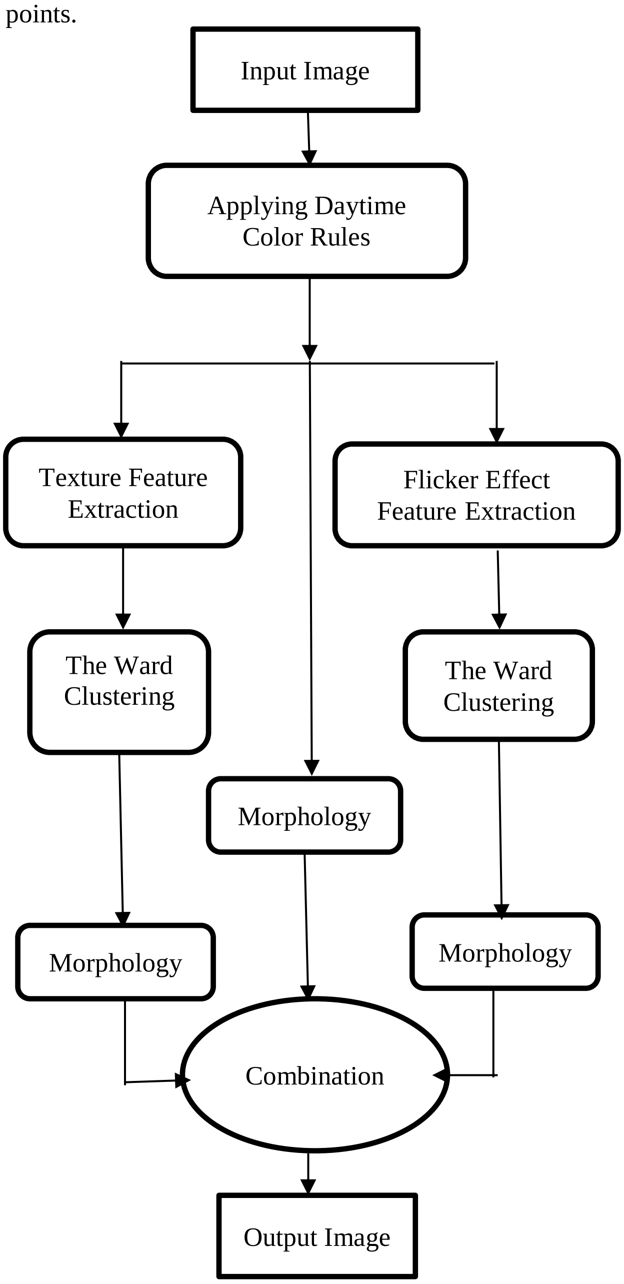

Fig. 1. Block diagram of the proposed algorithm

The seed points were determined using the weighted average of centroid coordinates of each segmented image. Finally, the quantitative indicators of fire identification were given according to the variation coefficient of fire area, the dispersion of centroid, and the circularity. Y. Xie et al. [12] presented a method that exploited both motion-flicker-based dynamic and deep static features for video fire detection. Dynamic features were extracted by analyzing the differences in motion and flicker features between fire and other objects in videos. An adaptive lightweight convolutional neural network (ALCNN) was proposed to extract the deep static features of fire. Finally, these features of fire were combined to establish a video fire detection method. K. Muhammad et al. [13] proposed an efficient CNN based system for fire detection in videos captured in uncertain surveillance scenarios. The approach used lightweight deep neural networks with no dense fully connected layers, making it computationally inexpensive. Experiments were conducted on benchmark fire datasets. Considering the accuracy, false alarms, size, and running time of their system, they believed that it was a suitable candidate for fire detection in uncertain Internet of Things (IoT) environment for mobile and embedded vision applications during surveillance. K. Muhammad et al. [14] proposed an energy-friendly and computationally efficient CNN architecture, inspired by the SqueezeNet architecture for fire detection, localization, and semantic understanding of the fire scene. It used smaller convolutional kernels and contained no dense, fully connected layers, which helped
keep the computational requirements to a minimum. Despite its low computational needs, the experimental results demonstrated that their solution achieved accuracies that were comparable to other, more complex models, mainly due to its increased depth. X. Wu et al. [15] presented a method to detect smoke using camera sensors. Static features including texture, color, wavelet, edge orientation histogram and dynamic features such as direction, speed, and change of the motion were extracted from the fire smoke to train, and were tested with different combinations. The Ada-Boost classifier was used to improve classification accuracy and was leaded to a satisfactory performance.

3. Proposed Method

Block diagram of the proposed method is presented in Fig. 1., which its steps are as following.

