Detection and Segmentation of Power Line Fires in Videos

Jie Shi, *Student Member, IEEE,* Wei Wang, *Student Member, IEEE,* Yuanqi Gao, *Student Member, IEEE,*
Nanpeng Yu, *Senior Member, IEEE*
University of California, Riverside
Riverside, California 92521
Email: {jshi005, wwang031, ygao024, nyu}@ucr.edu

***Abstract*—Poor vegetation management around power lines can cause severe fires that lead to tremendous economic losses, environmental degradation, and fatalities. The early discovery of a fire's presence is the key to avoiding catastrophic damages. In this paper, we propose a hybrid fire detection framework based on a deep convolutional neural network (CNN) and a pixel-based fire detector to automatically detect both the presence of fire and its scale and position information. The pre-trained deep CNN serve as a binary classifier to detect the presence of fire. The pixel-based fire detector is designed to find the fire pixels in the video frames, which indicate the scale and location of the fire. Case studies are carried out on six real-world videos to validate the proposed framework. It is shown that the proposed approach can effectively detect fire and locate the fire pixels in the testing fire videos.**

***Index Terms*—Video fire detection, power line, deep convolutional neural network, pixel-based fire detector.**

I. Introduction

Power lines can easily cause sparks and fires when encroached on by tree branches or other conducting objects. Facilitated by strong winds and dry conditions, these fires can grow quickly into catastrophic wildfires. Recently, California fire investigators announced that the power lines of Pacific Gas and Electric Company (PG&E) were confirmed to be the cause of the deadly Camp Fire in 2018 [1]. This wildfire killed 85 people and burnt a total of 153,336 acres, making it one of the most devastating fires in California's history.

Since early detection of a fire's presence is the key to avoiding catastrophic damages, electric utilities have been expanding their camera networks to achieve timely fire detection. There are two challenges to detecting wildfires from a vast network of cameras. First, manually watching live video from cameras is very labor intensive. Second, demanding system operators to quickly identify the scale and location of a fire is very unreasonable. Therefore, a technological solution of automatic detection of fire and identification of fire location and scale is highly desirable. The goal of this paper is to develop computer vision-based techniques to address the two challenges mentioned above.

Automatic video flame detection has been extensively studied in the literature. See [2] for a comprehensive review of the traditional approaches. Most of them are pixel-based methods that can easily locate fire and evaluate its scale. In recent years, researchers have started using deep neural networks to detect fire in either images [3] or videos [4]. It has been shown that deep convolutional neural networks achieve a higher fire detection accuracy than the traditional approaches [5]. In addition, deep learning based objective detection and instance segmentation methods such as Mask R-CNN [6] are capable of extracting fire region masks while performing fire detection. However, the success of these algorithms depend on obtaining a large amount of training images with boundaries for fire pixels, which is both labor-intensive and time consuming. Furthermore, deep learning based objective detection and instance segmentation methods take much longer training and testing time than pure classification algorithms.

In this study, we propose a novel two-stage automatic fire detection framework that exploits the merits from both the deep neural network and traditional video flame detection techniques. Specifically, our proposed framework consists of two key components. The first component is a 50-layer residual network (ResNet50) [7], which classifies given video frames into fire or non-fire images. The second component takes the fire images from the first component as inputs and identifies fire pixels to estimate the location and scale of the fire.

The rest of this paper is organized as follows. Section II presents the overall fire detection and positioning framework and the technical methods used in the two key components. Section III validates the proposed approach with real-world fire and non-fire videos. Section IV states the conclusion.

II. Overall Framework and Technical Methods

In this section, we first introduce the overall framework of the proposed video fire detection and positioning framework. Then we describe the technical methods used in the two key components, the deep convolutional neural network and the proposed pixel-based fire detector.

A. Overall Framework

The overall framework of our proposed two-stage video fire detection scheme is shown in Fig. 1. In the first stage, the current frame of a given video is classified as either a fire image or a non-fire image. If it is a fire image, then this frame is further processed by the pixel-based fire detector in the second stage. The pixel-based fire detector produces a fire region mask, which indicates the scale and location of a fire. Note that the pixel-based fire detector may not find any fire

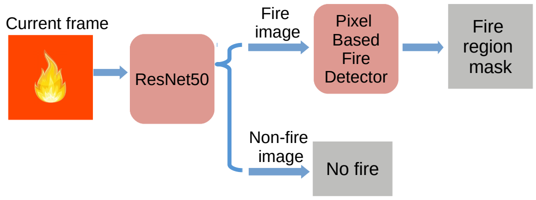


Fig. 1: The overall framework of the proposed approach.

