Unsupervised Method for Wildfire Flame Segmentation and Detection

EMIR BUZA AND AMILA AKAGIC, (Member, IEEE)
Faculty of Electrical Engineering, University of Sarajevo, 71000 Sarajevo, Bosnia and Herzegovina
Corresponding author: Emir Buza (ebuza@etf.unsa.ba)

ABSTRACT  In the last decade, there have been many reports on the negative impact of wildfires on various ecosystems. Unfortunately, wildfires have been intensifying as global temperatures, droughts, and other instances of extreme weather events rise around the world. These circumstances are forcing communities to vigorously address the uncontrolled spread of wildfires, where the ultimate goal is the protection of wildlife. At the same time, many disaster prevention and monitoring methods, based on image processing and computer vision, have been developed. In this paper, we present a new unsupervised method based on RGB color space for the early detection of wildfires from still images. From the analysis of existing state-of-the-art methods, it is evident that different methods explore different color spaces for the extraction of flame features. Our motivation was to use only RGB color space and thus eliminate the time-consuming task of color space conversion. The proposed method consists of several new image processing techniques used to efficiently extract flame features. It outperforms the existing methods, where an increase of 3% and 2% is recorded in the F1 score and Matthews correlation coefficient, respectively. Such performance demonstrates the merits of the proposed method for flame segmentation and detection.

INDEX TERMS  Image processing, feature extraction, object segmentation, object detection, unsupervised method.

I. INTRODUCTION
In the last 50 years, many reports have been published on how to understand, prevent and fight wildfires [1]–[3]. The phenomenon of wildfire is defined as a fire in an area of combustible vegetation occurring in rural areas. It is considered to be much more difficult to understand than controlled combustion, which has been extensively studied by different groups [4], [5]. Wildfires that spontaneously ignite and extinguish without any human intervention can have some beneficial effects on indigenous vegetation, animals, and ecosystems [6]. However, this only applies to smaller and localized wildfires. On the other hand, uncontrolled wildfires are difficult to control, hence it is important to early detect them and reduce the risk of their spreading.

The impact of wildfires on people and their property is the major driver for understanding and learning about wildfires [7]. There is no doubt that a wildfire poses a great threat to human health and causes material damage in almost all countries of the world [8]. There is practically no segment of human life where the impact of wildfires is not shown through air pollution and soil stripping. A combination of human activity and other factors can have catastrophic consequences for the ecosystem. These factors might include fuel availability, the physical environment, and weather conditions. In the last decade, many geographical regions are significantly more susceptible to the risk of larger regional wildfires than was the case one decade ago [9].

Fast and accurate detection of wildfires is one of the essential factors for reducing the risk of wildfires spreading. In recent years, image processing and machine learning-based algorithms have shown remarkable results for object detection using only two-dimensional data. When there are limited sources of data, it is important to design an efficient algorithm capable of accurately detecting flame pixels. This is challenging because detection is carried out with data from the visible spectrum under the conditions of changing parameters, such as background and brightness. The speed at which the algorithm is processing data also plays an important role since reliable information can identify the region under the fire faster. This valuable information can help firefighting forces react in a timely manner.

Traditional approaches for pixel-wise flame segmentation typically use two different data sources for processing, still
images and video sources. The still images are captured in different timing periods, with different background and brightness compositions. The analysis of these images requires a set of rules that are tailored to one or more color spaces. The rules are adjusted in accordance with the visual observations of objects to be classified. The methods that use still images are required to provide rules that can successfully detect foreground objects and eliminate all background noise. Other methods use video sources, which are recorded as a sequence of images with minor variations in the background. With this type of data, it is easier to detect small changes in the scene and thus eliminate the background pixels [10]. Eliminating the background is usually the first step in the process. Then, the extraction of flame pixels is carried out with a similar procedure as with still images. When the background noise is successfully removed, the method for extraction of flame pixel might require less restrictive rules. Some examples of color spaces that are used in the literature are RGB [10], YCbCr [11] [12], and CIELab [13].

Other methods combine different color spaces to increase the chance of detection of flame pixels, such as the methods that combine HSV, HSL, and HWB [14], and RGB and YCbCr [15] color spaces. Methods that extract information from still images are considered more challenging to design.

The main objective of this paper is to further the development of expert systems for the early detection of wildfire events. In that sense, this work proposes a new unsupervised method for efficient segmentation and detection of flame pixels in RGB color space. The benefit of processing images in RGB color space is that no color space conversion is required. The processing time for each image is thus significantly decreased. The proposed method is based on the efficient processing of still images, which means that a video source is not required to eliminate the background noise. All these characteristics make our algorithm suitable for the segmentation and detection of flame pixels in a fast manner.

The contribution of this study is summarized as follows.

1) We present an unsupervised method for flame segmentation and detection. The image transformation techniques are used to extract the flame features from the RGB color space. This color space is specifically chosen because it eliminates the need for conversion from one color space to another, which can be time-consuming. We also introduce several image processing techniques for the removal of background pixels that do not require the analysis of several video sources prior to processing, as is typically presented in state-of-the-art methods.

2) A comprehensive experiment has been conducted to evaluate the performance of six state-of-the-art methods and our method. Based on thirteen evaluation metrics it was determined that the proposed approach outperforms existing methods from the literature.

The method has the potential to provide information to firefighting ground forces in a timely manner, and significantly reduce the chances of wildfire spreading. It can be deployed on a cost-effective embedded platform, such as Rasberry Pi or a lightweight unmanned aerial vehicle with a cost-effective onboard computer. These devices can be installed in places where there is a high risk of wildfire occurring. The crucial benefit is that a wildfire can be detected based on a single image source.

The remainder of this article is organized as follows. Section II surveys the latest developments in wildfire segmentation and detection gathered and distilled from the literature survey. Section III provides details on the design of a new method for wildfire segmentation and detection. In Section IV, experimental results and evaluation criteria are presented. In Section V, conclusions are provided.

II. RELATED WORK

In this section, we review related literature and provide a brief overview of six state-of-the-art methods in the field of pixel-wise fire segmentation. The methods are presented in the corresponding subsection in the order of the year they were published. Each method is identified with the color spaces used for extraction. When two methods use the same color space, the surname of the first author is added. A short summary of methods is provided in Table 1.