3.1 Our Proposed Color Rules

So far by applying any defined color rules, a significant part of the fire zone is eliminated. Based on the observations, experiments, and measurements on applying all the previous color rules, our proposed color rules are seen in Equations (1) to (4). In the proposed color rules, attempts are made to eliminate non-fire pixels as much as possible while detecting the fire zone pixels completely. The proposed color rules are defined in the RGB, YC_bC_r and HIS color spaces. For each image pixel, the color coordinates R, Y, C_b, C_r and H are defined at spatial coordinates (i, j).

$$R_1 = \begin{cases} 1 & \text{if } (R(i.j) > 170) \text{and } (R(i.j) > R_{av} \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

$$R_2 = \begin{cases} 1 & \begin{array}{l} \text{if } \left(C_r(i.j) > C_b(i.j)\right) \& (C_b(i.j) \le C_{bav} \\ \& \left(Y(i.j) > C_b(i.j)\right) \text{and } (Y(i.j) \ge Y_{av}) \end{array} \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

$$R_3 = \begin{cases} 1 & \text{if } H(i.j) < 0 \cdot 46 \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

$$R_a = R_1 \cap R_2 \cap R_3 \quad (4)$$

where R_{av}, Y_{av}, and C_{bav}, are respectively the average intensity of R, Y, and C_b components in the input image. Due to applying the color rules, a binary mask is obtained with a same size as the input image, that its value is 1 in fire place and 0 in other places. If after applying the proposed color rules, the non-fire pixels cannot be removed completely, the remainder of non-fire pixels will be removed using texture and flicker effect features in the next steps of our method.

3.2 Fire Texture Extraction

The fire zone in the daytime has an uneven surface. So, we decided to use the uniformity feature for more differentiating of the fire zones and fire-like objects by using the angular second moment (ASM) [15]. On the other hand, the non-uniformity of fire zone texture in the HSI color space is influenced by its three components. Then, we choose RGB color space to analyse the fire zone texture. In the RGB color space, the fire zone texture non-uniformity is affected only by G and B components, and R does not share much. Therefore, using only G and B components to analyse the fire zone texture, requires less computation than using three H, S and I components in the HIS color space. Using only G and B components, also improves the ASM parameter performance in more accurately differentiation of the fire zones and fire-like objects. To compute the ASM parameter, each video frame is divided into 6x6 blocks. Inside each block and for each pixel, a 1x2 color vector consisting of G and B color components, is obtained. It is also assumed that pixels which the Euclidian distances of their color vectors are less than 0.5, are the same. Then, for each block pixel, uniformity of each block pixel is computed by Equation (5).

$$\mathrm{ASM} = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \mathrm{P}(i.j)^2 \quad .\mathrm{N} = 6 \tag{5}$$

In continue, the Ward classifier [16] is applied to cluster blocks based on their ASM values. In the proposed method, clusters that their mean values are smaller than a threshold value, are considered as the suspicious clusters of fire. The threshold value is selected 0.15 by try and error. Due to applying the texture feature extraction by Equation (6), a binary mask is obtained with a same size as the input image, that its value is 1 in fire block place, and 0 in the other places.

$$\mathrm{R_b} = \begin{cases} 1 & \begin{array}{c} \text{if ASM}(x.y) \in \text{Cluster}_i \\ \& \ (\text{Mean} - \text{Cluster}_i < 0 \cdot 15) \end{array} \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

where (x,y) is coordinates of each block and i changes from 1 to the number of clusters.

3.3 Fire Flicker Effect Extraction

Intensity of color components of a pixel over multiple frames is fluctuating which is called flicker effect. This feature can also be used to identify the fire zones. In the proposed method, intensity variations in a pixel color component at coordinate $(i\ ,j)$ in the nth frame is shown by $X_n\ (i\ ,j)$ and is obtained for each pixel of each frame, and feed to a two-stage filter bank [4] in Fig. 2, consists of a half-band high-pass filter with coefficients of {-0.25, 0.5, -0.25} and a half-band low-pass filter with coefficients {0.25, 0.5, 0.25}. Outputs of this filter are sub-signals $d_n\ [i\ ,j]$ and $e_n\ [i\ ,j]$ that their number of zero-crossing over a period of time, is considered as a flicker effect. However, considering this number as a fire flicker effect, the fire surface becomes discrete and grain-bound.

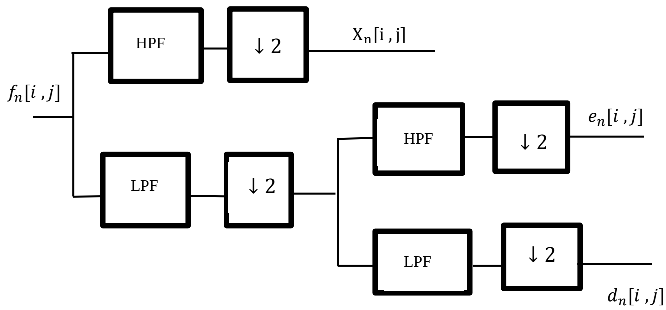

Fig. 2. A two-stage filter bank [4]

On the other hand, C_r component of the YC_bC_r color space, can also be used to extract the fire flicker effect because, the C_r color component is more distinguished, by the fire zone and fire-like objects. So, in the proposed method, each frame is divided into 6×6 blocks. Then, for each pixel block, the mean value of the number of zero-crossing of $d_n\ [i\ ,j]$ in the time interval from the first frame to the current frame in a video sequence, is computed and named average of flicker of block (AFB). In continue, the Ward classifier is applied to cluster blocks, based on their AFB values. In the proposed method, clusters that their mean values are higher than a threshold value, are considered as the suspicious clusters of fire. The threshold value is selected 0.5 by try and error. Due to the applying the flicker effect extraction by Equation (7), a binary mask is obtained with a same size of the input image, that its value is 1 in place of the fire block, and 0 in the other places.

$$R_c = \begin{cases} 1 & \begin{array}{c} if\ AFB(x.y) \in Cluster_i \\ \& \ (Mean - Cluster_i > 0 \cdot 5) \end{array} \\ 0 & otherwise \end{cases} \tag{7}$$

where (x,y) is coordinate of each block and i changes from 1 to the number of clusters.