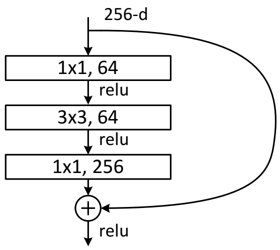


Fig. 2: A sample building block of ResNet50 [7].

pixel in a given frame even if it is classified as a fire image. In this case, the fire alert is still sent to the system operator without a fire region mask.

B. ResNet50

In this study, we adopt a 50-layer residual network as the binary classifier. The residual network is basically a deep convolutional neural network with identity short cut connections between certain layers. A residual network can be constructed by stacking individual building blocks [7]. Fig. 2 shows a sample building block in ResNet50. The number of filters varies for different blocks. Formally, a building block is defined as:

$$o_i = F(p_i, W_i) + p_i \tag{1}$$

where p_i and o_i are the input and output of block i. The function $F(p_i, W_i)$ represents the residual mapping (stacked convolutional layers), which will be learned. A complete ResNet50 is built by concatenating an input convolutional layer, a max pooling layer, different building blocks, an average pooling layer, and a fully connected layer in the end. Refer to [7] for the details of the structure of ResNet50. In this study, we use sigmoid activation function for the output layer. The output layer has only one neuron becuase our problem is a binary classification problem.

We train the ResNet50 using stochastic gradient decent based on Adam optimization algorithm [8]. The loss function to be minimized is the binary cross entropy given by

$$loss = -y_{nn} \log(\hat{y}_{nn}) - (1 - y_{nn}) \log(1 - \hat{y}_{nn}) \tag{2}$$

where $\hat{y}_{nn}$ is the output of the neural network. $y_{nn} \in \{0, 1\}$ is the true label of a given image. Label 0 and 1 correspond to non-fire images and fire images, respectively.

C. Pixel-based Fire Detection

The proposed pixel-based fire detector takes a frame classified as a fire image by ResNet50 as input and outputs a mask of fire pixels. The flowchart of the pixel-based fire detector is shown in Fig. 3. The fire frame is first processed by the color detection and motion detection modules which produce a fire color mask and a motion mask. The spatial variation check is then carried out on the regions covered by the fire color mask, which yields a spatial variation mask. Meanwhile, the logic AND of the fire color mask and motion mask is sent to the flick detection module that generates a flicker mask. The final fire region mask is obtained by performing a logic AND of the spatial variation mask and the flicker mask. We briefly introduce the motivation and the methodology used for each module below.

1) Color Detection: Typical flames have a color of either yellow or red. The color properties of a fire can be exploited to distinguish fire pixels from non-fire pixels. Numerous techniques have been proposed to detect fire based on the color of pixels in images. In this paper, we use a list of rules to find pixels with color properties of a fire.

Let (x_i, y_i) denote the spatial location of a pixel. The color of a pixel is represented in the YCbCr color space. Let Y_{mean}, Cb_{mean}, and Cr_{mean} denote the mean values of luminance, ChrominanceBlue, and ChrominanceRed channels of all the pixels in the frame. Then a pixel is classified as a fire color pixel if its color satisfies the following six rules adopted from [9]. These rules are selected due to their simplicity and computational efficiency.

Rule 1: $Y(x_i, y_i) > Cb(x_i, y_i)$, $Cr(x_i, y_i) > Cb(x_i, y_i)$
Rule 2: $Y(x_i, y_i) > Y_{mean}$
Rule 3: $Cb(x_i, y_i) < Cb_{mean}$, $Cr(x_i, y_i) > Cr_{mean}$
Rule 4: $|Cb(x_i, y_i) - Cr(x_i, y_i)| \geq \tau$
Rule 5: $f_1(Cr(x_i, y_i)) - Cb(x_i, y_i) \geq 0$
Rule 6: $Cb(x_i, y_i) - f_2(Cr(x_i, y_i)) \geq 0$
where $\tau = 40$. $f_1(x)$ and $f_2(x)$ are defined as follows:

$$f_1(x) = 7.79 \times 10^{-3} x^2 + 2.10x - 2.25 \tag{3}$$

$$f_2(x) = \begin{cases} 4.47 \times 10^{-2} x^2 - 16.94x + 1513.52 & x \leq 142 \\ 3.39 \times 10^{-5} x^2 + 0.77x - 98.31 & else \end{cases} \tag{4}$$

2) Motion Detection: In real-world cases, many non-fire objects such as the sunset and red traffic lights have a color very similar to a fire. Therefore, fire detection methods depending solely on color are likely to yield false positives. Note that uncontrolled fires are unstable and dynamic. Thus moving objection detection techniques can help reduce the number of false positives.