A. RGB COLOR SPACE

In Celik et al. [10] (2007), a real-time video sequence-based method is proposed. The method uses a combination of information from the foreground object(s) with temporal changes. Changes in the scene are extracted with an adaptive background subtraction algorithm, which is verified by the statistical color model. Approximately 800 frames or 20s (40fps) of video content are required to extract the mean value of each color channel. Thus, the foreground object segmentation is assisted with a background subtraction algorithm. The changes in pixel values are detected using a change detection map. A statistical color model is created using 16,309,070 pixels from a set of 150 images. Six rules based on RGB color space are used, where the two first rules are the same as the rules previously proposed by Chen et al. [16]. The rules are defined in Eq. (1)-(6).

\[
R(x, y) > R_{\text{mean}} \\
R_{\text{mean}} = \frac{1}{N} \sum_{i=1}^{K} R(x_i, y_i) \\
R(x, y) > G(x, y) > B(x, y) \\
\tau_{C11} \leq \frac{G(x, y)}{R(x, y) + 1} \leq \tau_{C12} \\
\tau_{C21} \leq \frac{B(x, y)}{R(x, y) + 1} \leq \tau_{C22} \\
\tau_{C31} \leq \frac{B(x, y)}{G(x, y) + 1} \leq \tau_{C32} 
\]

Here, \(R(x, y), G(x, y),\) and \(B(x, y)\) are red, green, and blue values of pixels located at \((x, y)\), \(N\) is the total number of pixels in a given image, \(R_{\text{mean}}\) is the mean value of red channel, and thresholds \(\tau_{C11}, \tau_{C12}, \tau_{C21}, \tau_{C22}, \tau_{C31},\) and
TABLE 1. An overview of related work for wildfire segmentation and detection. Citation count is based on Google Scholar report from February 2022.

| Section | Title of the paper                                                                 | Year published | Color space used |
|---------|----------------------------------------------------------------------------------|----------------|------------------|
| II.A.   | “Fire detection using statistical color model in video sequences” [10]            | 2007           | RGB              |
| II.B.   | “Fire detection in video sequences using a generic color model” [11]             | 2009           | YCbCr            |
| II.C.   | “Fast and efficient method for fire detection using image processing” [13]        | 2010           | CIELab           |
| II.D.   | “Fire recognition using rgb and ycbcr color space” [15]                          | 2015           | RGB, YCbCr       |
| II.E.   | “Forest fire detection based on color spaces combination” [14]                  | 2018           | HSV, HISL, and HWB|
| II.F.   | “An early flame detection system based on image block threshold selection using knowledge of local and global feature analysis” [12] | 2020           | YCbCr            |

\[ \tau_{c_2} \] are upper and lower bounds which authors observed on the channel ratios on the dataset. Thresholds are determined based on every dataset used for analysis.

**B. YCbCr COLOR SPACE (CELIK et al.)**

In Celik et al. [11] (2009), a rule-based fire detection method in YCbCr color space from video sources is proposed. This method relies on the fact that the proposed rules for pixel classification operate in the color space that separates luma from chrominance components. The transformation from RGB to YCbCr color space is defined in Eq. (7).

\[
\begin{bmatrix}
    Y \\
    Cb \\
    Cr
\end{bmatrix} =
\begin{bmatrix}
    0.2568 & 0.5041 & 0.0979 \\
    -0.1482 & -0.2910 & 0.4392 \\
    0.4392 & -0.3678 & -0.0714
\end{bmatrix}
\begin{bmatrix}
    R \\
    G \\
    B
\end{bmatrix} + \begin{bmatrix}
    16 \\
    128 \\
    128
\end{bmatrix}
\] (7)

Here, \( Y, Cb \) and \( Cr \) denote the luminance, and two chrominance components (blue and red), respectively. The range of \( Y \) is [16, 235], whilst the range of two chrominance components is [16, 240]. The rules defined in Eq. (8)-(11) are then considered.

\[
Y_{(x,y)} > Cb_{(x,y)} \quad (8)
\]

\[
Cr_{(x,y)} > Cb_{(x,y)} \quad (9)
\]

\[
(Y_{(x,y)} > Y_{mean}) \cap (Cb_{(x,y)} < C_{mean}) \cap (Cr_{(x,y)} > C_{mean}) \quad (10)
\]

\[
|Cb_{(x,y)} - Cr_{(x,y)}| \geq \tau \quad (11)
\]

Here, \( Y_{mean}, C_{mean}, \) and \( Cr_{mean} \) denote the mean values of luminance, and two chrominance channels, respectively. They are calculated using the equations Eq. (12)-(14), where \( N \) denotes the total number of pixels in a given image.

\[
Y_{mean} = \frac{1}{N} \sum Y_{(x,y)} \quad (12)
\]

\[
Cb_{mean} = \frac{1}{N} \sum Cb_{(x,y)} \quad (13)
\]

\[
Cr_{mean} = \frac{1}{N} \sum Cr_{(x,y)} \quad (14)
\]

**C. CIELab COLOR SPACE**

In Celik et al. [13] (2010), a novel fire color model is presented with rules defined in CIELab color space. The method uses video sequence to construct reliable background information based on the binary frame difference map and the binary background difference map. First, RGB data is converted into CIELab color space as defined in Eq. (15)-(18).

\[
\begin{bmatrix}
    X \\
    Y \\
    Z
\end{bmatrix} =
\begin{bmatrix}
    0.412453 & 0.357580 & 0.180423 \\
    0.212671 & 0.715160 & 0.072169 \\
    0.019334 & 0.119193 & 0.950227
\end{bmatrix}
\begin{bmatrix}
    R \\
    G \\
    B
\end{bmatrix}
\] (15)

\[
L^* = \begin{cases}
166 \times (Y/Y_n)^{1/3} - 16, & \text{if } (Y/Y_n) > 0.008856 \\
903.3 \times (Y/Y_n), & \text{otherwise}
\end{cases}
\] (16)

\[
a^* = 500 \times (f(X/X_n) - f(Y/Y_n)) \quad (17)
\]

\[
b^* = 200 \times (f(Y/Y_n) - f(Z/Z_n)) \quad (18)
\]

The range of data in \( L^*, b^* \), and \( a^* \) components is \([0, 100], [−110, 110], [−110, 110]\), respectively. Next, a set of four rules is defined to detect flame pixels, as denoted in Eq. (19)-(22).

\[
L^*(x, y) \geq L^*_m \quad (19)
\]

\[
a^{*m}(x, y) \geq a^*_m \quad (20)
\]

\[
b^{*m}(x, y) \geq b^{*m} \quad (21)
\]

\[
b^*(x, y) \geq a^*(x, y) \quad (22)
\]

Here, \( L^*_m, a^{*m}, \) and \( b^{*m} \) represent collection of average values of the \( L^*, a^*, \) and \( b^* \) color channels, respectively.

Additionally, the authors considered using the fifth rule, which is based on analysis of histogram of fire pixels for each different color plane, namely \((L^* - a^*), (L^* - b^*),\) and \((a^* - b^*)\). However, in our experiments, we did not notice a big difference in the number of pixels detected as flame when this rule is used.