3.4 The Ward Clustering

In texture and flicker effect features extraction for different fire videos in our experiments, except of a coherent cluster of the fire zone, the non-fire area has more than one cluster, which this number depends on the scene illumination conditions. We solve this problem by using the Ward clustering [16], which is one of the strongest hierarchical clustering methods, in which the number of clusters is automatically determined. In the Ward process, each data is considered as a cluster. Then for each pair of possible clusters among all existing clusters, those two clusters whose sum of the squares of clustering data from their merging are minimized, is selected. Then the two selected clusters are combined. These steps are repeated until the number of clusters reaches to the desired number.

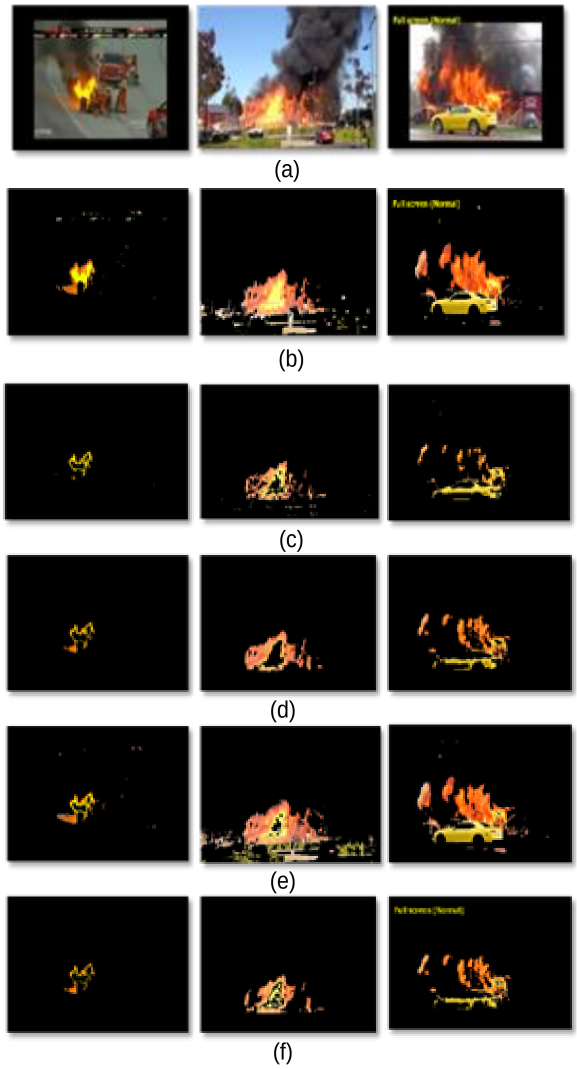

Fig. 3. Comparison of the color rules output. a) Input image, b) output of color rules in our method c) in [17], d) in [18] e) in [19] f) in [20].

4. Experimental Results

In order to evaluate performance of our method, fire photos and videos are collected from various bases that have a wide variety of outdoor fire types. Our database consists of 100 daytime fire photos and 200 daytime fire videos.

4.1 Results of Applying Our Color Rules

The goal of applying our color rules was to reduce the error caused by the removal of fire pixels. A number of results of applying our color rules, in comparison to the results of applying color rules in [17], [18], [19] and [20] are shown in Fig. 3. In addition, an experiment is designed to evaluate the results as following. 100 different fire images are selected from our database. Next, the fire areas are kept and the remainder are removed manually in all the images. Then the percent of fire pixels removal (*FPR%*) are calculated for all these output images by Equation (8). If, after applying the color rules of each method, the FPR% exceeds over 20% for an image, it is considered as an error.

$$FPR\% = 100 \times \frac{Number\ of\ Removed\ Fire\ Pixels}{Total\ Number\ of\ Fire\ Pixels} \quad (8)$$

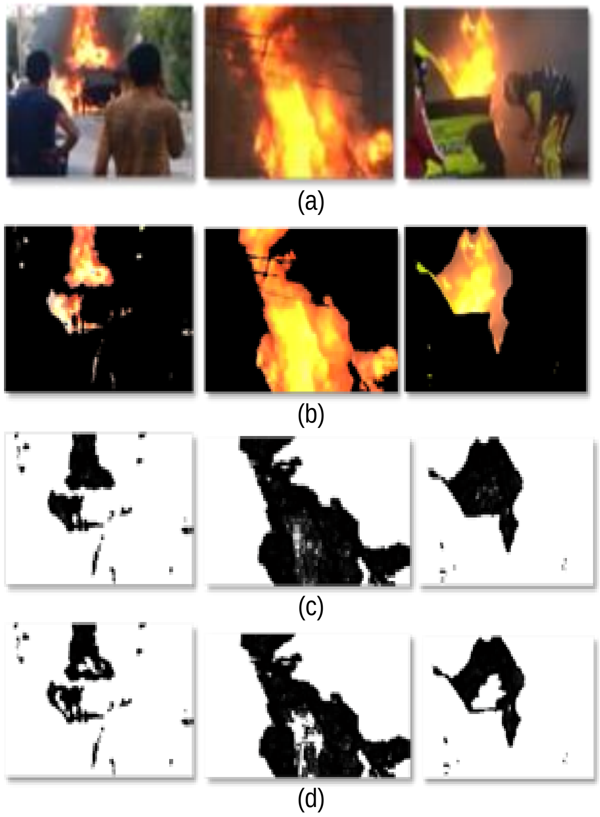

Fig. 4. Comparison of the numerical matrix image of texture feature, a) input image, b) color rules output, c) numerical matrix image of the texture extracted by our method, d) by the method in [2].

