NRC Research Press

ARTICLE

Forest fire flame and smoke detection from UAV-captured images using fire-specific color features and multi-color space local binary pattern[1]

F.M. Anim Hossain, Youmin M. Zhang, and Masuda Akter Tonima

Abstract: In recent years, the frequency and severity of forest fire occurrence have increased, compelling the research communities to actively search for early forest fire detection and suppression methods. Remote sensing using computer vision techniques can provide early detection from a large field of view along with providing additional information such as location and severity of the fire. Over the last few years, the feasibility of forest fire detection by combining computer vision and aerial platforms such as manned and unmanned aerial vehicles, especially low cost and small-size unmanned aerial vehicles, have been experimented with and have shown promise by providing detection, geolocation, and fire characteristic information. This paper adds to the existing research by proposing a novel method of detecting forest fire using color and multi-color space local binary pattern of both flame and smoke signatures and a single artificial neural network. The training and evaluation images in this paper have been mostly obtained from aerial platforms with challenging circumstances such as minuscule flame pixels, varying illumination and range, complex backgrounds, occluded flame and smoke regions, and smoke blending into the background. The proposed method has achieved *F*1 scores of 0.84 for flame and 0.90 for smoke while maintaining a processing speed of 19 frames per second. It has outperformed support vector machine, random forest, Bayesian classifiers and YOLOv3, and has demonstrated the capability of detecting challenging flame and smoke regions of a wide range of sizes, colors, textures, and opacity.

Key words: forest fire detection, unmanned aerial vehicles, artificial neural network, flame and smoke detection, remote sensing.

Résumé : Au cours des dernières années, la fréquence et la gravité des incendies de forêt ont augmenté, ce qui a incité les communautés de recherche à examiner activement des méthodes précoces de détection et d'extinction des incendies de forêt. La télédétection utilisant des techniques de vision par ordinateur peut fournir une détection précoce à partir d'un vaste champ observé et fournir des informations supplémentaires telles que l'emplacement et la gravité de l'incendie. Au cours des dernières années, la faisabilité de la détection des incendies de forêt, en combinant la vision par ordinateur et les plateformes aériennes comme les véhicules aériens pilotés et sans pilote, en particulier les véhicules aériens sans pilote de petite taille et à faible coût, a été mise à l'essai et s'est révélée prometteuse en fournissant des renseignements sur la détection, la géolocalisation et les caractéristiques

Received 31 January 2020. Accepted 30 June 2020.

F.M.A. Hossain, Y.M. Zhang, and M.A. Tonima. Department of Mechanical, Industrial and Aerospace Engineering, Concordia University, Montréal, QC H3G 1M8, Canada.
Corresponding author: Youmin Zhang (e-mail: ymzhang@encs.concordia.ca).
[1]This article is part of a virtual issue on innovative applications of unmanned aerial vehicles in forest science and management jointly published by the *Journal of Unmanned Vehicle Systems* and the *Canadian Journal of Forest Research*.

de l'incendie. Ce document ajoute à la recherche existante en proposant une nouvelle méthode de détection d'incendies de forêt en utilisant la couleur et la configuration binaire locale de l'espace multi-couleur des signatures de flammes et de fumée et un seul réseau de neurones artificiels. Les images d'entrainement et d'évaluation présentées dans ce document ont été principalement obtenues à partir de plates-formes aériennes présentant des circonstances difficiles, telles que des pixels de flamme minuscules, une luminosité et une portée variables, des arrière-plans complexes, des zones de fumée et de flamme occluses et de la fumée qui se mélange à l'arrière-plan. La méthode proposée a obtenu des scores *F*1 de 0,84 pour la flamme et de 0,90 pour la fumée tout en maintenant une vitesse de traitement de 19 images par seconde. Elle a donné de meilleurs résultats que la machine à vecteurs de support, la forêt aléatoire, les classificateurs bayésiens et YOLOv3 et a démontré la capacité de détecter les zones de flammes et de fumée difficiles d'un large éventail de tailles, de couleurs, de textures et d'opacité. [Traduit par la Rédaction]

Mots-clés : détection des incendies de forêt, véhicules aériens sans pilote, réseau de neurones artificiels, détection des flammes et de la fumée, télédétection.

1. Introduction

Forests play a major role in the survival of all living beings on this planet by providing food, shelter, and medicine. They absorb harmful greenhouse gasses to mitigate climate change, keep the soil safe from degradation, and act as a natural buffer to protect waterways from being contaminated by harmful chemicals (Jolly et al. 2015; Yuan et al. 2015). Annually, almost 300 000 people are affected and millions of hectares of land are burned globally due to forest fires (Lee et al. 2017). Between 2017 and 2019, wildfires in California forced the evacuation of approximately 1.1 million people, burning 1.48 million acres of land and causing the death of almost 100 people (Wong et al. 2020). The recent Australian wildfires, that cumulatively burned for months, also have been extremely disastrous, burning more than 1500 homes and causing the death of more than 23 people (Boylan and Lawrence 2020). A more recent forest fire started in an area called Chute-des-Passes in the Lac-St-Jean region of Quebec, Canada, has burnt over about 62 000 ha—or an area the size of the island of Montreal, plus about half of Laval (CBC News 2020). With rapid climate change and increased human activities, it is expected that the frequency of forest fires would double over the next century in North America, especially in western Canadian provinces (McKendry et al. 2019). Furthermore, a recent study conducted by Wang et al. (2017) has concluded that over the next century, the spread days of large-scale forest fires in Canada could increase by two–three times. According to Vardoulakis et al. (2020), climate change is the underlying cause behind the recent catastrophic wildfires of Australia, California, Southern Europe, and Southeast Asia, which may get more frequent and intense over time. By June 2020, 502 forest fires have occurred in Quebec, Canada, burning 63 400 ha of lands; this is more than two times the average burnt land and number of forest fires over the last 10 years (The Canadian Press 2020). These reports indicate that in the future, not only would the frequency of wildfires increase but also their severity. The time difference between detection and alarming the relevant authorities plays a crucial part in successful containment and suppression; therefore, early detection is the key to ensure that wildfires remain manageable (Akhloufi et al. 2018).

Conventionally, forest fires had been detected and monitored by human patrols and human observers from strategically located watchtowers; these are monotonous tasks associated with error and fatigue (Çetin et al. 2013; Allison et al. 2016; Akhloufi et al. 2018). As technology progressed, smoke and heat sensors were placed in different parts of forests to detect forest fire. However, these sensors require proximity to the fire for detection. Furthermore, they failed to provide additional information such as the exact location and

the size of the fire (Yuan et al. 2015; Allison et al. 2016). Forest fire detection using satellite images obtained from Moderate Resolution Imaging Spectroradiometer, Advanced Very High-Resolution Radiometer, Visible and Infrared Scanner, and Geostationary Operational Environmental Satellite, provided poor spatial and temporal resolution, which was not good enough for early detection and continuous monitoring (Giglio 2007). As computers and sensors became more powerful and readily available, computer vision algorithms using visual and infrared sensors started being used to remotely detect forest fires (Arrue et al. 2000; Jaber et al. 2001; Vicente and Guillemant 2002; San-Miguel-Ayanz and Ravail 2005; Stula et al. 2012). Detailed and up-to-date reviews of forest fire detection schemes proposed and implemented over the years can be found in Alkhatib (2014), Yuan et al. (2015), Allison et al. (2016), and Akhloufi et al. (2018). These algorithms were mostly implemented or designed for strategically located watchtowers. However, watch towers lack flexibility, have a limited field of view, and can be expensive to construct, especially in remote locations (Alkhatib 2014). The use of aerial platforms such as manned aircraft and unmanned aerial vehicles (UAVs) could address the problems of every traditional platform (Yuan et al. 2015; Allison et al. 2016; Akhloufi et al. 2018). Aircraft can provide a large field of view, flexibility, maneuverability, and rotary-wing aircraft can specifically hover over the fire; additionally, the use of UAVs significantly decreases the risks associated with the safety of the pilot and allows long-term monitoring. Small-sized rotary-wing UAVs are easy to deploy, can hover, and can take images from complex angles to better monitor fire, making it the most ideal platform for fire monitoring tasks (Sherstjuk et al. 2018).

While there are significant advantages of using UAVs for forest fire detection, it brings its own set of challenges. UAV operation comes with local regulatory restrictions concerning privacy, public safety, avoiding collisions, security, etc. (Mozaffari et al. 2019). For instance, in Canada, UAV regulations include obtaining a pilot license, always maintaining a line of sight, avoiding forest fires, etc. (Transport Canada 2018). However, these regulations are mostly for recreational flyers and forest fire monitoring operations are usually operated by local authorities and government agencies where some of these rules are relaxed: for example, the Ikhana missions conducted by National Aeronautics and Space Administration (NASA) and United States Forest Services during 2006–2010 (Ambrosia et al. 2011), the 2009 Crazy Mountain Complex fire monitoring by University of Alaska Fairbanks, and the Dutch FireFly project (Hossain et al. 2019). The success of these missions demonstrates the advantages of using UAVs in forest fire monitoring situations. It also shows that with stronger and lower-cost UAV platforms, available safe UAV control schemes, and fast and accurate detection algorithms, the government agencies could adopt UAV-based forest fire monitoring application more willingly. Safe and fault-tolerant UAV control has already experienced significant advancements (Z.X. Liu et al. 2019; Yu et al. 2019), and a robust detection scheme will take UAV-based forest fire monitoring one step closer to being the preferred method.

Additionally, UAVs are nonstationary platforms that introduce complexities when attempting to extract the motion feature of flame and smoke (Yuan et al. 2017). Furthermore, the flame pixels in the images obtained from aerial vehicles could vary widely in size depending on the range of the observed scene from the aircraft. In fact, a large area of flame could be represented by only a few pixels in the captured image. Also, for a robust, continuous, reliable, and practical fire detection system, it is important that visual signatures of fire, flame, and smoke are detected. The approach proposed in this paper attempts to solve this problem by proposing a single artificial neural network (ANN) that can detect both flame and smoke, under a wide variety of colors, ranges, shapes, and illumination.