**D. RGB AND YCbCr COLOR SPACE**

In Zaidi et al. [15] (2015), the authors proposed a method that performs fire pixel detection based on seven rules defined in two color spaces, RGB and YCbCr. The rules are presented in Eq. (23)-(28). The method uses only still images.

\[
(R_{(x,y)} > G_{(x,y)}) \cap (G_{(x,y)} > B_{(x,y)}) \quad (23)
\]

\[
(R_{(x,y)} > R_{mean}) \cap (G_{(x,y)} > G_{mean}) \\
\cap (B_{(x,y)} < B_{mean}) \quad (24)
\]

\[
Y_{(x,y)} \geq Cb_{(x,y)} \quad (25)
\]
FIGURE 1. Block diagram of the proposed method with four stages.

\[
\begin{align*}
Cr(x,y) &\geq Cb(x,y) \\ (Y(x,y) \geq Y_{\text{mean}}) \land (Cb(x,y) \leq Cb_{\text{mean}}) \\
\land (Cr(x,y) \geq Cr_{\text{mean}}) \\
(Cb(x,y) \leq 120) \land (Cr(x,y) \geq 150)
\end{align*}
\] (26)

\[
\begin{align*}
F(x,y) = \begin{cases} 
1, & \text{if (} Y(x,y) > Y_{\text{mean}} \land (Cr(x,y) > Cb(x,y)) \text{)} \\
0, & \text{otherwise}
\end{cases}
\end{align*}
\] (30)

E. HSV, HSL, AND HWB COLOR SPACES

In Dzigal et al. [14] (2019), the authors combined three different color spaces, namely HSV, HSL, and HWB, to segment an image. Six rules are defined, as presented in Eq. (29). In addition to these rules, the authors presented several techniques for image pre-processing and post-processing to increase the segmented area. The method uses only still images.

\[
C = (S > 35) \\
\land ((H \geq 330 \land H \leq 65) \\
\land (L > 70) \\
\lor (B < B_{\text{avg}} \land W < W_{\text{avg}}) \\
\lor (W > W_{\text{avg}} \land S > S_{\text{avg}}) \\
\lor (W \geq 98 \land B \leq 2))
\] (29)

F. YCbCr COLOR SPACE (HSU et al.)

In Hsu et al. [12] (2020), a flame color analysis method is proposed based on modified rules presented in [11]. The authors observed that [11] performs well for low-intensity fires, where dominant colors are red and yellow. However, for high-intensity fires, when the dominant color is white, the fire detection method [11] does not perform as well. The aforementioned deficiency is addressed by processing pixels where \( Y_{(x,y)} > 220 \) differently than pixels where \( Y_{(x,y)} \leq 220 \). Pixels associated with low-intensity fire are processed in the exact same way as in the original method [11], while the high-intensity fire pixels, where \( Y_{(x,y)} > 220 \), are processed as defined in Eq. (30).

III. METHODOLOGY

A block diagram of the proposed unsupervised rule-based method for flame segmentation and detection using RGB color space is presented in Fig. 1. The method starts with extracting the region of interest from the original image, denoted as (A). Then, divides the original image into several smaller images (B), which are called stripes. The image is first divided into horizontal stripes, and then into vertical stripes. In these stages, several transformations are performed to extract fire regions. The first three stages are independent and can occur in parallel. After features are extracted from stripes, two resulting images are combined into one in the Seed Selection stage (C). In the last stage, several transformations are performed to create a bounding box over the flame region and provide detection capabilities (D). We describe each stage in the corresponding subsection.

A. REGION OF INTEREST

Fire has a set of complex properties, which are used to determine the presence of fire on the image. These properties might include the presence of flame and smoke, and each can have a different texture and color. When there is a fire, there is always a background residue that is created from a mixture of fire properties, i.e. reflection of the fire on the background. Sometimes, it is very difficult to distinguish fire from the background. In this paper, we focus on flame segmentation and detection, where we specifically try to distinguish the smoke from flame, which is very challenging at times. Thus,
in the context of this method, the Region of Interest (ROI) is viewed as the region that contains fire.

A decision about segmentation is typically based on a set of rules, which depend on different conditions. If rules are very strict, this can result in large parts of an image not being recognized as flame. On the other hand, less restrictive rules might not provide useful information for further processing. In our work, we strive to detect as many flame regions as possible, while at the same time keeping the algorithm's complexity low. It is a well-known fact that the complexity of the algorithm affects its performance. Hence, we use less restrictive rules to include larger flame regions, and in the following steps, we perform various low-complexity techniques to provide more precise flame regions.

A new algorithm for extraction of the region of interest is presented in Algorithm 1. The algorithm consists of a set of image transformations that are performed on the original image to extract the flame regions. We describe the steps below, while the results of individual steps are presented in Fig. 2. The algorithm requires four inputs: the original RGB image 

\[ I \]

and, geometric means for the \( R \), \( G \), and \( B \) components of the original image, denoted as \( \mu_{[R]} \), \( \mu_{[G]} \), and \( \mu_{[B]} \), respectively.

First, the RGB components of a new image \( I' \) are created by subtracting the geometric means from the original RGB components (line 1). A set of rules is defined to create a new mask denoted as \( \beta_{(x,y)} \). The rules are defined in Algorithm 1 in line 2, while an example of a result is illustrated in (b). The mask is then applied to the original image (line 2), while an example of a result is illustrated in (b). The mask is then applied to the original image to create a new \( \alpha_{(x,y)} \) image, as defined in line 4, and shown in (e). The new binary image \( \gamma_{(x,y)} \) was created by applying the Otsu's threshold \( \tau_{Otsu} \) as in Fig. 2(d) on \( \alpha_{(x,y)} \) image. Smaller regions consisting of less than 8 white pixels are removed from the binary image \( \gamma_{(x,y)} \) (line 6). Then, a layer of pixels is added to the inner and outer boundaries of regions to connect smaller regions (line 7). The goal of this operation is to eliminate small background regions (black) inside foreground regions (white), and also to decrease the gaps between regions.

### B. HORIZONTAL AND VERTICAL IMAGE STRIPES

The main goal of this stage is to extract features by separately exploring different regions of the original image \( I \). The idea is similar to the methods presented in [17], [18]. It allows us to easily eliminate the parts of the image that do not contain the flame. The result of this phase is the creation of a binary mask which is applied to the original image to select flame pixels.

At first, the original image \( I \) is divided into several smaller images, which are called stripes. The image is first divided horizontally and then vertically. However, the order is not important for the execution of the algorithm. The number of stripes is three per orientation, which is determined experimentally. An illustration of this process is provided in Fig. 1. The algorithm for creating stripes and extracting features is presented in Algorithm 2. All image transformations applied to the stripes are described in Eq. (32)-(37).

The stripe is created from the original image \( I \), and it is denoted as \( S^{(i)} \):

\[
S^{(i)} \equiv P^{(i)}(x,y)[R,G,B]
\]

where \( P^{(i)}(x,y)[R,G,B] \subseteq P(x,y)[R,G,B] \). \( P(x,y) \) is a pixel in the original image \( I \), while \( P^{(i)}(x,y)[R,G,B] \) is a pixel value of \( (x,y) \) in the sub-image \( S^{(i)} \), \( \forall i = [1, 2, \ldots, k] \).