The above experiment was carried out for 100 daytime fire images in our database. The FPR% was calculated for all these images applying methods in [17], [18], [19], [20], and our method. The results of this experiment are listed in Table 1.

Table 1. Percent of error in the fire pixels removal

Method	Our	in [17]	in [18]	in [19]	in [20]
FPR%	9%	65%	62%	15%	46%

4.2 Results of Fire Texture Extraction

In order to show that texture feature is extracted more precisely by our method, method in [2] is selected for comparison which is more similar to our method. For a number of daytime fire images, color rules are applied. Then a numerical matrix is obtained for each output, in which 6×6 blocks are considered. The numerical value of each block is equal to the ASM of that block. We did this for both, our method and method in [2], and to compare these matrices, they are scaled in interval of (0, 255). The images of these numerical matrices are shown in Fig. 4. In the numeric matrices images, the smaller the ASM for a block, the block is seen darker, and vice versa. As it is seen in Fig. 4, a significant percentage of the fire block in

[2] has a large ASM value, which after clustering and thresholding, part of the fire is removed, however, performance of our method is more optimal.

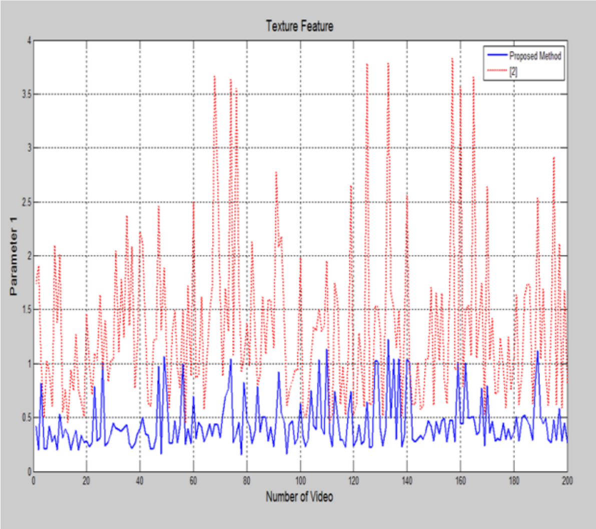

Fig. 5. Comparison of our method (blue) with method in [2] (red), based on *parameter1* for texture feature. Vertical and horizontal axes are *parameter1* and number of videos respectively.

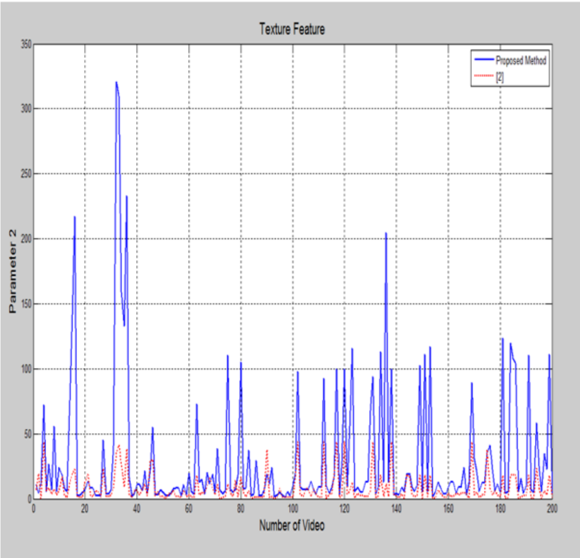

Fig. 6. Comparison of our method (blue) with method in [2] (red), based on *parameter2* for texture feature. Vertical and horizontal axes are *parameter2* and number of videos respectively.

In order to compare our method with method in [2] in texture extraction, an experiment is designed as follows. We considered 200 fire videos from our database. Next, in frame 31 for each of these videos, the color rules output blocks are considered and the ASM value of each block is computed by our method and method in [2]. After clustering on the blocks, mean and standard deviation of the fire cluster as well as the mean of non-fire cluster are calculated. In cases where there is more than one non-fire cluster, all non-fire clusters are merged and then their average is calculated.

Mean, and standard deviation of the fire cluster, and mean of the non-fire cluster are respectively named, Mean_Fire, and Sigma_Fire, and Mean_Non_Fire. A well-defined feature for good separation of an object in the clustering process, has two characteristics, 1-degree of similarity and proximity of numerical values of the object's components in feature space, 2-degree of non-similarity and distance between numerical values of the object's components from numerical values of the other components. A feature that works best in these two mentioned characteristics, will be more powerful in differentiation of the fire and non-fire zones. For the first characteristic, we define *parameter1* by Equation (9).