In this paper, we use a Gaussian mixture model (GMM) based background subtraction method [10], [11] to detect moving objects in a video. This approach is highly accurate and has shown good performance in outdoor environments. The advantage of this approach is that the background model can automatically adapt to the scene. The basic idea of the adopted motion detection algorithm is described here.

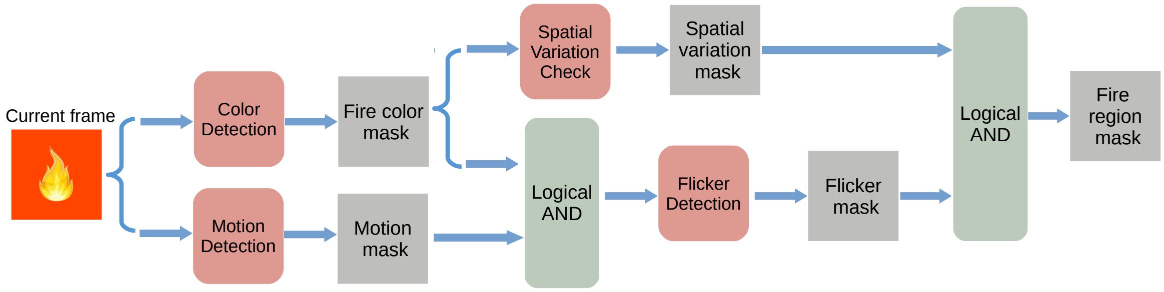


Fig. 3: Flowchart of the pixel-based fire detector.

Let $\boldsymbol{x}_{ij}^t$ be the RGB value of pixel (x_i, y_j) at time t and $X_T = \{\boldsymbol{x}_{ij}^t, \cdots, \boldsymbol{x}_{ij}^{t-T}\}$ be the set of T historic samples. The background model is defined as $p(\boldsymbol{x}_{ij}^t|BG)$, which is the distribution of $\boldsymbol{x}_{ij}^t$ given that $\boldsymbol{x}_{ij}^t$ is a background pixel. An estimate of a background model based on a GMM model with M components is initialized and updated recursively with streaming historic dataset X_T. The model only uses a subset of the M components as follows:

$$\hat{p}(\boldsymbol{x}_{ij}^t|BG) = \sum_{m \in M_B} \hat{\pi}_m \mathcal{N}(\boldsymbol{x}_{ij}^t; \hat{\boldsymbol{\mu}}_m, \hat{\sigma}_m^2 I) \tag{5}$$

where $\hat{\pi}_m$, $\hat{\boldsymbol{\mu}}_m$, and $\hat{\sigma}_m^2 I$ are the mixing weight, the mean vector, and the covariance matrix of the mth Gaussian component, respectively. M_B is the set of B components with the largest $\hat{\pi}_m$. Specifically, B is determined by

$$B = \underset{b}{\operatorname{argmin}} \left(\sum_{m=1}^{b} \hat{\pi}_m > (1 - c_f) \right) \tag{6}$$

where c_f is a tunable parameter. We refer the readers to [10] for the specific procedures of GMM parameter update.

3) Flicker Detection: Fire detection based on both color and motion information can still cause false positives. For example, a moving red vehicle can be easily mistaken for fire. Generally speaking, fire flames flicker in uncontrolled fires. This phenomenon can be captured by cameras because fire colored pixels can appear and disappear on the edge of turbulent flames. Therefore, a flicker detection module is introduced to further reduce the possibility of false positives.