Several thresholds are defined to help identify the flame pixels in the stripe. Namely, thresholds \( \tau_{1}^{(i)} \) and \( \tau_{2}^{(i)} \) which are extracted from the original image \( I \), and \( \tau_{1}^{(i)} \) and \( \tau_{2}^{(i)} \) which are extracted from stripes \( S^{(i)} \), where \( \forall i = [1, 2, \ldots, k] \). The algorithm for creating threshold values is described in Algorithm 3.

Additionally, three constants \( \mu^{(i)} \), \( \epsilon^{(i)} \), and \( \delta^{(i)} \) are defined in Eq. (32)-(34), respectively. The first constant, \( \mu^{(i)} \), is the mean value of the stripe. This value is used to calculate a modified standard deviation \( \epsilon^{(i)} \). The modified standard deviation

---

**Algorithm 1 Region of Interest**

**Require:** \( I, \mu_{[R]}, \mu_{[G]}, \) and \( \mu_{[B]} \) \( \triangleright \) Original input image \( I \).

1: \( I'_{(x,y)[R,G,B]} \leftarrow [I_{(x,y)[R]} - \mu_{[R]}, I_{(x,y)[G]} - \mu_{[G]}, I_{(x,y)[B]} - \mu_{[B]}] \)
2: \( \beta_{(x,y)} \leftarrow \begin{cases} 0, \quad & I'_{(x,y)[R]} < I'_{(x,y)[B]} \\ 0, \quad & I'_{(x,y)[R]} + I'_{(x,y)[B]} < I'_{(x,y)[G]} \\ 1, \quad & \text{otherwise} \end{cases} \)
3: \( \alpha_{(x,y)[R,G,B]} \leftarrow I_{(x,y)[R,G,B]} \cdot \beta_{(x,y)} \)
4: \( \alpha'_{(x,y)} \leftarrow 2 \cdot \alpha_{(x,y)[R]} - 2 \cdot \alpha_{(x,y)[B]} \)
5: \( \gamma_{(x,y)} \leftarrow \begin{cases} \alpha'_{(x,y)} < \tau_{Otsu} \\ 0, \quad & \gamma_{(x,y)} \geq \tau_{Otsu} \end{cases} \)
6: \( \hat{\gamma}_{(x,y)} \leftarrow \text{Remove small objects} \quad \triangleright \)
7: \( \hat{f}_{(x,y)} \leftarrow \text{Fill in small regions in the binary image} . \hat{\gamma} . \)
8: \( \text{return} \hat{f} \)
Algorithm 2 Horizontal and Vertical Image Stripes

Require: $I$ ➤ Original RGB image. $Point$ ➤ Number of sub-images that can be considered as rows and columns, respectively.

1: $[\tau_1^*, \tau_2^*] \leftarrow$ from Algorithm 3 ($I$) ➤ $\tau_1^*$ and $\tau_2^*$ are calculated with Algorithm 3 for image $I$.
2: $i \leftarrow 1$
3: for $l \leftarrow 1$; $l \leftarrow Point(1)$; $l \leftarrow 1$ do
4: for $j \leftarrow 1$; $j \leftarrow Point(2)$; $j \leftarrow 1$ do
5: $S^{(i)} \leftarrow StripedImage_{1,j}$ ➤ Striped image $S^{(i)}$ is calculated with Algorithm 3 for stripe image $S^{(i)}$.
6: $g_{(x,y)}^{(i)} \leftarrow Eq. (31)$. ➤ Binary image is calculated by using Eq. (29), (32), (33), (34), and (35).
7: $g_{(x,y)}^{(i)} \leftarrow Remove small objects ➤ Removes connected components with less than 8 pixels from the binary image $g_{(x,y)}^{(i)}$.
8: $h_{(x,y)}^{(i)} \leftarrow Fill in small regions in the binary image $g_{(x,y)}^{(i)}$. ➤ Set $R$, $G$, and $B$ pixels to zero of the image $S^{(i)}$ where binary image $h_{(x,y)}^{(i)} = 0$.
9: $i \leftarrow i + 1$
10: end for
11: end for

Algorithm 3 Calculation of Thresholds: $\tau_1$ and $\tau_2$

Require: $I^*$ ➤ Input image. 

Ensure: $I^*_{(x,y)[R]} \neq 0 \lor I^*_{(x,y)[G]} \neq 0 \lor I^*_{(x,y)[B]} \neq 0$ ➤ The input image must have three channels.

1: $[\nu[3], \nu[5]] \leftarrow histogram(I^*_{(x,y)[R]})$ ➤ histogram of pixels per $R$ level ($\nu[3]$) of the image $I^*$.
2: $[\nu[7], \nu[9]] \leftarrow histogram(I^*_{(x,y)[G]})$
3: $[\nu[1], \nu[2]] \leftarrow histogram(I^*_{(x,y)[B]})$
4: $\eta \leftarrow findAll(\nu[3] > \nu[7] \lor \nu[9] > \nu[7] \lor \nu[1] < \nu[2])$
5: $\tau_1 \leftarrow median(\eta)$
6: tmp $\leftarrow \eta(1)$
7: for $i \leftarrow 2$; $i \leftarrow sizeOf(\eta); i \leftarrow i + 1$ do
8: if $\eta(i) < tmp + 10$ then
9: tmp $\leftarrow \eta(i)$
10: else
11: $\tau_1 \leftarrow \eta(i)$
12: break
13: end if
14: if $i == sizeOf(\eta)$ then
15: if $\eta(i) > 100$ then
16: $\tau_1 \leftarrow \frac{\text{sizeOf}(\eta)}{\text{sizeOf}(\eta)} \text{sizeOf}(\eta)$
17: else
18: $\tau_1 \leftarrow \eta(i)$
19: end if
20: end if
21: end for
22: $\eta \leftarrow findAll(\nu[3] > \nu[7] \lor \nu[9] > \nu[7] \lor \nu[1] < \nu[2])$
23: $\tau_2 \leftarrow 0$
24: tmp $\leftarrow \eta(1)$
25: for $i \leftarrow 2$; $i \leftarrow sizeOf(\eta); i \leftarrow i + 1$ do
26: if $\eta(i) < tmp + 5$ then
27: tmp $\leftarrow \eta(i - 1)$
28: else
29: $\tau_2 \leftarrow \eta(i)$
30: break
31: end if
32: if $i == sizeOf(\eta)$ then
33: $\tau_2 \leftarrow \eta(i)$
34: end if
35: end for
36: $\tau_2 \leftarrow Eq. 37$. ➤ Recalculation of $\tau_2$ using Eq. (37).
37: return $[\tau_1, \tau_2]$

Algorithm 1 Calculation of Histograms $\mu^{(i)}$, $\epsilon^{(i)}$, and $\delta^{(i)}$

Require: $I^{(i)}$ ➤ Image of size $N \cdot M$. 