$$parameter1 = \frac{Sigma\_Fire}{Mean\_Fire} \quad (9)$$

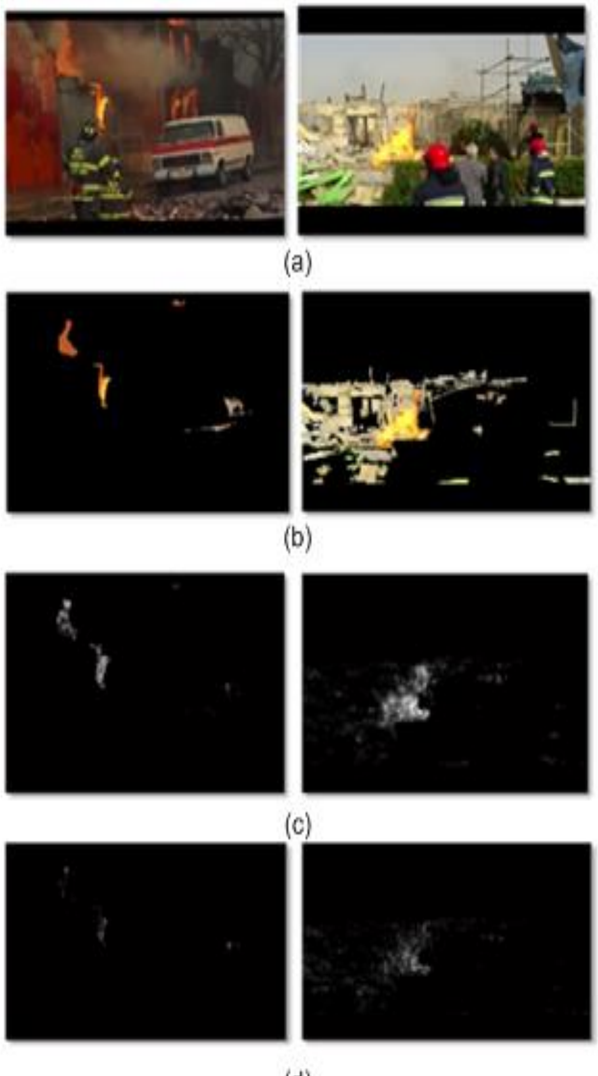

Fig. 7. Comparison of the numerical matrix image of flicker effect, a) output of the color rules, b) numerical matrix image of flicker effect for our method, d) for the method in [4]

After clustering based on a selected feature, for the clustered fire zone, the lower *parameter1* is, the better is the feature used in Equation (9). For the second characteristic, we define *parameter2* by Equation (10).

$$parameter2 = \frac{|\mathrm{Mean\_Fire} - \mathrm{Mean\_Non\_Fire}|}{\mathrm{Mean\_Fire}} \quad (10)$$

After clustering based on a selected feature, the higher the *parameter2* is, the better is the feature by Equation (10). In Fig. 5, our method and method in [2] are compared using *parameter1*. As it is seen, in 99% of the tested fire videos, the *parameter1* value for our method is less than that for the method in [2], which indicates the better performance of our method for fire detection. We also obtained mean of the *parameter1* values by applying our method and method in [2] for 200 tested fire videos as following.

Mean of $Parameter1$ values for method in [2] $= 1.3227$
Mean of $Parameter1$ values for our method $= 0.4462$

As it is seen, average of *parameter1* in these 200 videos has decreased by 66% for our method in comparison with method in [2]. In Fig. 6, our method and method in [2] are compared using *parameter2*. As it is seen, in 87% of the tested fire videos, *parameter2* value for our method is greater than that for method in [2], which indicates better performance of our method. We also obtained mean of the *parameter2* values by applying our method and method in [2] for 200 tested fire videos as following.

Mean of $Parameter2$ values for method in [2] $= 8.2037$
Mean of $Parameter2$ values for our method $= 30.2053$

As it is seen, average of *parameter2* values in these 200 videos has increased by 73% for our method in comparison with method in [2].

4.3 Results of Fire Flicker Effect Extraction

In order to compare our method with method in [4], output of the color rules in a randomly selected frame for a number of fire videos are considered. In this frame, the numerical matrices of the fire areas are obtained. For our method, this matrix consists of 6×6 blocks, where the numerical value of each block is equal to the AFB value of the block. Also, for our method, the flicker effect calculations is done on Cr component of the daytime fire videos. However, for method in [4], this matrix presented the zero-crossing number of R component intensity variation of each pixel. To compare these matrices, they are scaled in the range of (0,255). Images of these numerical matrices for a number of daytime fire videos, are presented in Fig. 7. As it is seen, the numerical matrix image of the fire flicker effect in our method has advantages including, the fire zones are brighter, more continuous, having larger numerical values and are easier separated after clustering and thresholding. While the identified fire zone by method in [4], is discrete and granular. To compare the flicker effect extraction by our method and method in [4], an experiment is designed as following. We considered 200 fire videos from our database. Next, in frame 31 for each of these videos, output of the color rules are considered. Then for each pixel of this frame by method in [4], the number of zero-crossing of $d_n[i,j]$ and $e_n[i,j]$ are calculated and clustering is performed. In our method, the AFB value for each block is calculated and clustering performed on the blocks. In continue, in both methods, mean and standard deviation of the fire cluster as well as mean of the non-fire cluster are calculated. In cases there is more than one non-fire cluster, the non-fire clusters are merged and then mean is calculated. Finally, *parameter1*, and parameter2, are computed by Equations (9), and (10) for experiment results evaluation. In Fig. 8, our method and method in [4] are compared by using *parameter1*. As it is seen, in 72% of the tested fire videos, *parameter1* values for our method is less than those for method in [4], which indicates the better performance of our method. Mean of *parameter1* values also is obtained in 200 tested fire videos for both methods as following.