The flicker frequency is often around 10 Hz regardless of the burning material and the burner [12], [13]. Therefore, flicker pixels can be detected by evaluating the spectrum of historic color values. Specifically, we use a two-stage filter bank based on discrete wavelet transform (DWT) to decompose the pixel color signals [14]. A two-stage filter bank can provide sufficient resolution in our application given that common videos have frames per second (fps) between 20 to 40. Fig. 4 shows the structure of the DWT filter bank. In this paper, we use Daubechies wavelet with 5 vanishing moments. T is a buffer of historic red channel values (or Y component in YUV color space) of a given pixel. The buffer size is set to be 45. T_H^1 is the detail coefficients of stage 1. T_H^2 and T_L^2 are the detail coefficients and approximation coefficients of stage 2, respectively. Given a video of 30 fps, T_H^1 covers the frequency

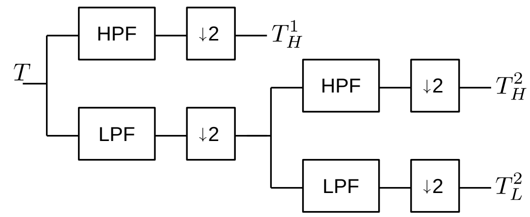


Fig. 4: Two-stage filter bank based on DWT.

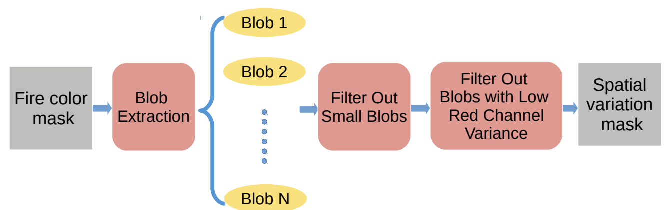
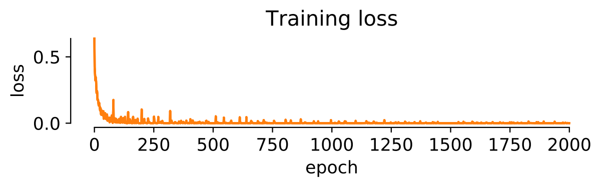


Fig. 5: Spatial variation check process.

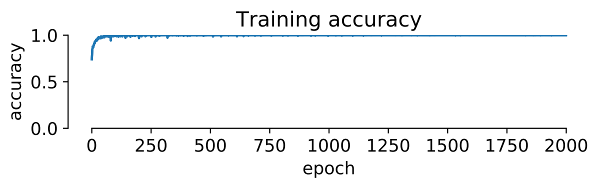
band between 7.5 Hz to 15 Hz. T_H^2 covers the frequency band between 3.75 Hz to 7.5 Hz. Therefore, we concatenate T_H^1 and T_H^2 into a single time series $T_F = [T_H^1, T_H^2]$. Then a pixel is labeled as a flicker pixel if the mean square root of its corresponding T_F is greater than a predefined threshold, which is set as 5 in the case study.

4) Spatial Variation Check: It has been discovered that flame colors of uncontrolled fires vary spatially across the fire region. In other words, the color value of the flame region has a significant variance. Therefore, this property can be exploited to distinguish fire flames from other objects with fire color but little color variance.

In this paper, we use a simple spatial variation check process illustrated in Fig. 5. First, we carry out connected-component labeling [15] to extract all the fire color blobs from the fire color mask produced by the color detection module. Then, we filter out the blobs that have a number of pixels less than 10. For each surviving blob, we calculate the variance of red channel values among all its pixels. If the variance is greater than a predefined threshold (100 in the case study), then the corresponding blob passes the spatial variation check.

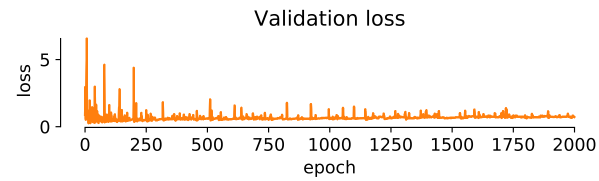


(a) Training loss.

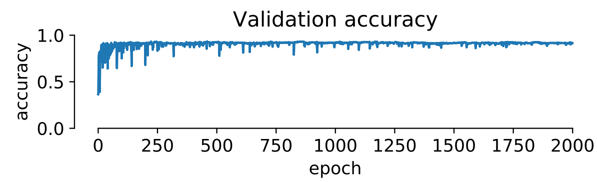


(b) Training accuracy.

Fig. 6: Training loss and accuracy.



(a) Validation loss.



(b) Validation accuracy.

Fig. 7: Validation loss and accuracy.

III. Case Study

In this section, we test our proposed video fire detection approach with several real-world fire and non-fire videos. First, our binary classifier ResNet50 is trained with fire and non-fire images downloaded from Google and Baidu. Then, the whole algorithm is evaluated on three fire videos and three non-fire videos downloaded from YouTube. The results will be discussed in detail in the following subsections.