Ensure: $\mu^{(i)}$, $\epsilon^{(i)}$, and $\delta^{(i)}$ ➤ Calculated rules for finding $\tau_1$ and $\tau_2$.

1: $\mu^{(i)} = \frac{1}{N \cdot M} \sum_{x,y} p^{(i)}_{(x,y)[gray]}$
2: $\epsilon^{(i)} = \frac{255}{\sqrt{N - M - 1}} \sum_{x,y} p^{(i)}_{(x,y)[gray]} - \mu^{(i)}$²
3: $\delta^{(i)} = \begin{cases} \frac{\tau_1^*/(\tau_2^* \cdot 2),}{\epsilon^{(i)} < \tau_1^*/(\tau_2^* \cdot 2),} \\ \text{otherwise} \end{cases}$

The value $\tau_1$ in Eq. (35) is defined based on the value of $\tau_1^{(a)}$, which is also used in rules for creating a new binary image $g_{(x,y)}^{(i)}$ defined in Eq. (36). The binary image $g_{(x,y)}^{(i)}$ is defined with a set of rules, including individual RGB color information from stripes, and $\tau_1$ and $\delta$.

$t_1^{(i)} = \begin{cases} \tau_1^{(a)}, & \tau_1 > \tau_1^{(i)} \\ \tau_1^{(i)}, & \text{otherwise} \end{cases}$

After the binary image $g_{(x,y)}^{(i)}$ is created, the small connected components are removed from the binary image. These components are typically less than 8 pixels in size (line 8 in Algorithm 2). Additionally, a layer of pixels is added to the
inner and outer boundaries of regions to connect smaller regions (line 9). Then, the created mask is applied to each stripe (line 10), and a new RGB image $R_{[X,Y],[R,G,B]}$ for each stripe orientation is created. The resulting images are denoted as $R^{(h)}$ and $R^{(v)}$ to represent horizontal and vertical stripe images, respectively. An example of the original image, horizontal and vertical image stripes is shown in Fig. 3.

The threshold $\tau_2$ used in Algorithm 3 is defined in Eq. (37). The threshold is quantized to increase the difference between the flame and surrounding pixels, i.e. smoke. This enabled more control in the selection of flame pixels. The impact of using $\tau_1$ and $\tau_2$ is best demonstrated in Fig. 4, where the original image is provided in (a). The image contains a small region of flame and a large region of smoke. Here, the goal is to select only the flame region and avoid the smoke region. In (b) and (c) the region without tuning (without thresholds) is shown, which results in the smaller flame region being selected. On the other hand, in (d) and (e), it is observable that a bigger flame region is selected with thresholds. Details on how to create a bounding box around a set of flame pixels, as shown in Fig. 4, are provided in the Flame Detection stage.

$$\tau_2 = \begin{cases} 40, & \tau_2 < 3 \\ \tau_2 \cdot 10, & \tau_2 < 5 \\ \tau_2 \cdot 5, & \tau_2 < 10 \\ 5, & \text{otherwise} \end{cases}$$  (37)

**C. SEED SELECTION**

In this stage, a set of simple operations is applied to combine the two resulting RGB images from the previous stage, $R^{(h)}$ and $R^{(v)}$. In Algorithm 4, we describe all necessary image transformations. The resulting image $f_{(i,j)}$ is created using only the R channel from the $R^{(h)}$ and $R^{(v)}$ images. The R channel is used because the red color is the dominant color
The creation of the mask involves a set of checks to determine the value of each pixel in the R channel for $R^{(h)}$ and $R^{(v)}$ images (lines 5-14). Once the checks are performed, two resulting images are combined with a simple AND operation (line 15). The resulting image is a binary image. An example result is shown in Fig. 5.

**D. FLAME DETECTION**

In the flame detection stage, two binary images from the Region of Interest and Seed Selection are combined with a special technique to create the final image, as is illustrated in Fig. 1. A number of bounding boxes are drawn around a set of connected pixels before the images are combined. When a pixel is not connected to any other pixel(s), a bounding box is also drawn around it. An example of this process is provided in Fig. 6. in (a) and (b).

Next, an intersection is created between images (a) and (b), and the resulting image is shown in (c). In this process, there are three possible scenarios as illustrated in Fig. 7. The Region of Interest is shown by the blue rectangle and Seed Selection by the red rectangle. The **Case 1** represents the situation where the region of interest contains all seed selection pixels. The **Case 2** shows the region of interest intersecting with a small part of the seed selection. And finally, **Case 3** shows a situation when the seed selection completely overlays the region of interest. When there is an intersection between these two regions, the blue region is always selected. There is also a special case when the Algorithm 4 returns no seeds in the resulting image. In this case, there is no intersection between the blue and red rectangles, thus the blue rectangle is selected.

The resulting image is shown in Fig. 6. (d) where the flame region is indicated with the bounding box.

**IV. EVALUATION RESULTS AND DISCUSSION**

**A. DATASET**

The Corsican Fire DataBase (CFDB) [19] is a dataset with wildfire images and sequences of images acquired in areas of visible and near-infrared spectrums in different outdoor conditions (vegetation, climate, distance, brightness, etc). The dataset is available\textsuperscript{1} to the scientific community working on image processing and computer vision tasks related to the detection of different fire scenarios. All images are created outdoors. Images within the dataset have a range of different resolutions, from 183 $\times$ 242 pixels to 4000 $\times$ 3000 pixels. The dataset consists of a large variety of images of real-life situations. The dataset has images that are taken in heterogeneous environments such as forests, rocks, and snow. Images have different characteristics depending on the different conditions they have been captured in, such as time of day (day, night), and forecast (sunny, cloudy, gray sky, blue sky).

Experts manually segmented the flame areas in each image. Images are categorized into three classes according to the dominant color of fire pixels: white - yellow, orange, and red. Analysis of pixels characterized by a dominant class of fire color is presented in Table 2. In addition to 500 RGB images, the dataset also contains 100 near-infrared images (NIR), and sequences of 5 video sources in RGB and NIR (540 images each). There are a total of 231,693,736 of flame pixels (23.01%), and 775,071,092 of non-flame pixels (76.99%). In total, the dataset consist of 1,006,764,828 pixels.

The method proposed in this paper is tested on a set of 500 RGB images. The dataset is described in detail in [20], and it is publically available.\textsuperscript{2}

**B. EVALUATION METRICS**

The proposed method is evaluated on all pixels in the datasets (1,006,764,824 pixels). The binary pixel classification performance is reported in terms of the number of true positive (TP), true positive (TN), false-positive (FP), and

\begin{table}[h]
\centering
\begin{tabular}{|c|c|c|}
\hline
\textbf{Dataset} & \textbf{No. of Pixels} & \textbf{Percent [%]} \\
\hline
Flame Pixels & 231 693 736 & 23.01\% \\
Non-flame Pixels & 775 071 092 & 76.99\% \\
\hline
Total: & 1 006 764 828 & 100.00\% \\
\hline
\end{tabular}
\caption{Distribution of dominant fire pixel color in the Corsican fire database.}
\end{table}

\textsuperscript{1}The dataset is available upon a request at http://cfdb.univ-corse.fr/index.php.