Mean of *Parameter1* values for method in [4] = 0.7704
Mean of *Parameter1* values for our method = 0.6189

The mean of *parameter1* values in these 200 fire videos has decreased by 20% for our method than method in [4]. performance of our method. Mean of *parameter2* values are obtained in 200 tested fire videos for our method and method in [4] as following. In Fig. 9, our method and method in [4] are compared by using *parameter2*. As it is seen, in 72% of the tested fire videos, *parameter2* values for our method are greater than those for method in [4], which indicates the better performance of our method.

Mean of *Parameter2* values for method in [4] = 0.5991
Mean of *Parameter2* values for our method = 0.7474

The mean of *parameter2* values in these 200 fire videos has increased by 25% in our method than method in [4].

4.4 Results of Applying Proposed method and Comparison

In order to evaluate the proposed method performance in comparison with a number of recent fire detection methods, we implemented the proposed method and all the selected methods in MATLAB R2018a-64 bit and windows 8.1 operating system. The programs are run on personal computer with Intel(R) core(TM) 2 Duo CPU @ 2.53 GHz, 4GB RAM and applied them to our 200 tested fire videos. The accuracy results are shown in Table 2. The average of running time of the proposed method on the 200 tested fire videos also is 18.47 sec.

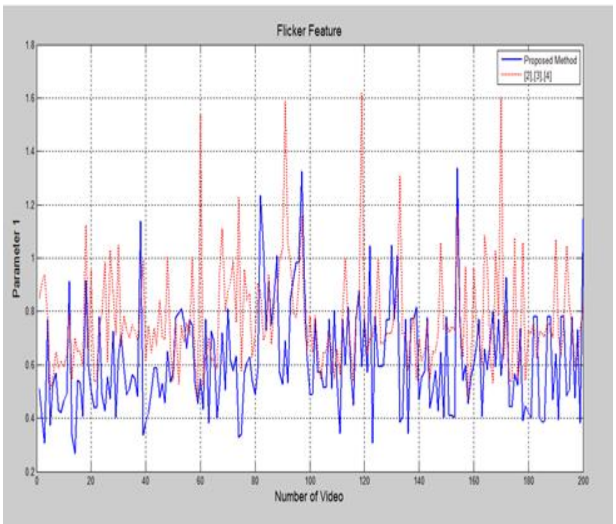

Fig. 8. Comparison of our method (blue) with method in [4] (red), based on *parameter1* for the flicker effect feature. Vertical and horizontal axes are *parameter1* and number of videos respectively.

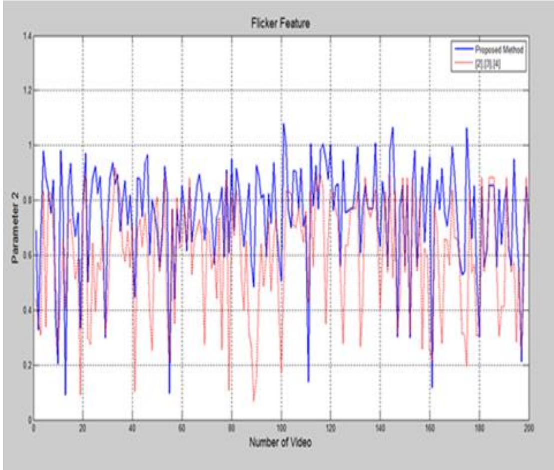

Fig.9. Comparison of our method (blue) with method in [4] (red), based on *par2* for the flicker effect feature. Vertical and horizontal axes are *par2* and number of videos respectively.

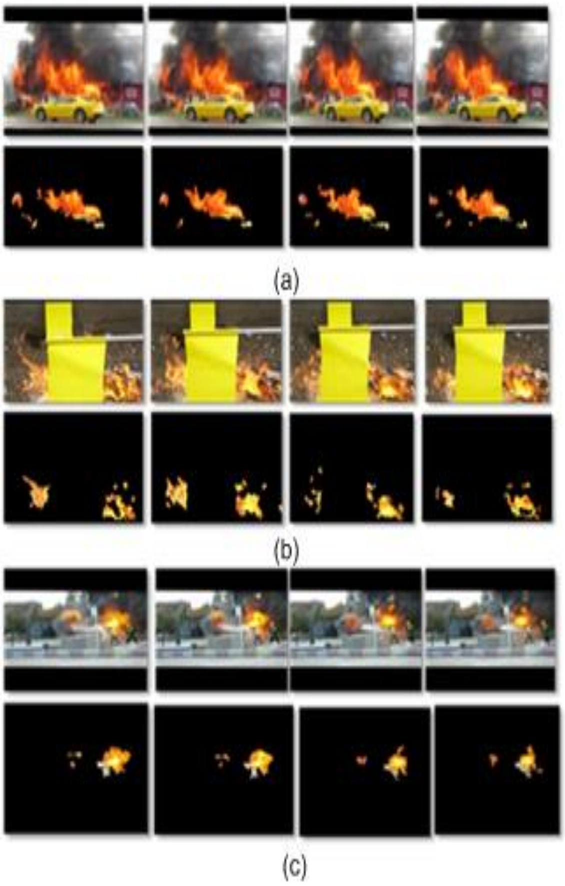

Fig. 10. Results of our method on 4 different frames of a) video#1, b) video#2, c) video#3

In result, the proposed method is not only very simple but also is more accurate and enough fast to detect outdoor fire and localize it with minimum removal of fire pixels.