The main contributions of this paper can be outlined as follows:

- Proposing a novel method of fire detection using both the texture and color features of flame and smoke from aerial static images and with an ANN classifier, that can be incorporated into a small-sized UAV.
- Displaying the advantage of using multi-space local binary pattern (LBP) features from YCbCr color space's chrominance blue (*Cb*), chrominance red (*Cr*), and hue, saturation, and value (HSV) color space's saturation color channels to extract flame- and smoke-specific texture features.
- Implementing maximum pixel values as a feature along with commonly used mean pixel values of a region to detect small flame pixels, which improved detection performance and allowed robust distinguishing capability between flame and smoke.

This paper has been organized as follows: Section 2 highlights the relevant work done in this field; Section 3 elaborates the overall approach by describing the color spaces, features, and classifiers used; and Section 4 discusses the results through images, tables, and comparative analysis.

2. Relevant work

Fire detection using computer vision has been an active area of research for almost three decades. The principal idea behind computer vision based fire detection is to detect the two visual signatures of fire, flame and smoke, by extracting their useful features and then setting different rules to segment the flame and smoke regions or by sending the feature vector through a machine learning classifier such as Bayes classifier, support vector machine (SVM), ANN, etc. However, the majority of these approaches have focused on detecting only a single signature from images and videos. There are some limitations to using only a single signature. For instance, a forest fire in its early stages may not even have a flame or it could be hidden beneath the forest canopy. On the other hand, smoke appears earlier than flame and can be detected from far away. However, during nighttime operations, the smoke becomes invisible. Therefore, for a robust and continuous forest fire detection system, it is important to use both signatures. In general, there are three approaches taken to visually detect flame and smoke as illustrated in Fig. 1.

Although several articles have been published on computer vision based fire detection techniques, the detection of both flame and smoke is still scarce (Çetin et al. 2013). Both flame and smoke detection using aerial platforms, especially UAVs, is even rarer (Yuan et al. 2015; Hossain et al. 2019). Table 1 highlights the most prominent research conducted thus far to solve the problem of detecting smoke and flame together. Most of the work utilizes two sets of rules or classifiers to separately detect smoke and flame, making it two times more computationally expensive. If UAV-based platforms are considered, it is important to reduce the computational complexity for onboard computation.

Table 2 summarizes the existing work pertinent to the scope of the paper: UAV-based forest fire detection using computer vision techniques. In August 2006, Esposito et al. (2007) experimented with a fixed-wing mini UAV called "1st", supported by the Department of Space Science and Engineering of Italy. The UAV was equipped with a visual camera, a thermal camera, and two hyperspectral sensors operating in the visible-near infrared and near-infrared regions. The aim of the experiment was forest fire prevention, management, damage mitigation, and postfire burnt area mapping. Although Esposito et al. (2007) mentioned the capability of detecting fire from hotspots and smoke plumes, any fire detection scheme or results were not reported. Between 2006 and 2010, NASA and the United States Forest Service conducted 14 missions and flew over 57 fires using the Ikhana

Fig. 1. Common approaches to flame and smoke detection using computer vision.

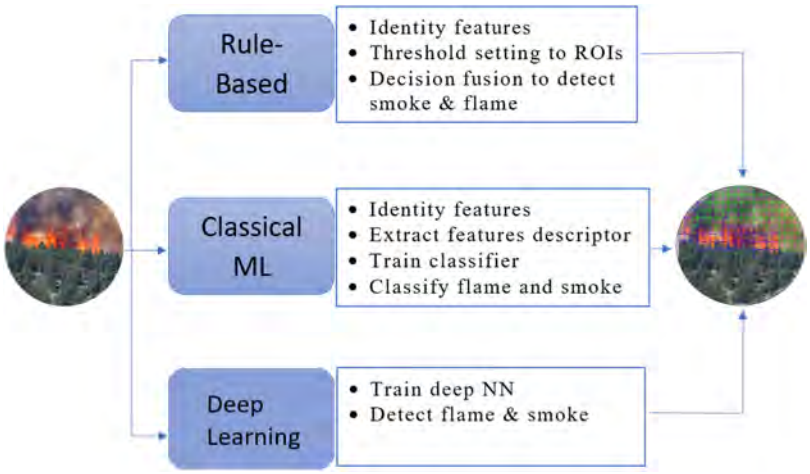


Table 1. Flame and smoke detection methods using computer vision.

Authors	Approach	Features
Wittkopp et al. 2001	Rules-based	Mean intensity and standard deviation of grayscale images.
Chen et al. 2004	Rules-based	RGB and HSV color values and frame differencing.
Han and Lee 2006	Rules-based	RGB color rules for flame detection.
Lai et al. 2007	Rules-based	Color, motion and edge rules for smoke detection. RGB and YCbCr color rules along with motion information for flame detection. Change of intensity histogram and power change for smoke detection.
Ho 2009	Classical machine learning (fuzzy logic)	Two separate fuzzy logic classifiers for flame and smoke. Motion history image, HIS color space and flicker analysis.
Çelik et al. 2007	Classical machine learning (fuzzy logic) for flame and rules-based for smoke	YCbCr color values for flame and RGB and HSV color values for smoke detection.
Yuan 2010	Classical machine learning (Bayes classifier)	Gaussian mixture model for color feature extraction and dynamic change of shape to detect flame. Sum of absolute difference and histogram of motion orientation for smoke.
Kolesov et al. 2010	Classical machine learning (ANN)	Optimal mass transport.
Yu et al. 2013	Rules-based for flame and classical machine learning (ANN) for smoke.	Thresholding sum of pixel values over time for flame detection and optical flow for smoke.
Frizzi et al. 2016; Kim and Lee 2019	Deep learning	N/A

Note: RGB, red–green–blue; HSV, hue, saturation, value; Y, luminance; Cb, chrominance blue; Cr, chrominance red; HIS, hue, intensity, saturation; ANN, artificual neural networks; N/A, not available.

Table 2. Existing literature on unmanned aerial vehicle based forest fire detection.

Approach	Reference	Sensors	Flame	Smoke	Hotspot
Rules-based	Esposito et al. 2007	Visual, visual near infrared, near infrared, thermal infrared	—	√*	√*
	Ambrosia et al. 2011	16-channel hyperspectral	—	—	√
	Yuan et al. 2017, 2019	Visual and infrared	√	√	—
	Sun et al. 2017	Visual	√	—	—
	Sherstjuk et al. 2018	Visual and infrared	√	√	—
	Wardihani et al. 2018	Infrared	√	—	—
	Merino et al. 2012	Visual and infrared	√	—	—
Classical machine learning	Not available				
Deep learning	Zhao et al. 2018; Jiao et al. 2019	Visual	√	√	—
	Chen et al. 2018	Visual	√	—	—
	Alexandrov et al. 2019	Visual	—	√	—

*No specific detection algorithm was reported.

UAV and the Autonomous Modular Scanner-wildfire, a hyperspectral 16-band payload system (Ambrosia et al. 2011). Potential hotspots were detected by setting rules on band 7, 11, and 12.

The COMETS project, funded by the European Commission, was aimed at creating a complete fire monitoring system that provides fire detection, diagnosis, and prognosis (Merino et al. 2012). They established a co-operative UAV system with two unmanned helicopters and an unmanned blimp. Fire detection was a combined effort of a fire sensor, red–green–blue (RGB) rules in visual images and infrared image fusion. A similar approach was proposed by Sherstjuk et al. (2018), using three fixed-wing micro UAVs for patrol missions and a rotary-wing micro UAV for the confirmation mission. An infrared camera was used to detect potential fire regions. Following the suspicion, the onboard optical camera used RGB rules and texture classification to detect smoke. Sherstjuk et al. (2018) noted that because the UAV control system took over a significant portion of the computation load, much of the image processing task had to be done using the ground command station.

Both flame and smoke were also detected by Yuan et al. (2017, 2019), where they used a QBall X4 quadrotor for laboratory experiments. Their fire detection scheme used "Lab" color space rules, optical flow rules, and infrared channel fusion to detect the flame. Smoke detection was done using an extended Kalman filter-based rule. Similarly, Sun et al. (2017) used YCbCr color rules and Lucas–Kanade optical flow to detect flame using a UAV. They used a Parrot AR-Drone for laboratory experiments. Both these approaches exposed one of the main weaknesses of combining multiple rules for flame and smoke detection. Rules-based methods generally combine multiple "AND" operations (i.e., if FeatureA > Threshold1 AND FeatureB < Threshold2, overlapping region = FIRE); therefore, information loss may occur during this process. The final resulting image could have fewer flame or smoke pixels than reality and it would cause problems in the case of a minuscule flame or smoke region and with further processing such as estimating the area of the fire, geolocation, and creating a fire front model.

While the rules-based UAV-based forest fire detection approaches have displayed satisfactory performance in indoor and controlled prescribed fire environments, applying them in real forest fire monitoring tasks could significantly affect their performance due to varying illuminance and other environmental factors (Akhloufi et al. 2018). Machine learning based methods could make the flame and smoke detection algorithms more robust to the changes in the environment. The last decade has observed the surge of deep learning methods

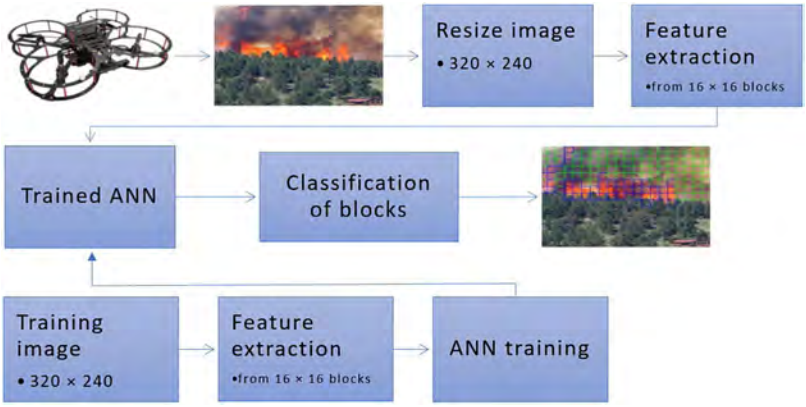
in computer vision tasks and over the last five years, many fire detection algorithms with convolutional neural networks (CNNs) have been proposed. UAV PHECDA II, a fixed-wing UAV was developed by Zhao et al. (2018) for wildfire detection; their approach can be divided into three main parts. First, they have proposed a region of interest (RoI) detection using Bayes-based saliency detection. After extracting a RoI, they extracted color moment, image energy, and entropy of each RoI to form a feature vector. Second, they used two separately trained logistic regression classifiers to classify the potential RoIs using the feature vector. If flame and smoke are confirmed by the classifiers, a minimum bounding rectangle method is used to crop the region. Third, the resulting data are processed through their CNN model called Fire_Net for flame or smoke confirmation. They have reported several limitations during their experiments such as lack of real-time video transmission due to blind zones of global packet radio service/global system for mobile communication (GPRS/GSM) network and analyzing videos after the UAV finished its survey mission. Additionally, the authors reported a processing speed of 41.5 ms for the Fire_Net module, in an offline mobile graphics processing unit (GPU). This is not counting the saliency-based region extraction, feature vector extraction, and classification using logistic regression.