\textsuperscript{2}Available upon a request at http://cfdb.univcorse.fr/index.php.

...
false-negative (FN) pixels. Based on these values, thirteen different metrics are used to compare our results with different state-of-the-art methods. The metrics are presented in Eq. (38)-(50).

- **Accuracy**
  \[
  ACC = \frac{TP + TN}{P + N}
  \]  
  \(38\)

- **Misclassification Rate**
  \[
  MR = \frac{FP + FN}{P + N} = 1 - ACC
  \]  
  \(39\)

- **Sensitivity or Recall, or True Positive Rate**
  \[
  TPR = \frac{TP}{N} = \frac{TP}{TP + FN}
  \]  
  \(40\)

- **Specificity or True Negative Rate**
  \[
  TNR = \frac{TN}{N} = \frac{TN}{TN + FP}
  \]  
  \(41\)

- **Precision or Positive Predictive Value**
  \[
  PPV = \frac{TP}{P'} = \frac{TP}{TP + FP}
  \]  
  \(42\)

- **Negative Predictive Value**
  \[
  NPV = \frac{TN'}{N'} = \frac{TN}{TN + FN}
  \]  
  \(43\)

- **Fall-out or False Positive Rate**
  \[
  FPR = \frac{FP}{N} = \frac{FP}{FP + TN} = 1 - TNR
  \]  
  \(44\)

- **False Negative Rate**
  \[
  FNR = \frac{FN}{P} = \frac{FN}{TP + FN} = 1 - TPR
  \]  
  \(45\)

- **False Discovery Rate**
  \[
  FDR = 1 - \frac{TP}{TP + FP} = 1 - PPV
  \]  
  \(46\)

- **False Omission Rate**
  \[
  FOR = 1 - \frac{TN}{TN + FN} = 1 - NPV
  \]  
  \(47\)

- **F1 Score**
  \[
  F1 = 2 \cdot \frac{PPV \cdot TPR}{PPV + TPR} = 2 \cdot \frac{TP}{TP + FP} \cdot \frac{TP}{TP + FN}
  \]  
  \(48\)

- **Balanced Accuracy**
  \[
  BACC = \frac{TPR + TNR}{2}
  \]  
  \(49\)

- **Matthews Correlation Coefficient**
  \[
  MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP) \cdot (TP + FN) \cdot (TN + FP) \cdot (TN + FN)}}
  \]  
  \(50\)

### C. DISCUSSION OF RESULTS

Results of the state-of-the-art methods and our method are presented in Table 3. We present the analysis and results for thirteen evaluation metrics. From the analysis, it is evident that some metrics should not be observed in isolation, but rather analyzed in pairs as is common practice in the literature. Examples of metric pairs are (a) sensitivity (recall) and specificity, and (b) sensitivity and precision. Results indicate that the proposed method outperforms the state-of-the-art methods in several evaluation metrics. In the following section, we analyze and discuss all presented metrics and provide some insights into the trade-off between them.

The accuracy of the proposed method is 93%. This metric shows the total number of accurate predictions, including true positive and true negative predictions. The same result was obtained with Dzigal et al. [14]. This indicates that approximately 936,291,290 out of 1,006,764,828 pixels are correctly classified. Other methods have lower results with only two having above 90% accuracy, while the rest achieve much lower results. The proposed method also has the best result in the Misclassification Rate (MR), which indicates the fraction of the predictions that are misclassified without distinguishing between positive and negative predictions. The method presented by Dzigal et al. [14] also has the same result.

The sensitivity and specificity metrics provide a better insight into how many positives and negatives are classified
TABLE 3. Evaluation results of thirteen different evaluation metrics on the six state-of-the-art methods and our method.

| # | Metrics                        | Celik et al. (2010) | Celik et al. YCrCb (2013) | Celik et al. YCrCb (2013) | Zaidi et al. RGB (2013) | Zaidi et al. YCrCb (2013) | Dzigal et al. YCrCb (2014) | Har et al. YCrCb (2012) | Proposed Method RGB |
|---|-------------------------------|---------------------|---------------------------|---------------------------|------------------------|---------------------------|---------------------------|------------------------|---------------------|
| #1 | Accuracy (ACC)                | 0.84                | 0.91                      | 0.81                      | 0.86                   | 0.93                      | 0.91                      | 0.91                   | 0.93                |
| #2 | Misclassification Rate (MCR)  | 0.16                | 0.09                      | 0.19                      | 0.14                   | 0.07                      | 0.09                      | 0.07                   | 0.31                |
| #3 | Sensitivity or Recall (TPR)   | 0.24                | 0.6                       | 0.37                      | 0.31                   | 0.79                      | 0.64                      | 0.86                   | 0.31                |
| #4 | Specificity or True Negative Rate (TNR) | 0.93 | 0.86 | 0.91 | 0.92 | 0.82 | 0.85 | 0.81 | 0.86 |
| #5 | Positive Predictive Value or Precision (PPV) | 0.9 | 0.9 | 0.63 | 0.93 | 0.95 | 0.89 | 0.86 | 0.83 |
| #6 | Negative Predictive Value (NPV) | 0.83 | 0.9 | 0.85 | 0.84 | 0.94 | 0.91 | 0.95 | 0.95 |
| #7 | Fall-out or False Positive Rate (FPR) | 0.07 | 0.14 | 0.09 | 0.08 | 0.18 | 0.15 | 0.19 | 0.19 |
| #8 | False Negative Rate (FNR)     | 0.76                | 0.4                       | 0.63                      | 0.69                   | 0.21                      | 0.36                      | 0.14                   | 0.36                |
| #9 | False Discovery Rate (FDR)    | 0.1                 | 0.1                       | 0.37                      | 0.37                   | 0.79                      | 0.64                      | 0.86                   | 0.86                |
| #10 | False Omission Rate (FOR)     | 0.24                | 0.6                       | 0.37                      | 0.37                   | 0.79                      | 0.64                      | 0.86                   | 0.86                |
| #11 | PI Score (F1)                 | 0.35                | 0.69                      | 0.4                       | 0.44                   | 0.79                      | 0.7                       | 0.82                   | 0.82                |
| #12 | Balanced Accuracy (BACC)      | 0.62                | 0.79                      | 0.66                      | 0.65                   | 0.88                      | 0.81                      | 0.90                   | 0.90                |
| #13 | Matthews Correlation Coefficient (MCC) | 0.38 | 0.67 | 0.37 | 0.47 | 0.76 | 0.68 | 0.78 | 0.78 |

accurately. The sensitivity refers to the probability of a positive test, indicating the number of flame pixels that are accurately classified. From the results, our method has the highest result of 86%. This means that approximately 199,256,612 out of 231,693,736 flame pixels are accurately classified. The next best score is achieved by Dzigal et al. [14] with 79%, while other methods achieved modest results with the lowest score of 24% for Celik et al. [10]. The specificity refers to the probability of a negative test, indicating the number of non-flame pixels that are accurately classified. Since there are much more non-flame pixels than flame pixels (i.e. 76.99% vs 23.01%), the misclassification can lead to a significant decrease in the overall accuracy. The specificity of our method is 81%, which indicates that approximately 627,807,584 out of 775,071,092 non-flame pixels are accurately classified. The best result for specificity is for Celik et al. [10], where the score is 93%.