Table 2. Accuracy comparison of the proposed method with a number of fire detection methods

Method	Accuracy
in [5]	87.23%
in [6]	95.63%
in [7]	96.95%
Proposed	97.16%

4.5 Visually Performance

In order to evaluate our method performance visually, three challenging videos are selected and output of our method over 4 different frames of these videos are displayed in Fig. 10. In the first video, a moving yellow car passes in front of the fire. Since the car has same color as fire and moves and also, the fire zone is sticking to the car body, this video is a difficult test that our method has shown acceptable performance on it. The second video, illustrates a moving yellow flag in front of the fire. The third video shows a firefighter with yellow stripes and yellow hat who is approaching to the fire. In both videos, our method has performed well. In addition, Fig. 11 shows more results of applying our method to the fire videos including fire-like objects.

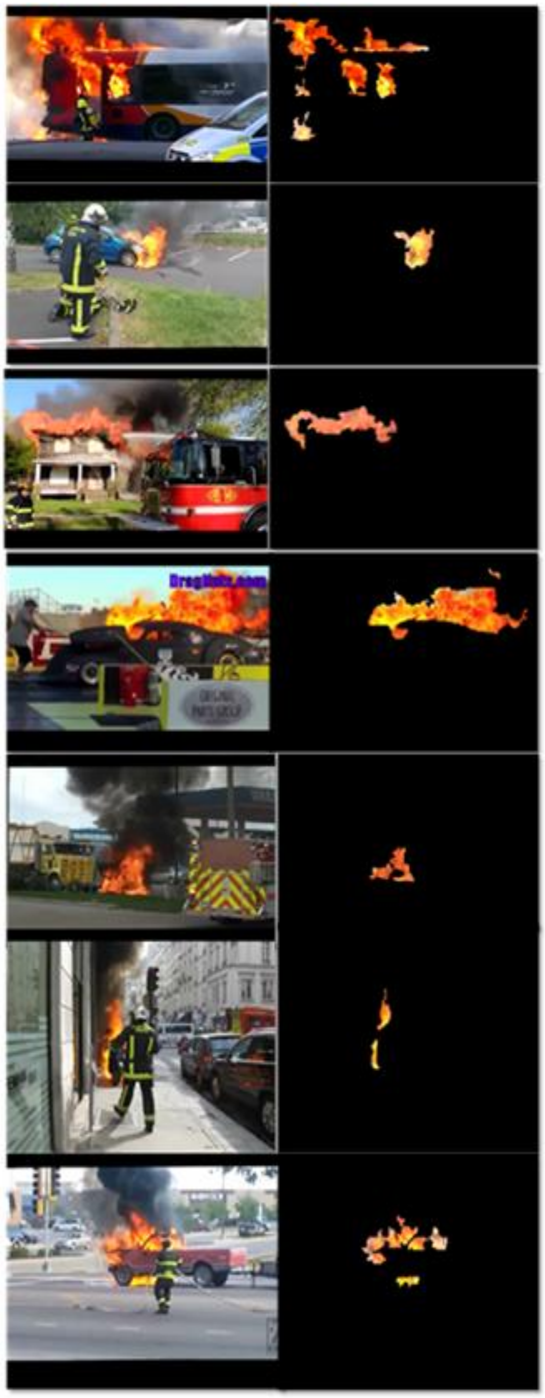

Fig. 11. Results of the proposed method on various fire images in the presence of fire-like objects

5. Conclusion

Fire detection system based on deep learning approaches improved substantially the accuracy of fire detection, however the major concern with these approaches are their implementation in the real-world surveillance networks, due to their high memory and computational requirements for inference.

In this research, we proposed a simple and automatic fire detection method that can detect fire area close to the fire-like objects in the fire video frames at a higher accuracy than other similar classical methods. For this purpose, among all the fire features, color, texture, and flicker effect were selected. First, new color rules were proposed that removed less percentage of the fire pixels. Then, to remove the whole of non-fire areas, texture and flicker effect features were also extracted from the fire zones.

For fire texture extraction, the ASM value was computed using G and B color components instead of H color component which caused to prevent removal of some part of the fire. In order to extract the flicker effect feature, instead of extracting the zero-crossing number of the discrete wavelet decomposition signals of each pixel, the AFB value of each 6×6 fire block was computed, which prevented the discrete and granularity of the fire area.

In addition, the Cr component was used which has more fluctuations and more differentiation of the fire and fire-like areas. Furthermore two experiments were designed to evaluate our method results on fire detection videos on our database. By using our method, 80% percentage of the fire pixels were maintained, and all the fire-like areas and non-fire were eliminated.

It also has to be noticed that in our method the camera must be fixed for its correct functional. So in the geographic areas where extreme winds are premature, the fixed surveillance cameras are shaken. As a result, the image will fluctuate, which will disrupt the correct function of our method in the flicker effect extraction. Therefore, use of the image stabilization techniques that can come up with our method to tackle such a challenge can be recommended.