Alexandrov et al. (2019) experimented with different well-established CNN models such as R-CNN (Ren et al. 2015), YOLO (You Only Look Once) (Redmon et al. 2016), and SSD (Single Shot Multibox Detector) (Liu et al. 2016) along with Haar and LBP texture descriptors to detect smoke from UAV captured images. Their study concluded that the SSD algorithm performed the worst both in terms of accuracy and computation. A similar approach was taken in Lee et al. (2017), where AlexNet (Krizhevsky et al. 2012), GoogLeNet (Szegedy et al. 2016), VGG-13 (Simonyan and Zisserman 2014), and their variations were used to detect fire from aerial platforms. Recently, Jiao et al. (2019) used YOLOv3, a state-of-the-art object detection algorithm to detect wildfire flame and smoke. Their method used a small, but wide, dataset that contained flame and smoke images of different sizes and colors taken from different ranges, luminances, etc. They reported a detection rate of approximately 83% with a frame rate of 6.5 frames per second on a DJI Manifold onboard processor.

Although deep learning based approaches have demonstrated promising results, the implementation of such computationally expensive algorithms for small size, low-cost UAVs is still difficult. UAVs need to process much information from their onboard sensors while maintaining proper flight control and navigation, and if computationally expensive image processing tasks are added, it becomes a challenge to maintain proper, safe, and steady operation within its available low power and computational resources (Lu et al. 2018). Although modern mobile GPUs are well known for being able to perform computationally exhausting tasks, integration of such resources is still challenging as these systems demand high power (an onboard system with such capability is counterproductive to UAVs desired lightweight functionality). Communicating the raw data to a server may act as a solution to this problem; however, high-quality wireless video transmission—even when possible—adds significant latency to the system thus rendering the whole process more computationally expensive than desired in addition to risking the security of the data (Carrio et al. 2017; Kyrkou et al. 2018). This was further demonstrated by Hulens et al. (2016) through experimentation that higher computational complexity causes more power consumption and less flight time for UAVs.

From the existing approaches and challenges discussed in this section, the problem statement can be summarized as such: for continuous, reliable, and early fire detection, it is necessary to formulate an approach that is lightweight, deployable in a fixed-wing or rotary-wing UAV, and can detect both smoke and flame under different environmental conditions. This paper attempts to address these challenges by bridging the gap between often unreliable rule-based methods and computationally expensive deep-learning

Fig. 2. Proposed methodology.



methods through the proposed framework of flame and smoke detection using a color-texture descriptor and classical machine learning.

3. Methodology

The overall principle of the proposed method is illustrated in Fig. 2. The idea is that one or more UAVs would be patrolling over forests, capturing images as they fly. The ANN used in the proposed method is trained using 320×240 images. Each image was divided into 300 16×16 image-sized blocks and features were extracted from each block. The captured images from the UAVs would also be resized into 320×240 images and the same features would be extracted from the 16×16 blocks. The feature vector from each block will go through the trained ANN to perform the classification of flame, smoke, or background. The final output will display detected flame and smoke blocks with a rectangular overlay.

This section will highlight the features used to train and test the network, along with the working principle of an ANN and its training procedure.

3.1. Color spaces

Both flame and smoke display strong, albeit not unique, color features. Flame color could be red, orange, or yellow, and smoke color could be anything between whiteish gray to blackish gray. Despite this wide inter-class and intra-class color variations, color is still one of the most prominent features of flame and smoke. Generally, images are captured in RGB format. Figures 3 and 4 display an example of an RGB aerial image used to train the classifier followed by its representation in each of the three channels.

Figure 4 displays one disadvantage of using this color space for flame and smoke detection. It can be observed that the sky in the background and flame pixels are saturated in both R and G channels. This is because in RGB color space, luminance and chrominance are not decoupled. To remove this limitation, the better approach is to convert the images into YCbCr, HSV, or CIE Lab color spaces. Figure 5 shows the *Y*, *Cb*, *Cr*, *H*, *S*, *V*, *L*, *a*, and *b* color channel representation of Fig. 3. It can be observed that with *Cb*, *Cr*, *H*, *S*, *a*, and *b*, the color information of smoke and flame are prominent while *Y*, *S*, and *L* color spaces display the luminance of flame prominently.

HSV color space mimics how human eyes perceive color; hue is the pure color of the pixel, saturation denotes how much white is mixed with the pure color, and value represents the intensity of the pixel. In YCbCr, *Y* is the luminance component, *Cb* is blue subtracted from the luminance and *Cr* is red subtracted from luminance. Similarly, in CIE Lab

Fig. 3. A red–green–blue training image.

Fig. 4. Red, green, and blue color split of Fig. 3.

color space, L represents the luminance, "a" represents the colors within green to magenta, and "b" represents the colors within blue to yellow. The mathematical equations to perform RGB to HSV, YCbCr, and CIE Lab conversion are as follows:

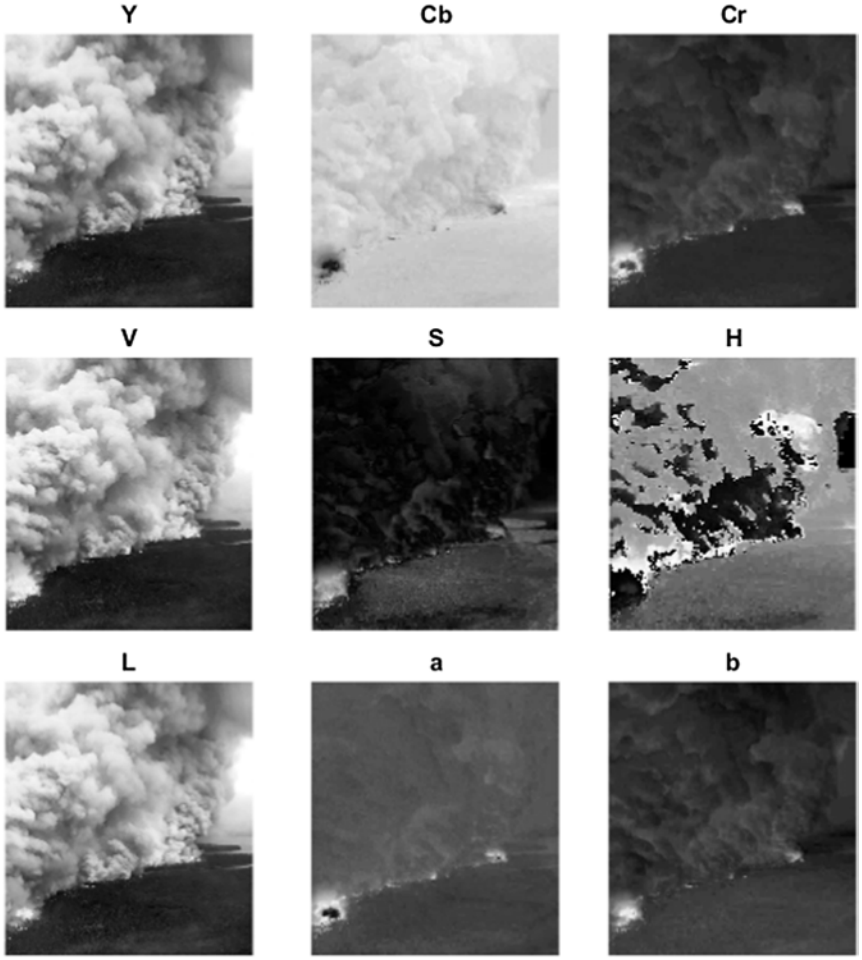
$$(1)\quad h = \begin{cases} 0, & \text{if } \max(R,G,B) = \min(R,G,B) \\ \left(60^\circ \times \frac{g-b}{\max(R,G,B)-\min(R,G,B)} + 0^\circ\right) \bmod 360^\circ, & \text{if } \max(R,G,B) = r \\ \left(60^\circ \times \frac{b-r}{\max(R,G,B)-\min(R,G,B)} + 120^\circ\right) \bmod 360^\circ, & \text{if } \max(R,G,B) = g \\ \left(60^\circ \times \frac{r-g}{\max(R,G,B)-\min(R,G,B)} + 240^\circ\right) \bmod 360^\circ, & \text{if } \max(R,G,B) = b \end{cases}$$

$$(2)\quad s = \begin{cases} 0, & \text{if } \max(R,G,B) = 0 \\ \frac{\max(R,G,B)-\min(R,G,B)}{\max(R,G,B)} = 1 - \frac{\min(R,G,B)}{\max(R,G,B)}, & \text{otherwise} \end{cases}$$

$$(3)\quad v = \max(R,G,B)$$

$$(4)\quad Y = 0.299R + 0.587G + 0.114B$$

Fig. 5. *Y*, *Cb*, *Cr*, *V*, *S*, *H*, *L*, *a* and *b* color channel representation of Fig. 3. Y, luminance; Cb, chrominance blue; Cr, chrominance red; V, value; S, satruation; H, hue; L, lightness.