In practice, sensitivity and specificity are rarely considered in isolation. An ideal system should have high sensitivity and specificity, thus it is common practice to evaluate the trade-off between the metrics. If the difference is large, the method does not have good segmentation performance at all. The results show that the difference is the smallest in Dzigal et al. [14], where the difference is only 3%. For our method, the sensitivity is 86% and specificity is 81%, which indicates a 5% difference. However, our method results in greater overall sensitivity and the difference between sensitivity and specificity are not as large as with methods such as Celik et al. [10] (RGB), Celik et al. [13] (CIELab), or Zaidi et al. [15] (YCrCb), where the difference is 69%, 54%, and 61%, respectively.

The Positive Predictive Value (PPV) or precision is the ratio between the true positives and the sum of all positives (true positive and false positive). In the context of the problem we are addressing, this includes the number of accurately classified flame and non-flame pixels. Essentially, the precision indicates how many times an accurate prediction of a specific class occurs per a false prediction of the same class. This metric is usually related to the sensitivity or recall metric, where the tradeoff is analyzed. The tradeoff analysis is important where an imbalance between the number of samples is present. The ideal system should have large values for both metrics. The same is true for sensitivity and specificity. The precision of our model is 83%, and the sensitivity is 86%. There is a 3% difference between these two values, which is the smallest difference overall. The best result for precision is 95% for Zaidi et al. [15]. However, the sensitivity of this method is only 31%, which indicates that many flame pixels are incorrectly classified as non-flame pixels.

The Negative Predictive Value (NPV) is the ratio between the true negatives and the sum of all negatives (true negative and false negative). It is essentially a probability that a negative test result is accurately predicted. Our method has the result of 95%, which means that majority of non-flame pixels are accurately predicted as non-flame pixels. This is also the highest result when compared to other methods. The next best result of 94% is obtained by Dzigal et al. [14].
The next four metrics are Fall Positive Rate (FPR), False Negative Rate (FPR), False Discovery Rate (FDR), and False Omission Rate (FOR), which are related to TNR, TRP, PPV, and NPV, respectively. Once TNR, TRP, PPV, and NPV are calculated, the results for FPR, FPR, FDR, and FOR can be calculated. The sum of a specific pair of metrics should be equal to 1.

The F1 score (F1) is the harmonic mean of precision and recall. Its highest possible value is 1, which indicates perfect precision and recall. It is also known as the Sørensen-Dice coefficient or Dice similarity coefficient (DSC). The result of our method is 82%, which is the highest result among all tested methods. The following best algorithm is Dzigal et al. [14]. The Balanced Accuracy (BACC) is a metric used for evaluating how good a binary classifier is. It is especially useful when the classes are imbalanced. It represents a mean between sensitivity and specificity. The result of our method is 90%, which is the highest result among all tested methods.

The Matthews correlation coefficient (MCC) is a more reliable statistical rate which has a high score only if the prediction has good results in all of the confusion matrix categories (true positives, false negatives, true negatives, and false positives), proportionally both to the size of positive elements and the size of negative elements in the dataset [21]. The MCC score is considered more informative in evaluating binary classifications than accuracy and F1 score, which are the most often used in literature. From the results, it is evident that all methods have much lower results in MCC than the Accuracy, while the MCC score in some methods is higher than the F1 score. Overall, our method has the highest score of 78%. The next best result of 76% is again obtained by Dzigal et al. [14].

There is one previous work on image segmentation that is based only on RGB color space, and one that uses a combination of RGB and YCbCr color spaces, described in [10] and [15], respectively. In [10], the authors propose a method based on the analysis of video sequences. The authors use a background subtraction algorithm to detect changes in the scene. This method tries to distinguish between static and dynamic objects; hence, it can successfully remove the background objects, which helps in localizing foreground objects in the image. Once background objects are removed, the authors define a set of rules to select the remaining pixels as flame pixels. The rules are described in Eq. (1)-(6). In [15], the authors use additional color space (YCbCr) with RGB to define a set of six rules to select flame pixels in still images, as described in Eq. (23)-(28). These works are different than ours in the following terms. The algorithm in [10] requires video sources to extract common features, while [15] requires color space conversion from RGB to YCbCr of each image before performing image segmentation. In Table 3., we provide a comprehensive evaluation of six state-of-the-art methods and our method, where we demonstrate that our method achieves better results in terms of accuracy, misclassification rate, sensitivity, specificity, F1, balanced accuracy, Matthews correlation coefficient, and others than the state-of-the-art. From all metrics, F1 and MCC are considered the most relevant for image segmentation. In Table 3., the F1 score of our method is 0.82, and MCC is 0.78. The method described in [10] has an F1 score of 0.35, and MCC is 0.38. This means that our method has a 47% better F1 score and a 40% better MCC value. The method described in [15] has an F1 score of 0.44, and MCC is 0.47. This means our method has a 38% better F1 score and a 31% better MCC coefficient value. These metrics indicate that our method works equally well in finding flame and non-flame pixels in still images, which cannot be said for the other two algorithms that use RGB color space.

The method described in [10] differs greatly from our results as has been presented in Table 3. Specifically, our method has better results in terms of ACC, MR, TPR, NPV,
FNR, FOR, F1, BACC, and MCC. Some evaluation metrics cannot be considered in isolation when comparing two or more methods. These metrics are (a) sensitivity and specificity, and (b) sensitivity and precision. In literature, the trade-off between these two metrics is commonly explored. Both metrics in a pair should achieve similar results to consider them relevant. For example, in the method [10] sensitivity is 0.24 and specificity is 0.93, while in our method sensitivity is 0.86 and specificity is 0.81. There is a significant difference in results, where a smaller difference means better performance. In [15], the authors use RGB and YCrCb colors spaces on still images, while our method uses only RGB color space on still images. For method [15], sensitivity is 0.31 and specificity is 0.92, while in our method sensitivity is 0.86 and specificity is 0.81. These results indicate that our method achieves much better results.