References

[1] O. Gunay, K. Tasdemir, B. U. Toreyin, and A, E. Cetin, “Video based wildfire detection at night”, Fire Safety, Vol. 44, No. 6, pp. 860-868, 2009. https://doi.org/10.1016/j.firesaf.2009.04.003

[2] J. Zhao et al., “SVM based forest fire detection using static and dynamic features”, Computer Science and Information Systems, Vol. 8, No. 3, pp. 821-841, 2011. https://doi.org/10.2298/CSIS101012030Z

[3] A. Gupta, N. Bokde, D. Marathe and Kishore, “A novel approach for video based fire detection system using spatial and texture analysis”, Indian Journal of Science and Technology, Vol. 11, No. 19, pp. 1-17, 2018. https://doi.org/10.17485/ijst/2018/v11i19/88825

[4] B. Ugur Toreyin, Y. Dedeoglu, U. Gudukbay, and A. Enis Cetin, “Computer vision based method for real-time fire and flame detection”, Pattern Recognition Letters, Vol. 27, pp. 49-58, 2006. https://doi.org/10.1016/j.patrec.2005.06.015

[5] D. H. Lee, S. H. Lee, T. Byun, N. IK. Cho, “Fire detection using color and motion models”, IEEE Transaction on Smart and Computing, Vol. 6, No. 4, pp. 237-245, 2017. https://doi.org/10.5573/IEIESPC.2017.6.4.237

[6] F. Gong, C. Li, W. Gong, X. Li, X. Yuan, Y. Ma, and T. Song, “A real-time fire detection method from video with multifeature fusion”, Computational Intelligence and Neuroscience, Vol. 2019, No. 1, pp. 1-17, 2019. https://doi.org/10.1155/2019/193971

[7] F. Karimi Zarchi, V. Derhami, A. Latif, and A. Ebrahimi, “Fire detection using video sequences in urban out-door environment”, Signal and Data Processing, Vol. 16, No. 3, pp. 61-78, 2019.

[8] A. Mouelhi, M. Bouchouicha, M. Sayadi, and E. Moreau, "Fire tracking in video sequences using geometric active contours controlled by artificial neural network", 4th International Conference on Advanced Systems and Emergent Technologies, Tunisia, 2020. https://doi.org/10.1109/IC_ASET.2020.9318289

[9] M. Ajith, and M. Martinez-Ramon, "Unsupervised segmentation of fire and smoke from infra-red videos", IEEE Access, Vol. 17, pp. 182381-182394, 2019. https://doi.org/10.1109/ACCESS.2019.2960209

[10] A. Gaur et al., "Fire Sensing Technologies: A Review", IEEE Sensors, Vol. 19, No. 9, pp. 3191 – 3202, 2019. https://doi.org/10.1109/JSEN.2019.2894665

[11] X. Chen, A. Qing, K. Yu, and Y. Ban, "A novel fire identification algorithm based on improved color segmentation and enhanced feature data", IEEE Transaction on Instrumentation and Measurement, Vol. 70, 2021. https://doi.org/10.1109/TIM.2021.3075380

[12] Y. Xie, J. Zhu, Y. Cao, Y. Zhang, D. Feng, Y. Zhang, and M. Chen, "Efficient video fire detection exploiting motion-flicker-based dynamic features and deep static features", IEEE Access, Vol. 8, pp. 81904-81917, 2020. https://doi.org/10.1109/ACCESS.2020.2991338

[13] K. Muhammad, S. Khan, M. Elhoseny, S. H. Ahmed, and S. Wook Baik, "Efficient fire detection for uncertain surveillance environment", IEEE Transaction on Industrial Informatics, Vol. 15, No. 5, pp. 3113-3122, 2019. https://doi.org/10.1109/TII.2019.2897594

[14] K. Muhammad, J. Ahmad, Z. Lv, P. Bellavista, P. Yang, S. Wook Baik, "Efficient deep CNN-based fire detection and localization in video surveillance applications", IEEE Transactions on Systems, Man, and Cybernetics: Systems, Vol. 49, No. 7, pp. 1419 – 1434, 2018. https://doi.org/10.1109/TSMC.2018.2830099

[15] X. Wu, X. Lu, and H. Leung, "A video based fire smoke detection using robust AdaBoost", Sensors, Vol. 18, No. 11, 3780, 2018. https://doi.org/10.3390/s18113780

[16] N. Randriamihamison, N. Vialaneix, and P. Neuvial, "Applicability and interpretability of Ward's hierarchical agglomerative clustering with or without contiguity constraints", Journal of Classification, Vol. 38, pp. 363-389, 2021. https://doi.org/10.1007/s00357-020-09377-yff

[17] G. F. Shidik, F. N. Adnan, C. Supriyanto, and R. A. Pramunendar, and P. Andono, "Multi-color feature, background subtraction and time frame selection for fire detection", *Int. Conf. on Robotics, Biomimetic, Intelligent Computational Systems*, 2013. https://doi.org/10.1109/ROBIONETICS.201306743589

[18] V. Vipin, "Image processing based forest fire detection", *Int. Journal of Emerging Technology and Advanced Engineering*, vol. 2, no. 2, pp. 87-95, 2012.

[19] H. Jin, and R. B. Zhang, "A fire and flame detecting method based on video", *Int. Conf. on Machine Learning and Cybernetics*, pp. 2347-2352, 2009. https://doi.org/10.1109/ICMLC.2009.5212165

[20] Ti Nguyen-Ti, Thuan Nguyen-Phuc, and Tuan Do-Hong, "Fire detection based on video processing method", *Int. Conf. on Advanced Technologies for Communications*, pp. 106-110, 2013. https://doi.org/10.1109/ATC.2013.6698087