(5) $Cb = B - Y$

(6) $Cr = R - Y$

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

(8) $$L = \begin{cases} 116 \times \left(\frac{Y}{Y_n}\right)^{\frac{1}{3}} - 16, \text{if } \left(\frac{Y}{Y_n}\right) > 0.008856 \\ 116 \times \left(\frac{Y}{Y_n}\right), \text{if } \left(\frac{Y}{Y_n}\right) \leq 0.008856 \end{cases}$$

(9) $$a = 500 \times \left\{ \left(\frac{X}{X_n}\right)^{1/3} - \left(\frac{Y}{Y_n}\right)^{1/3} \right\}$$

(10) $$b = 200 \times \left\{ \left(\frac{Y}{Y_n}\right)^{1/3} - \left(\frac{Z}{Z_n}\right)^{1/3} \right\}$$

where X_n, Y_n and Z_n are known as the tristimulus values of reference white points.

For the proposed algorithm, to identify flame and smoke regions and to separate them from other bright objects in nature, mean values of *Cb*, *Cr*, *H*, *S*, *V*, *a*, and *b* color spaces are obtained from each block. Along with these features, to identify small flame pixels present in a block, max values of *Y*, *Cr*, *H*, *S*, *a*, and *b* values from each block are extracted as well.

3.2. Texture features

While color features can be a powerful tool to identify potential flame and smoke blocks, they also introduce false alarms by detecting similar colored objects in nature; to separate these objects from flame and smoke, more discriminating features need to be extracted. Generally, smoke and flame have fluctuating pixel values within a given area, giving it a rough texture. On the other hand, similar colored objects such as red, yellow, or orange flowers; red cars; fall leaves; road surfaces, etc. have a smooth surface and the pixel values do not fluctuate as much. Thus, texture features can represent whether the region under question has a smooth or rough surface. In the proposed method, gray level co-occurrence matrix (GLCM) and local binary pattern (LBP) have been experimented with as texture feature extractors along with the variance in *Cb* color space.

3.2.1. Gray level co-occurrence matrix (GLCM)

GLCM is one of the most popular texture feature extractors. A GLCM forms a matrix where each element is a representation of how many times a pair of gray values (*i*, *j*) have occurred in a neighboring manner. For instance, a GLCM with 256 gray levels will form a 256×256 matrix where the element in the first column of the first row will represent how many times the gray values (0, 0) have occurred as a neighbor. Subsequently, the second column of the first row will represent how many times the pixel values (0, 1) have occurred as neighboring pairs.

In the proposed method, four gray level co-occurrence matrices were formed from each block, observing the *right*, *left*, *up*, and *down* neighbors. From each GLCM, three statistical features called *contrast*, *homogeneity*, and *energy* were extracted. Contrast represents the local variation within a block, energy represents uniformity and homogeneity represents the closeness of the distribution of the GLCM (Partio et al. 2002). These can be mathematically represented by (Haralick et al. 1973):

$$\text{Contrast} = \sum_{i,j} |i-j|^2 p(i,j) \tag{11}$$

$$\text{Energy} = \sum_{i,j} p(i,j)^2 \tag{12}$$

$$\text{Homogeneity} = \sum_{i,j} \frac{p(i,j)}{1+|i-j|} \tag{13}$$

here, *i* and *j* are rows and columns of the GLCM. In total, 12 GLCM features are extracted from each block for training and testing purposes.

3.2.2. Local binary pattern (LBP)

LBP is another powerful texture feature extractor introduced by Ojala and Pietikäinen (1999). In this method, an image is divided into 3×3 neighborhoods and a threshold is applied to the neighborhood depending on the gray value of the center pixel. Higher or equal values from the center pixel are set to 1 and lower values are set to zero. This creates a binary 3×3 neighborhood. Each binary digit in a clockwise or anticlockwise sequence is multiplied with a weight and then summed. If 8 surrounding neighbors from the

center pixel are considered, then the LBP is an 8-bit number implying that there could be 256 possible patterns. Finally, a histogram is created of the patterns found throughout the image. Mathematically, the calculation is done as:

$$\text{LBP} = \sum_{n=0}^{N-1} i(n)2^n \quad (14)$$

where N is the number of neighbors considered. To efficiently compute LBP and to reduce the number of features, a modification of LBP, uniform LBP has been implemented that reduces the number of features to 59 from 256 (Topi et al. 2000).

Generally, LBP is implemented on grayscale images. An RGB image is converted into grayscale and then LBP is applied to that grayscale image. For face recognition, eye detection, etc. this method works without problems. Faces and eyes have fixed size and shape, unlike flame and smoke. Grayscale images cannot uniquely distinguish flame and smoke from the background and other objects in nature. Therefore, in the proposed method, a novel idea has been implemented to develop a method optimized for flame and smoke detection: along with the use of grayscale images, the proposed method extracts LBP of *S* color channel. The difference of pixel value between *Cb* and *Cr* color channel can be a powerful tool to identify flame areas (Çelik and Demirel 2009). Therefore, in addition to grayscale and *S* channel LBPs, *Cr-Cb* subtraction is performed for each block and the LBP of the result has been extracted. Overall, 177 LBP features have been extracted from each block.

3.3. Artificial neural network (ANN)

Figure 6 illustrates a type of feedforward ANN called multi-layer perceptron (MLP). In simple terms, an MLP takes in a feature vector and generates a prediction about the output. There are three types of layers available in such ANNs: the input layer, hidden layers, and the output layer. Each layer contains a number of "neurons". The number of neurons in the input and output layers is the same as the size of the input feature vector and output classes. The number of hidden layers and the neurons in each of the hidden layers is chosen by the users. As it can be observed from Fig. 6, the neurons in MLPs have a "dense" connection, which means the neurons are interconnected with other neurons of the previous and following layers.

It is called an Artificial Neural Network because it mimics the mathematical model of biological neurons proposed by McCulloch and Pitts (1943). The mathematical model is as follows:

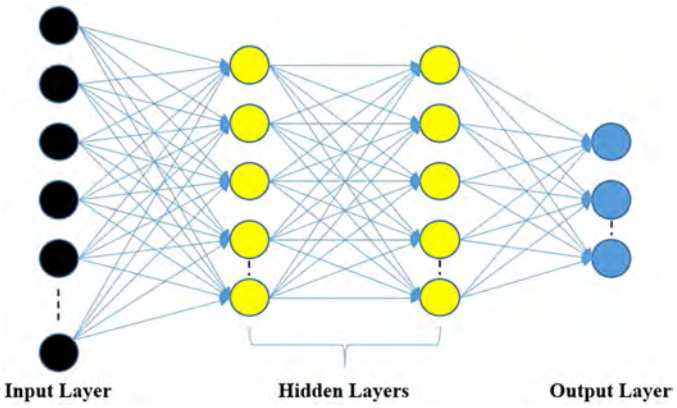
$$Z^L = W^L A^{L-1} + b^L \quad (15)$$

here, A^{L-1} denotes the feature vector from the previous layer (input features when L is the first hidden layer), W^L and b^L are weight matrix and bias vector of the layer L and they are the learnable parameters of the ANN. Before the output Z^L gets passed on to the next layer, an operation called "activation" is performed. The purpose of activation is to inject nonlinearity into the same and to ensure that for certain inputs, only certain neurons are fired up. The mathematical model of the activation operations conducted within a layer then becomes:

$$A^L = f(Z^L) \quad (16)$$

where A^L is the final output of the layer and then it gets passed on to the next layer. Some of the most popular activation functions are sigmoid, hyperbolic tangent, and Rectified Linear

Fig. 6. Simple multilayer perceptron network.



Unit (ReLU). For this paper, ReLU activation was used. The mathematical operation of ReLU is as follows:

$$\text{ReLU}(x) = \max(x, 0) \tag{17}$$

Finally, at the output layer, an activation function called "softmax" has been implemented. Softmax takes in a vector of numbers and creates a probability distribution of the output classes. The mathematical formula of softmax is as follows:

$$A_i^L = \frac{e^{Z_i^L}}{\sum_{j=1}^{n} e^{Z_j^L}} \tag{18}$$

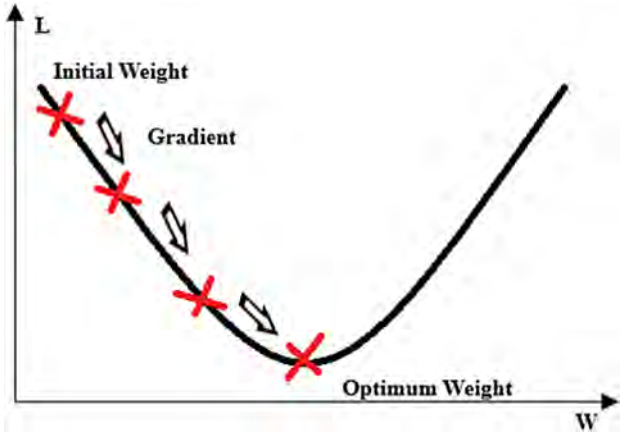
where A_i^L is the output of the ith neuron in layer L, Z_i^L is the ith value of the layer L and n is the total number of neurons in the layer L. The softmax activation function in the output layer generates the probability of which class the block belongs to.

3.3.1. Network training

Before a neural network can be implemented for testing, it needs to be trained. ANN is a "supervised" machine learning algorithm where the ground-truth label of all training data needs to be known. As explained previously, a neural network makes a prediction using a "forward pass" where processed data from one layer is passed on to the next layer. The training process of a neural network flows in the opposite direction, starting from the output layer and propagating back to the first hidden layer. This process is called "backpropagation".