It is important to note that the state-of-the-art methods rely on feature extraction from video sources, while our method is not dependent on video sources and can work on both, video sources and still images. Typically, video sources are used to extract some common features of the fire, which can easily be identified by the method of subtracting two consecutive frames. In this case, the background is typically static, while the fire has dynamic characteristics that change from frame to frame. This preprocessing step is used in the majority of the state-of-the-art methods. It significantly reduces the background noise and helps in extracting the fire features. The proposed method is not dependent on the existence of video sources and can extract features from still images with satisfactory results.

Some examples of results of the proposed method are displayed in Fig. 8.

V. CONCLUSION

This paper presents an unsupervised method for wildfire flame segmentation and detection. The method consists of several new image processing techniques that can efficiently extract flame features from still images. We first extract the region of interest with less restrictive rules, where the goal is to discover as many flame regions as possible. Then, we explore different regions of the image, which we call horizontal and vertical image stripes. The stripes are created by dividing the original image into three stripes in two different orientations. Features are extracted from each stripe, where the end result is combined in a single RGB image per orientation. We combine two resulting RGB images with different image transformations, where only the R channel is considered since the red color is a dominant color of flame. The resulting image is the black and white image or mask. In the flame detection step, the resulting images from the region of interest and seed selection are used to detect multiple flame regions.

The method is evaluated with thirteen evaluation metrics together with the six state-of-the-art methods. The proposed method outperforms the state-of-the-art methods in nine evaluation metrics. Additionally, we explore the trade-off between two pairs of metrics (a) sensitivity and specificity, and (b) sensitivity and precision. This analysis provides a better insight into the performance of the method. Good segmentation performance is indicated if the difference between metrics is small. Our method has the highest sensitivity value among evaluated methods, where an increase of 7% is recorded compared to the state-of-the-art. The sensitivity of our method is 86%, while the specificity is 81%. The precision of our method is 83%, which indicates the difference of only 3% with respect to sensitivity. This difference is the smallest difference overall, while other methods have significantly larger differences. The proposed method also achieves the highest results for the F1 score and Matthews correlation coefficient, where an increase of 3% and 2% has been recorded, respectively. In recent literature, the Matthews correlation coefficient is considered more informative in evaluating binary classification, hence such performance demonstrates the merits of the proposed method for wildfire flame segmentation and detection.

There are two important advantages of our methods in relation to other methods. The first advantage is that the proposed method can efficiently extract flame features from RGB color space, which eliminates the need for color conversion. The second advantage is that the method can efficiently segment and detect multiple flame regions from a single image. Therefore, the analysis of several pre-existing videos for extraction of features is not necessary, which is typically a part of the majority of the state-of-the-art methods.

REFERENCES

[1] C. S. Stevens-Rumann, K. B. Kemp, P. E. Higuera, B. J. Harvey, M. T. Rother, D. C. Donato, P. Morgan, and T. T. Veblen, “Evidence for declining forest resilience to wildfires under climate change,” *Ecol. Lett.*, vol. 21, no. 2, pp. 243–252, Feb. 2018.

[2] J. San-Miguel-Ayanz, E. Schulte, G. Schmuck, A. Camia, P. Strobl, G. Liberta, C. Giovando, R. Boca, F. Sedano, P. Kempe, D. McInerney, C. Witherow, S. S. de Oliveira, M. Rodrigues, T. Durrant, P. Corti, F. Oehler, L. Vilar, and G. Amatallia, “Comprehensive monitoring of wildfires in Europe: The European forest fire information system (EFFIS),” in *Approaches to Managing Disaster: Assessing Hazards, Emergencies and Disaster Impacts*. London, U.K.: IntechOpen, 2012.

[3] J. G. Quintiere, *Principles of Fire Behavior*. Boca Raton, FL, USA: CRC Press, 2016.

[4] A. Valavanidis, N. Iliopoulos, G. Gotsis, and K. Fiotakis, “Persistent free radicals, heavy metals and PAHs generated in particulate soot emissions and residue ash from controlled combustion of common types of plastic,” *J. Hazardous Mater.*, vol. 156, nos. 1–3, pp. 277–284, Aug. 2008.

[5] K. McGrattan, R. McDermott, J. Floyd, S. Hostikka, G. Forney, and H. Baum, “Computational fluid dynamics modelling of fire,” *Int. J. Comut. Fluid Dyn.*, vol. 26, nos. 6–8, pp. 349–361, Jul. 2012.

[6] H. Heidari, M. Arabi, and T. Warziniack, “Effects of climate change on natural-caused fire activity in western U.S. National Forests,” *J. Hazardous Mater.*, vol. 26, nos. 6–8, pp. 349–361, Jul. 2012.

[7] J. C. Wilczak, D. C. Donato, P. Morgan, and T. T. Veblen, “Evidence for declining forest resilience to wildfires under climate change,” *Ecol. Lett.*, vol. 21, no. 2, pp. 243–252, Feb. 2018.

[8] S. E. Finlay, A. Moffat, R. Gazzard, D. Baker, and V. Murray, “Health impacts of wildfires,” *PLoS Currents*, vol. 4, Nov. 2012.

[9] S. E. Finlay, A. Moffat, R. Gazzard, D. Baker, and V. Murray, “Health impacts of wildfires,” *PLoS Currents*, vol. 4, Nov. 2012.

[10] S. E. Finlay, A. Moffat, R. Gazzard, D. Baker, and V. Murray, “Health impacts of wildfires,” *PLoS Currents*, vol. 4, Nov. 2012.
EMIR BUZA received the bachelor’s, master’s, and Ph.D. degrees from the Faculty of Electrical Engineering, University of Sarajevo, in 2002, 2009, and 2014, respectively. He is now an Associate Professor with the Department of Computer Science and Informatics, University of Sarajevo. He authored and coauthored numerous papers and two books on the topic of databases. His research interest includes data mining, including databases, big data, and bioinformatics. His Ph.D. research focused on developing methodologies in machine learning that help the classification of biological data, like protein sequences, and microarrays data. He has expanded his research to include image segmentation; and computer vision.

AMILA AKAGIC (Member, IEEE) received the bachelor’s and master’s degrees from the Faculty of Electrical Engineering, University of Sarajevo, in 2006 and 2009, respectively, and the Ph.D. degree from Keio University, Japan, in 2013. She is now an Associate Professor with the Department of Computer Science and Informatics, University of Sarajevo. Her past research mainly focused on finding new ways to accelerate compute-intensive parts of an algorithm by means of offloading them to an FPGA. Her current research interest includes computer architecture, including reconfigurable architectures, high-performance computing, and heterogeneous computing. She has expanded her research to include artificial intelligence, computer vision, image segmentation, machine learning, deep learning, and digital signal processing.

* * *

E. Buza, A. Akagic: Unsupervised Method for Wildfire Flame Segmentation and Detection