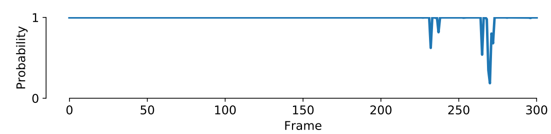
A. Training of ResNet50

ResNet50 is trained on 4,500 downloaded images, which include 1,660 fire images and 2,840 non-fire images. The images are resized to 224×244 and then divided randomly into two sets. The first set is the training dataset, which consists of 3,600 images. The second set is the validation dataset, which consists of 900 images. An NVIDIA RTX 2080 Ti GPU is used to accelerate the training process.

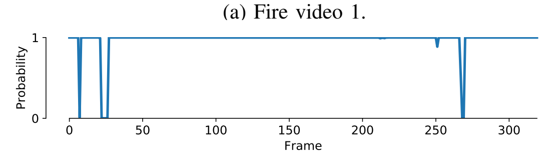
The entry values of the input array are scaled to be in the range of [0, 1]. The training batch size is set to be 32. We train our ResNet50 for 4,000 epochs. The training loss and accuracy with respect to epoch are given in Fig. 6. Similarly, the validation loss and accuracy with respect to epoch are displayed in Fig. 7. It can be seen that ResNet50 exhibits a strong ability to classify fire and non-fire images. The training accuracy reaches almost 1 after a few hundred epochs of training. The validation accuracy also increases rapidly in the first few hundreds of epochs. It then reaches a plateau and fluctuates around 0.9 as the training process continues.

B. Testing on Videos

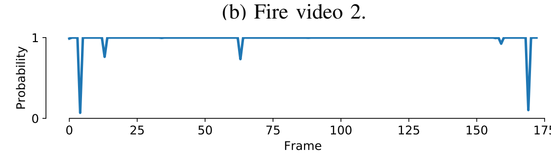
We downloaded three power line fire videos and three non-fire videos from YouTube. All the frames in the fire videos have fire. Similarly, all the frames in the non-fire videos do not have fire. Each video is around 10 seconds. The trained ResNet50 is used to classify each frame in the videos. A frame is classified as fire image once the corresponding output probability is above 0.5. The probabilities of frames in the fire videos and non-fire video 1 are shown in Fig. 8 and Fig. 9, respectively. All frames in non-fire video 2 and 3 are correctly classified as non-fire images. As shown in the figures, the trained ResNet50 achieves high classification accuracy in the testing videos. Note that all these six tests are out-of-sample tests. None of the frames in these six videos are included in the training dataset. There are a few false negatives in all three fire videos and false positives in non-fire video 1. In practice, a few scattered false negatives will not have a significant



(a) Fire video 1.



(b) Fire video 2.



(c) Fire video 3.

Fig. 8: Classification results of frames in the fire videos.

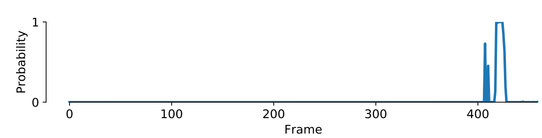
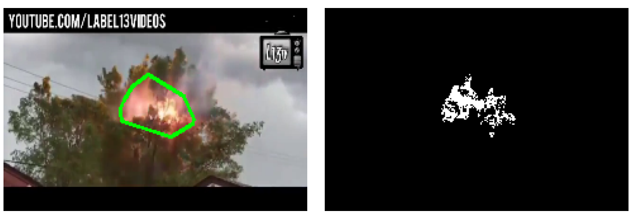


Fig. 9: Classification results of frames in the non-fire video 1.

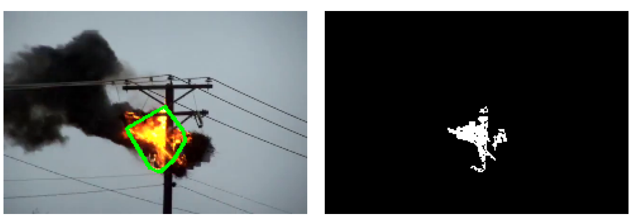
(a) False negative frame sample. (b) False positive frame sample.

Fig. 10: False negative frame sample in fire video 1 and false positive frame sample in non-fire video 1.

(a) Frame sample in fire video 1. (b) Fire region mask in fire video 1.



(c) Frame sample in fire video 2. (d) Fire region mask in fire video 2.



(e) Frame sample in fire video 3. (f) Fire region mask in fire video 3.