During backpropagation, the predicted output is evaluated against the ground truth using a "loss function". The most common cost functions are mean squared error, cross-entropy loss, etc. (Khan et al. 2018). The value of the loss function indicates how accurately the network can make predictions on the training set. Based on the loss function value, the network then adjusts the value of the weights and biases. The principle behind this adjustment is illustrated in a simple manner using Fig. 7. The idea is that, when the weights and biases are initialized at the beginning of network training, the loss is high and for some certain value of the weight, the loss function would be minimum. The objective of the training is to find those optimum weights and biases. As the system approaches optimum weights, the loss gets smaller and the opposite happens when going further away from

Fig. 7. The basic principle of gradient descent.



the optimum weights. With that in mind, the weight vs loss graph can be imagined as a parabola, as illustrated in Fig. 7.

If the gradient of this parabola at any given weight is iterated, it would be possible to eventually reach the minimum. This process is known as "gradient descent" and can be mathematically formulated as:

$$W_{\text{new}} = W_{\text{old}} - \mu \frac{dL}{dW} \tag{19}$$

here, L is the loss function, W_{old} is the previous weight and μ is the learning rate of descent. The learning rate is an important parameter because if it is too small, it will take a long time to converge and if it is too big, the updated weight may not converge towards the minimum. The same formula is applicable for updating the bias parameters.

The straightforward gradient descent as the optimizer has some limitations. For example, in real cases, the weight versus loss graph is not such a nice shape as the parabola illustrated in Fig. 7. In fact, for any practical neural network, the shape of the weight versus loss curve is nonconvex where several local minima could exist, and the training process could get stuck in any of those local minima before reaching the global minimum. As addressed before, the learning rate also plays a major role in network training (Zhang and Li 1999). Furthermore, the network could also suffer from exploding or vanishing gradient problems (Khan et al. 2018). Another problem with neural networks, both deep and shallow, is that they could suffer from overfitting. It happens when the network "learns too much" and fails to generalize. When overfitting occurs, the network's prediction accuracy in the training set is close to 100% but for validation and evaluation data, it predicts with poor accuracy. Avoidance of local minima and overfitting are active research topics (Li and Malik 2017; L. Liu et al. 2019; Sonoda et al. 2019), and research thus far suggests that there are a few strategies such as random weight initialization, momentum, regularization techniques, etc. that could be adopted to achieve avoidance of good enough convergence without overfitting the network (Khan et al. 2018). The following subsections will briefly highlight the strategies adopted in this paper to achieve these goals.

3.3.1.1. Momentum and adaptive moment estimation (Adam)

As it has been mentioned in the previous section, there could be several local minima in the weight versus loss function surface and any gradient-based learning method will most likely converge towards a local minimum. Among all the local minima that could exist within the error surface, not all of them are equal. There would be local minima with high loss, which are considered "bad", and local minima that are fairly flat and have a low loss, which could be considered as "good". In fact, it is often speculated that a flat local minimum with significantly low loss value could generalize better than the global minimum (Kleinberg et al. 2018). Additionally, it has been observed that the learning rate has an important role in escaping the bad local minima (Loshchilov and Hutter 2016). It has been proven that the noise present in Stochastic Gradient Descent can settle to a local minimum with low loss value (Zhang et al. 2017). The addition of momentum and adaptive learning rate to the gradient descent method can achieve faster convergence while being able to avoid bad local minima.

In gradient descent with momentum, eq. (19) is modified to be:

$$W_{\text{new}} = W_{\text{old}} - a_t \tag{20}$$

$$a_t = \mu \frac{dL}{dW} + \gamma a_{t-1} \tag{21}$$

here, the term γ is called the momentum and it uses the current gradient as well as the previous gradient to suppress or increase its acceleration. It can be imagined as a ball rolling down a hill and more momentum is added to the ball when rolling down the direction of the maximum slope. By this physical analogy, it could be imagined that adding momentum allows gradient descent-based learning to "jump over" small peaks and valleys (Khan et al. 2018).

Over the years, many variants of gradient descent with momentum has been proposed such as Nesterov momentum (Nesterov 1983), Adaptive Gradient (AdaGrad) (Duchi et al. 2011), Adaptive Delta (Zeiler and Fergus 2014), Adam (Kingma and Ba 2017), etc. Adam is currently one of the most popular optimization algorithms to train neural networks and it addresses the problems of its predecessors such as vanishing learning rate problems and providing optimal solutions for sparse gradients. Adam modifies eq. (19) as such (Khan et al. 2018):

$$W_{\text{new}} = W_{\text{old}} - \frac{\mu}{\sqrt{\hat{E}[\delta^2]_t} + \epsilon} \hat{E}[\delta]_t \tag{22}$$

$$\hat{E}[\delta]_t = \frac{E[\delta]_t}{1 - \gamma_1^t} \tag{23}$$

$$\hat{E}[\delta^2]_t = \frac{E[\delta^2]_t}{1 - \gamma_2^t} \tag{24}$$

$$E[\delta]_t = \gamma_1 E[\delta]_{t-1} + (1 - \gamma_1)\delta_t \tag{25}$$

$$E[\delta^2]_t = \gamma_1 E[\delta^2]_{t-1} + (1 - \gamma_2)\delta_t^2 \tag{26}$$

here, $E[\delta]_t$ and $E[\delta^2]_t$ represent the running mean and variance of the gradients and $\hat{E}[\delta]_t$ and $\hat{E}[\delta^2]_t$ are their initialization corrected estimates. γ_1 and γ_2 are the parameters for the running mean and variance of the gradients. Adam has been proven to be one of the best performing optimizers, achieving optimum performance using momentum and adaptive learning rate.

3.3.1.2. Dropout

To avoid the problem with network overfitting, several regularization techniques such as data augmentation, batch normalization, l^1 and l^2 norm, early stopping, etc. have been proposed. The method applied in this paper is called "dropout". It is one of the most popular methods of regularization and the idea behind it is quite simple: a certain percentage of the neurons "freeze" and stop learning after a certain point during training to prevent overfitting (Khan et al. 2018). The neurons are chosen randomly but the percentage of "frozen" neurons is user-defined.

In summary, the neural network in this paper has been trained using mini-batch gradient descent with batch size = 500 and Adam optimizer with $\gamma_1 = 0.9$, $\gamma_2 = 0.999$. Through trials and errors, dropout = 0.3 (30%) was selected. These measures were taken to achieve a good optimum while avoiding overfitting.

4. Results

A major problem with UAV-based forest fire detection is that there are no publicly available data sets of aerial forest fire images and videos (Zhao et al. 2018; Jiao et al. 2019). Therefore, for this paper, a data set was created by gathering images from the internet, using sites such as Google Images, Shutterstock, Flickr, and news reports of forest fires. This data set consists of mostly aerial forest fire images, along with aerial images of forest backgrounds. The resolution of the downloaded images varies from 3000×3000 pixels to 512×512 pixels. The forests in these images include the recent wildfires of California and Australia along with images from British Columbia, Alberta, North and South Carolina, Colorado, Indonesia, etc. The total data set has 460 images, where 360 images were used for training and 100 images were used for evaluation purposes. To display the strength of the proposed method, several combinations of features were evaluated and the results are summarized in Table 3. Table 4 shows a comparative analysis between the proposed method, the proposed features with an SVM classifier and a state-of-the-art deep learning algorithm, YOLOv3.

The ANNs used in this paper all have a single hidden layer, however, depending on the number of features used, the number of neurons in the hidden layer has been different. The networks were all trained with 108 000 image blocks extracted from the training data set and tested on 30 000 image blocks extracted from the test set. In the training set, the number of flame, smoke, and background blocks was 14 744, 27 697, and 65 559, respectively. In the evaluation set, the number of flame, smoke, and background blocks was 1916, 12 078, and 16 006, respectively. The comparatively fewer flame blocks can be attributed to the fact that one of the main purposes of the proposed method was to be able to detect minuscule flame pixels, resulting in many training and testing images where there were only a handful of flame blocks. In total, 206 features have been extracted from each block: 16 color features (8 average + 8 max values), 177 LBP features (59 features each from gray level, saturation and *Cb*-*Cr* color channels), 12 GLCM features (contrast, homogeneity and energy from 4 directions), and the variance in *Cb* color space as an additional texture discriminator.

The performance metrics used in this paper are precision, recall, and *F*1 score. High precision indicates a low false alarm rate, high recall indicates low missed detection and *F*1 score is the weighted average of precision and recall. Mathematically, they are formulated as:

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}} \quad (27)$$

Table 3. Results obtained using various combinations of features.

	Precision			Recall			*F*1		
Features	Flame	Smoke	Background	Flame	Smoke	Background	Flame	Smoke	Background
All features (color, LBP, GLCM)	0.89	0.90	0.85	0.73	0.81	0.93	0.80	0.86	0.89
Color + all LBPs	**0.89**	**0.93**	**0.89**	**0.80**	**0.88**	**0.95**	**0.84**	**0.90**	**0.92**
Color + LBP_gray	0.85	0.89	0.86	0.75	0.83	0.92	0.80	0.86	0.89
Color features only	0.87	0.80	0.77	0.72	0.67	0.86	0.79	0.73	0.81
Average color only	0.87	0.79	0.67	0.38	0.45	0.89	0.53	0.57	0.76
Max color only	0.85	0.78	0.74	0.75	0.60	0.85	0.79	0.68	0.79
Texture only	0.64	0.88	0.81	0.52	0.77	0.90	0.57	0.82	0.85
LBP_gray only	0.43	0.87	0.82	0.16	0.75	0.88	0.85	0.24	0.81
Three LBPs combined	0.60	0.88	0.80	0.47	0.77	0.90	0.85	0.52	0.82
Average color + three LBPs	0.89	0.91	0.81	0.55	0.77	0.94	0.68	0.84	0.87

Note: Bold values indicate the best results. LBP, local binary pattern; GLCM, gray level co-occurrence matrix.

Table 4. Performance comparison.

	Flame			Smoke			Runtime
Method	Precision	Recall	*F*1	Precision	Recall	*F*1	(FPS)
Proposed method (color + multi-space LBP + ANN)	0.89	**0.80**	**0.84**	0.93	**0.88**	**0.90**	**19**
Color + multi-space LBP + SVM	0.89	0.72	0.80	0.90	0.86	0.88	2
Color + multi-space LBP + random forest classifier	0.92	0.57	0.71	0.92	0.78	0.84	16
Color + multi-space LBP + Bayes classifier	0.37	0.67	0.48	0.68	0.92	0.79	**19**
YOLOv3	**0.93**	0.47	0.62	**0.98**	0.64	0.77	4

Note: FPS, frames per second; LBP, local binary pattern; ANN, artificial neural network; SVM, support vector machine.

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}} \tag{28}$$

$$F1 = \frac{2 \times \text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}} \tag{29}$$

The performance metrics presented in Tables 3 and 4 demonstrate the potential of the proposed method in detecting both flame and smoke from an image along with the performance boost obtained from the novel features implemented in this paper. In Table 3, the best performance was obtained by utilizing all color and LBP features with 384 neurons in the hidden layer. It was also observed that adding the novel multi-color space LBP and max color values in the feature vector bolster the performance of flame and smoke detection compared to using only the more common average color values and LBP_gray features. Using maximum color values from the blocks increases the detection of small flame pixels, texture features can discriminate smoke and flame from similar colored objects, and the features combined create a robust fire recognition system through the detection of flame and smoke. It was also demonstrated that adding GLCM features does not improve system performance.

In Table 4, a summary of the comparison shows that the proposed method performs better than the SVM, random forest and Bayesian classifiers and YOLOv3. Although the precision, recall and *F*1 scores are similar, the runtime of SVM is significantly longer than

the proposed method. Additionally, ANN, SVM, and random forest classifiers displayed a better *F*1 score than YOLOv3. This experiment showed that flame and smoke specific feature extraction provides a better overall result than the deep learning approach. In terms of precision only, YOLOv3 performed better, signifying that it had fewer false positives. However, YOLOv3 struggled to detect transparent smoke and minuscule flame regions, resulting in a lower recall rate than the proposed method and the SVM method. The YOLOv3 network was trained using Google Collaboratory and the rest of the feature extraction, classifier training, and evaluation were implemented on a computer with Intel Core i7-9750H processor, 16 GB DDR4 RAM and 512 GB SSD. To train the YOLOv3 network, data augmentation was implemented using random cropping and resizing.

Sample result images along with their original images are illustrated in Figs. 8–15. Figure 8 contains forest fire images captured from a close distance. Figure 9*a* shows that tan-colored smoke blocks were detected at the top and smoke-colored nonsmoke blocks were rejected at the bottom of the image. Figure 9*b* shows the robustness of the method during nighttime and with a complex background. In Fig. 9*c*, the red structure at the bottom right corner has been correctly rejected as a fire. Images in Fig. 10 contain transparent smoke and the results presented in Fig. 11 show that the transparent smoke blocks have been correctly classified, even when the smoke blocks were almost blended into the background. Figure 12 contains images under challenging conditions such as low illumination, occlusion, and very small flame pixels. The results displayed in Fig. 13 show satisfactory detection by the proposed approach. The results displayed in Fig. 15 further demonstrate the ability of the proposed method to detect flame and smoke pixels under different conditions. In Fig. 15*a*, the system detected very small fire blocks, and in Fig. 15*b* it accurately detected the smoke and rejected the similar-looking cloud blocks. Its ability to distinguish between cloud and smoke can also be observed in Fig. 11*b*. The results obtained by the proposed method in Figs. 15*a* and 11*b* can be directly compared with the results obtained using YOLOv3 in Fig. 16. The YOLOv3 approach failed to detect the minuscule flame pixels of Fig. 14*a* and the transparent smoke blocks of Fig. 10*b*, further demonstrating the superiority of the proposed approach. Throughout the samples, it has been demonstrated that the system can detect flame and smoke or different size, shape, color, and opacity. Through these validation results, it can be concluded that the proposed approach can be a powerful and practical tool in UAV-based forest fire detection.

5. Conclusions and future works

We proposed a novel forest fire detection method using both flame and smoke signatures. The novelty of this paper includes using both signatures for fire detection to ensure continuous and reliable operation along with using multi-space local binary pattern, that extracted the LBP of gray-level, saturation, and *Cb*-*Cr* color channels combined with using maximum color values within image blocks to identify small flame pixels. Furthermore, the paper focuses on aerial images so that the system can be incorporated into a UAV for UAV-based forest fire monitoring.

The results demonstrated that with the proper features, a single neural network can be a practical and real-time solution to detect both flame and smoke blocks from a UAV-captured static image. This idea is bolstered by experiments that showed the proposed method performs better than the SVM classifier and YOLOv3 object detector. Testing results illustrated in Figs. 8–15 show that it can detect smoke and flames with varying sizes and colors and under a wide range of environmental conditions. A comparison with YOLOv3 showed the superiority of the proposed method in detecting minuscule flame pixels and transparent smoke. Overall, the proposed method has shown the potential of being a practical, real-time solution to the UAV-based forest fire detection problem. However, the

Fig. 8. Close range forest fire images.

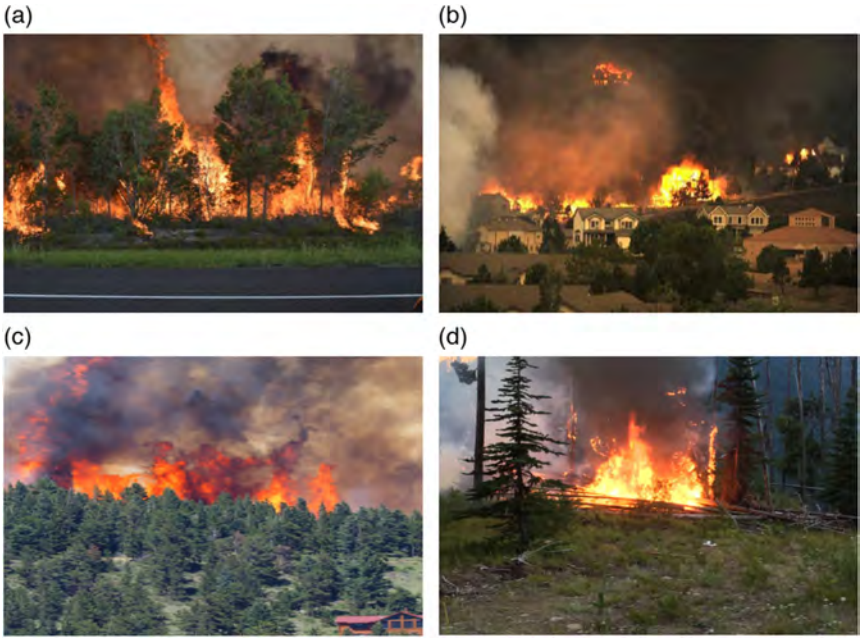


Fig. 9. Results for Fig. 8 images displaying detection of smoke under different illumination and range, detecting different colored smoke blocks in (*a*)–(*d*), flame and smoke detection during nighttime (*b*) (detected flame in blue boxes and detected smoke in green boxes).

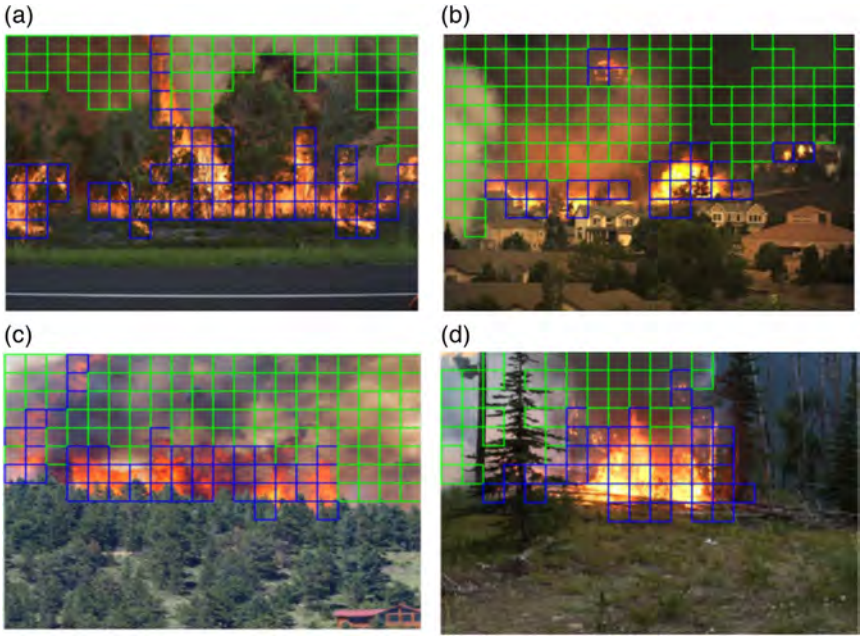


Fig. 10. Forest fire images under bright illumination and transparent smoke.

Fig. 11. Result of Fig. 10 images, displaying successful flame and smoke detection with bright illumination, small flame regions and transparent smoke blocks (detected flame in blue boxes and detected smoke in green boxes).

Fig. 12. Aerial forest fire images with very small fire areas.

precision, recall, and *F*1 scores presented in Table 3 along with some of the presented images indicate that more improvement in the algorithm is required before it can be incorporated into a completely autonomous system. For example, the network fails to detect smoke blocks with a smooth texture and the small flame pixel detection could be further improved. To improve the result, future work will involve increasing the training data set, making sure that there are no errors in labeling ground truth, and modifying the multi-space LBP algorithm to optimize it for fire detection. Future work will also include incorporating the proposed method onboard a UAV for a real-time field test. Furthermore, recent advancements in deep learning techniques are undeniable and further investigation will be conducted into developing an efficient CNN that could be deployed into a UAV for fire detection.

Fig. 13. Results of Fig. 12 images, displaying successful detection of minuscule flame blocks under low illumination and smoke occlusion (detected flame in blue boxes and detected smoke in green boxes).