Fig. 11: Sample frames of fire videos and fire region masks.

impact. This is because fire can still be detected in a timely manner with videos having a frame rate between 20/s and 30/s. However, even a few false positives can be annoying in real-wold applications. Unfortunately, the proposed algorithm can not completely avoid false positives. Fig. 10 shows two frame samples that are classified incorrectly. The cause of the false negative sample on the left hand side is unclear. For the false positive sample, our ResNet50 possibly mistakenly identified the red lights of the fire truck as a real fire.

Fig. 11 shows three samples of fire region segmentation produced by our pixel-based fire detector. The gray images on the right display the fire region masks. The color images on the left are the corresponding original frames in the fire videos, where green outline boxes represent the convex hulls of the detected fire pixels. As illustrated in the figures, the fire regions can be effectively identified by our proposed pixel-based fire detector for all the testing fire videos.

IV. Conclusion and Future Work

This paper proposes a novel video fire detection framework for power line fire safety. It consists of a deep convolutional neural network (ResNet50) and a pixel-based fire detector. ResNet50 serves as a binary classifier to detect the presence of fire in a given video frame. The pixel-based fire detector then identifies the corresponding fire region in a fire video frame. The pixel-based fire detector consists of four major modules: color detection, motion detection, flicker detection, and spatial variation check. The pixels that pass the tests of all four modules are considered fire pixels, from which the fire location and scale can be estimated. Case study on real-world power line fire videos show that the proposed framework can effectively detect the presence of fire and locate the corresponding fire pixels in the testing fire videos.

In the future, we will investigate how to further reduce the false classification rate in our proposed automatic video fire detection framework. With the fast expansion of camera networks, we plan to develop an algorithm to select the optimal placement of new cameras.

References

[1] "CAL FIRE investigators determine cause of the camp fire," California Department of Forestry and Fire Protection, May 2019. [Online]. Available: http://calfire.ca.gov/

[2] A. E. Çetin, K. Dimitropoulos, B. Gouverneur, N. Grammalidis, O. Günay, Y. H. Habiboğlu, B. U. Töreyin, and S. Verstockt, "Video fire detection–review," *Digital Signal Processing*, vol. 23, no. 6, pp. 1827–1843, Dec. 2013.

[3] J. Sharma, O.-C. Granmo, M. Goodwin, and J. T. Fidje, "Deep convolutional neural networks for fire detection in images," in *International Conference on Engineering Applications of Neural Networks*, Aug. 2017, pp. 183–193.

[4] 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. 18 174–18 183, Mar. 2018.

[5] K. Muhammad, S. Khan, M. Elhoseny, S. H. Ahmed, and S. W. Baik, "Efficient fire detection for uncertain surveillance environment," *IEEE Transactions on Industrial Informatics*, Feb. 2019.

[6] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proceedings of the IEEE International Conference on Computer Vision*, Oct. 2017, pp. 2961–2969.

[7] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2016, pp. 770–778.

[8] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, Dec. 2014.

[9] T. Toulouse, L. Rossi, T. Celik, and M. Akhloufi, "Automatic fire pixel detection using image processing: a comparative analysis of rule-based and machine learning-based methods," *Signal, Image and Video Processing*, vol. 10, no. 4, pp. 647–654, Apr. 2016.

[10] Z. Zivkovic, "Improved adaptive Gaussian mixture model for background subtraction," in *Proceedings of the 17th International Conference on Pattern Recognition*, vol. 2. IEEE, Aug. 2004, pp. 28–31.

[11] Z. Zivkovic and F. Van Der Heijden, "Efficient adaptive density estimation per image pixel for the task of background subtraction," *Pattern Recognition Letters*, vol. 27, no. 7, pp. 773–780, May 2006.

[12] Z. Xu and J. Xu, "Automatic fire smoke detection based on image visual features," in *2007 International Conference on Computational Intelligence and Security Workshops*, Dec. 2007, pp. 316–319.

[13] B. W. Albers and A. K. Agrawal, "Schlieren analysis of an oscillating gas-jet diffusion flame," *Combustion and flame*, vol. 119, no. 1-2, pp. 84–94, Oct. 1999.

[14] B. U. Töreyin, Y. Dedeoğlu, U. Güdükbay, and A. E. Cetin, "Computer vision based method for real-time fire and flame detection," *Pattern Recognition Letters*, vol. 27, no. 1, pp. 49–58, Jan. 2006.

[15] G. Stockman and L. G. Shapiro, *Computer Vision*. Prentice Hall, 2001.