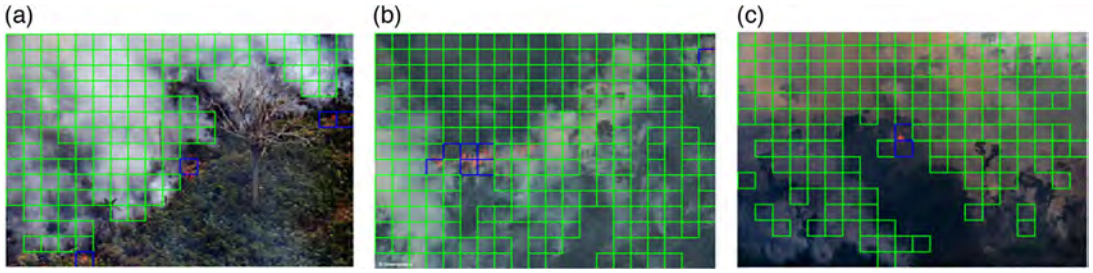
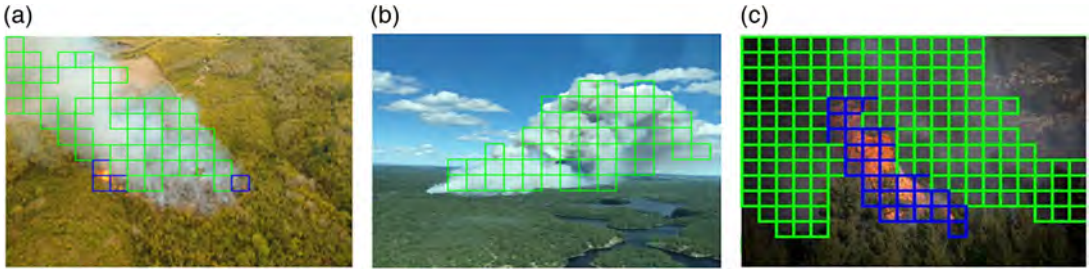


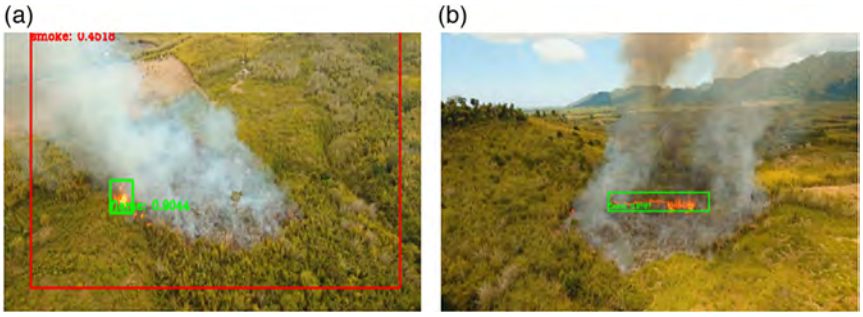
Fig. 14. More sample test images under different conditions.

Fig. 15. Results of Fig. 14 images, displaying successful detection of minuscule flame blocks (*a*), rejecting clouds from being detected (*b*) and demonstration the capability of detecting a wide range of smoke and flame (detected flame in blue boxes and detected smoke in green boxes).



As forest fire occurrences across the world are becoming more frequent and severe, developing a robust, continuous, and reliable forest fire detection system is a current need, and a UAV-based approach can address all the limitations of the current methods. While existing strict UAV regulations remain a constraint in implementing UAV-based forest fire monitoring, previous missions conducted by the government agencies and local authorities encourage the possibility of implementing such methods in real scenarios. The successful implementation of a fire detection algorithm that is robust under environmental conditions, and has high accuracy, combined with the safe UAV flight control strategies can make UAV-based forest fire detection the most preferred choice. The method proposed in this

Fig. 16. Flame and smoke detection using YOLOv3 (red bounding box denotes smoke detection and green denotes flame detection). (*a*) is a comparison with Fig. 15*a*, failing to detect minuscule flame pixels. (*b*) is a comparison with Fig. 11*b*, failing to detect transparent smoke.



paper has the potential for real implementation and the presented results will guide future research towards finding an optimized forest fire detection algorithm.

Acknowledgement

This work is supported by the Natural Sciences and Engineering Research Council of Canada (NSERC).

References

Akhloufi, M.A., Castro, N.A., and Couturier, A. 2018. UAVs for wildland fires. *In* Autonomous Systems: Sensors, Vehicles, Security, and the Internet of Everything, Orlando, Fla., USA, 15–19 April 2018. International Society for Optics and Photonics. doi: 10.1117/12.2304834.

Alexandrov, D., Pertseva, E., Berman, I., Pantiukhin, I., and Kapitonov, A. 2019. Analysis of machine learning methods for wildfire security monitoring with an unmanned aerial vehicles. *In* 24th Conference of Open Innovations Association (FRUCT), Moscow, Russia, 8–12 April 2019. pp. 3–9. doi: 10.23919/FRUCT.2019.8711917.

Alkhatib, A.A.A. 2014. A review on forest fire detection techniques. Int. J. Distrib. Sens. Netw. **10**(3): 597368. doi: 10.1155/2014/597368.

Allison, R.S., Johnston, J.M., Craig, G., and Jennings, S. 2016. Airborne optical and thermal remote sensing for wildfire detection and monitoring. Sensors, **16**(8): 1310. doi: 10.3390/s16081310. PMID: 27548174.

Ambrosia, V.G., Wegener, S., Zajkowski, T., Sullivan, D., Buechel, S., Enomoto, F., et al. 2011. The Ikhana unmanned airborne system (UAS) western states fire imaging missions: From concept to reality (2006–2010). Geocarto Int. **26**(2): 85–101. doi: 10.1080/10106049.2010.539302.

Arrue, B.C., Ollero, A., and Matinez de Dios, J.R. 2000. An intelligent system for false alarm reduction in infrared forest-fire detection. IEEE Intell. Syst. Appl. **15**(3): 64–73. doi: 10.1109/5254.846287.

Boylan, J.L., and Lawrence, C. 2020. The development and validation of the bushfire psychological preparedness scale. Int. J. Disaster Risk Reduct. **47**: 101530. doi: 10.1016/j.ijdrr.2020.101530.

Carrio, A., Sampedro, C., Rodriguez-Ramos, A., and Campoy, P. 2017. A review of deep learning methods and applications for unmanned aerial vehicles. J. Sens. **2017**: 3296874. doi: 10.1155/2017/3296874.

CBC News. 23 June 2020. Crews continue to battle Chute-des-Passes wildfire as dozens of forest fires burn across Quebec. Available from https://www.cbc.ca/news/canada/montreal/chute-des-passes-fire-update-1.5624343 [accessed 26 June 2020].

Çelik, T., and Demirel, H. 2009. Fire detection in video sequences using a generic color model. Fire Saf. J. **44**(2): 147–158. doi: 10.1016/j.firesaf.2008.05.005.

Çelik, T., Özkaramanlı, H., and Demirel, H. 2007. Fire and smoke detection without sensors: Image processing based approach. *In* 15th European Signal Processing Conference, Poznan, Poland, 3–7 September 2007. pp. 1794–1798.

Çetin, A.E., Dimitropoulos, K., Gouverneur, B., Grammalidis, N., Günay, O., Habiboglu, Y.H., et al. 2013. Video fire detection — Review. Digit. Signal Process. **23**(6): 1827–1843. doi: 10.1016/j.dsp.2013.07.003.

Chen, T.-H., Wu, P.-H., and Chiou, Y.-C. 2004. An early fire-detection method based on image processing. *In* International Conference on Image Processing, Singapore, Singapore, 24–27 October 2004. Vol. 3, pp. 1707–1710. doi: 10.1109/ICIP.2004.1421401.

Chen, Y., Zhang, Y.M., Xin, J., Yi, Y., Liu, D., and Liu, H. 2018. A UAV-based forest fire detection algorithm using convolutional neural network. *In* 2018 37th Chinese Control Conference (CCC), Wuhan, China, 25–27 July 2018. IEEE. pp. 10305–10310. doi: 10.23919/ChiCC.2018.8484035.

Duchi, J., Hazan, E., and Singer, Y. 2011. Adaptive subgradient methods for online learning and stochastic optimization. J. Mach. Learn. Res. **12**(61): 2121–2159.

Esposito, F., Rufino, G., Moccia, A., Donnarumma, P., Esposito, M., and Magliulo, V. 2007. An integrated electro-optical payload system for forest fires monitoring from airborne platform. *In* IEEE Aerospace Conference, Big Sky, Mont., USA, 3–10 March 2007. pp. 1–13. doi: 10.1109/AERO.2007.353054.

Frizzi, S., Kaabi, R., Bouchouicha, M., Ginoux, J.-M., Moreau, E., and Fnaiech, F. 2016. Convolutional neural network for video fire and smoke detection. *In* 42nd Annual Conference of the IEEE Industrial Electronics Society, Florence, Italy, 23–26 October 2016. pp. 877–882. doi: 10.1109/IECON.2016.7793196.

Giglio, L. 2007. Characterization of the tropical diurnal fire cycle using VIRS and MODIS observations. Remote Sens. Environ. **108**(4): 407–421. doi: 10.1016/j.rse.2006.11.018.

Han, D., and Lee, B. 2006. Development of early tunnel fire detection algorithm using the image processing. *In* Advances in visual computing. *Edited by* G. Bebis, R. Boyle, B. Parvin, D. Koracin, P. Remagnino, A. Nefian, et al. Springer, Berlin, Heidelberg, Germany. pp. 39–48. doi: 10.1007/11919629_5.

Haralick, R.M., Shanmugam, K., and Dinstein, I. 1973. Textural features for image classification. IEEE Trans. Syst. Man Cybern. **SMC-3**(6): 610–621. doi: 10.1109/TSMC.1973.4309314.

Ho, C.-C. 2009. Machine vision-based real-time early flame and smoke detection. Meas. Sci. Technol. **20**(4): 045502. doi: 10.1088/0957-0233/20/4/045502.

Hossain, F.A., Zhang, Y.M., and Yuan, C. 2019. A survey on forest fire monitoring using unmanned aerial vehicles. *In* 3rd International Symposium on Autonomous Systems (ISAS), Shanghai, China, 29–31 May 2019. pp. 484–489. doi: 10.1109/ISASS.2019.8757707.

Hulens, D., Verbeke, J., and Goedemé, T. 2016. Choosing the best embedded processing platform for on-board UAV image processing. *In* Computer Vision, Imaging and Computer Graphics Theory and Applications, Berlin, Germany, 11–14 March 2015. *Edited by* J. Braz, J. Pettré, P. Richard, A. Kerren, L. Linsen, S. Battiato, and F. Imai. Springer International Publishing, Berlin, Germany. pp. 455–472. doi: 10.1007/978-3-319-29971-6_24.

Jaber, A., Guarnieri, F., and Wybo, J.L. 2001. Intelligent software agents for forest fire prevention and fighting. Saf. Sci. **39**(1): 3–17. doi: 10.1016/S0925-7535(01)00021-2.

Jiao, Z., Zhang, Y.M., Xin, J., Mu, L., Yi, Y., Liu, H., and Liu, D. 2019. A deep learning based forest fire detection approach using UAV and YOLOv3. *In* 1st International Conference on Industrial Artificial Intelligence (IAI). Shenyang, China, 23–27 July 2019. pp. 1–5. doi: 10.1109/ICIAI.2019.8850815.

Jolly, W.M., Cochrane, M.A., Freeborn, P.H., Holden, Z.A., Brown, T.J., Williamson, G.J., and Bowman, D.M.J.S. 2015. Climate-induced variations in global wildfire danger from 1979 to 2013. Nat. Commun. **6**(1): 7537. doi: 10.1038/ncomms8537. PMID: 26172867.

Khan, S., Rahmani, H., Shah, S.A.A., and Bennamoun, M. 2018. A guide to convolutional neural networks for computer vision. Synth. Lect. Comput. Vis. **8**(1): 1–207. doi: 10.2200/S00822ED1V01Y201712COV015.

Kim, B., and Lee, J. 2019. A video-based fire detection using deep learning models. Appl. Sci. **9**(14): 2862. doi: 10.3390/app9142862.

Kingma, D.P., and Ba, J. 2017. Adam: A method for stochastic optimization. ArXiv14126980 Cs. Available from http://arxiv.org/abs/1412.6980 [accessed 1 May 2020].

Kleinberg, B., Li, Y., and Yuan, Y. 2018. An alternative view: When does SGD escape local minima? *In* 35th International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2018. pp. 2698–2707. Available from http://proceedings.mlr.press/v80/kleinberg18a.html [accessed 1 May 2020].

Kolesov, I., Karasev, P., Tannenbaum, A., and Haber, E. 2010. Fire and smoke detection in video with optimal mass transport based optical flow and neural networks. *In* IEEE International Conference on Image Processing, Hong Kong, China, 26–29 September 2010. pp. 761–764. doi: 10.1109/ICIP.2010.5652119.

Krizhevsky, A., Sutskever, I., and Hinton, G.E. 2012. ImageNet classification with deep convolutional neural networks. *In* Advances in neural information processing systems. *Edited by* F. Pereira, C.J.C. Burges, L. Bottou, and K.Q. Weinberger. Curran Associates, Inc., Red Hook, N.Y., USA. Vol. 25, pp. 1097–1105. Available from http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf [accessed 20 September 2019].

Kyrkou, C., Plastiras, G., Theocharides, T., Venieris, S.I., and Bouganis, C.-S. 2018. DroNet: Efficient convolutional neural network detector for real-time UAV applications. *In* Design, Automation Test in Europe Conference Exhibition (DATE), Dresden, Germany, 19–23 March 2018. pp. 967–972. doi: 10.23919/DATE.2018.8342149.

Lai, C.L., Yang, J.C., and Chen, Y.H. 2007. A real time video processing based surveillance system for early fire and flood detection. *In* IEEE Instrumentation Measurement Technology Conference IMTC 2007, Warsaw, Poland, 1–3 May 2007. pp. 1–6. doi: 10.1109/IMTC.2007.379190.

Lee, W., Kim, S., Lee, Y.-T., Lee, H.-W., and Choi, M. 2017. Deep neural networks for wild fire detection with unmanned aerial vehicle. *In* IEEE International Conference on Consumer Electronics (ICCE), Las Vegas, Nev., USA, 8–10 January 2017. pp. 252–253. doi: 10.1109/ICCE.2017.7889305.

Li, K., and Malik, J. 2017. Learning to optimize neural nets. ArXiv170300441 Cs Math Stat. Available from http://arxiv.org/abs/1703.00441 [accessed 1 May 2020].

Liu, L., Jiang, H., He, P., Chen, W., Liu, X., Gao, J., and Han, J. 2019. On the variance of the adaptive learning rate and beyond. Available from https://openreview.net/forum?id=rkgz2aEKDr [accessed 1 May 2020].

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., and Berg, A.C. 2016. SSD: Single shot multibox detector. *In* Computer Vision — ECCV 2016. *Edited by* B. Leibe, J. Matas, N. Sebe, and M. Welling. Springer International Publishing, Cham, Switzerland. pp. 21–37. doi: 10.1007/978-3-319-46448-0_2.

Liu, Z.X., Zhang, Y.M., Yuan, C., Ciarletta, L., and Theilliol, D. 2019. Collision avoidance and path following control of unmanned aerial vehicle in hazardous environment. J. Intell. Robot. Syst. **95**(1): 193–210. doi: 10.1007/s10846-018-0929-y.

Loshchilov, I., and Hutter, F. 2016. SGDR: Stochastic gradient descent with warm restarts. Available from https://arxiv.org/abs/1608.03983v5 [accessed 1 May 2020].

Lu, Y., Xue, Z., Xia, G.-S., and Zhang, L. 2018. A survey on vision-based UAV navigation. Geo-Spat. Inf. Sci. **21**(1): 21–32. doi: 10.1080/10095020.2017.1420509.

McCulloch, W.S., and Pitts, W. 1943. A logical calculus of the ideas immanent in nervous activity. Bull. Math. Biophys. **5**(4): 115–133. doi: 10.1007/BF02478259.

McKendry, I.G., Christen, A., Lee, S.-C., Ferrara, M., Strawbridge, K.B., O'Neill, N., and Black, A. 2019. Impacts of an intense wildfire smoke episode on surface radiation, energy and carbon fluxes in southwestern British Columbia, Canada. Atmos. Chem. Phys. **19**(2): 835–846. doi: 10.5194/acp-19-835-2019.

Merino, L., Caballero, F., Martínez-De-Dios, J.R., Maza, I., and Ollero, A. 2012. An unmanned aircraft system for automatic forest fire monitoring and measurement. J. Intell. Robot. Syst. **65**(1–4): 533–548. doi: 10.1007/s10846-011-9560-x.

Mozaffari, M., Saad, W., Bennis, M., Nam, Y.-H., and Debbah, M. 2019. A tutorial on UAVs for wireless networks: Applications, challenges, and open problems. IEEE Commun. Surv. Tutor. **21**(3): 2334–2360. doi: 10.1109/COMST.2019.2902862.

Nesterov, Y. 1983. A method for unconstrained convex minimization problem with the rate of convergence $O(1/k^2)$. Dokl. USSR, **269**: 543–547.

Ojala, T., and Pietikäinen, M. 1999. Unsupervised texture segmentation using feature distributions. Pattern Recognit. **32**: 477–486. doi: 10.1016/S0031-3203(98)00038-7.

Partio, M., Cramariuc, B., Gabbouj, M., and Visa, A. 2002. Rock texture retrieval using gray level co-occurrence matrix. *In* Proceedings of the 5th Nordic Signal Processing Symposium, Trondheim, Norway, 4–7 October 2002.

Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. 2016. You only look once: Unified, real-time object detection. *In* Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 779–788.

Ren, S., He, K., Girshick, R., and Sun, J. 2015. Faster R-CNN: Towards real-time object detection with region proposal networks. *In* Advances in neural information processing systems. pp. 91–99.

San-Miguel-Ayanz, J., and Ravail, N. 2005. Active fire detection for fire emergency management: Potential and limitations for the operational use of remote sensing. Nat. Hazards, **35**(3): 361–376. doi: 10.1007/s11069-004-1797-2.

Sherstjuk, V., Zharikova, M., and Sokol, I. 2018. Forest fire monitoring system based on UAV team, remote sensing, and image processing. *In* IEEE Second International Conference on Data Stream Mining Processing (DSMP), Lviv, Ukraine, 21–25 August 2018. pp. 590–594. doi: 10.1109/DSMP.2018.8478590.

Simonyan, K., and Zisserman, A. 2014. Very deep convolutional networks for large-scale image recognition. ArXiv14091556 Cs. Available from http://arxiv.org/abs/1409.1556 [accessed 30 September 2019].

Sonoda, S., Ishikawa, I., Ikeda, M., Hagihara, K., Sawano, Y., Matsubara, T., and Murata, N. 2019. The global optimum of shallow neural network is attained by ridgelet transform. ArXiv180507517 Cs Stat. Available from http://arxiv.org/abs/1805.07517 [accessed 1 May 2020].

Stula, M., Krstinic, D., and Seric, L. 2012. Intelligent forest fire monitoring system. Inf. Syst. Front. **14**(3): 725–739. doi: 10.1007/s10796-011-9299-8.

Sun, H., Song, G., Wei, Z., Zhang, Y., and Liu, S. 2017. Bilateral teleoperation of an unmanned aerial vehicle for forest fire detection. *In* IEEE International Conference on Information and Automation (ICIA), Macau, China, 18–20 July 2017. pp. 586–591. doi: 10.1109/ICInfA.2017.8078976.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. 2016. Rethinking the inception architecture for computer vision. *In* Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2818–2826.

The Canadian Press. 24 June 2020. Major Quebec forest fire under control; province lifts ban on open fires. Available from https://montreal.ctvnews.ca/major-quebec-forest-fire-under-control-province-lifts-ban-on-open-fires-1.4998957 [accessed 26 June 2020].

Topi, M., Timo, O., Matti, P., and Maricor, S. 2000. Robust texture classification by subsets of local binary patterns. *In* Proceedings of the 15th International Conference on Pattern Recognition, Barcelona, Spain, 3–7 September 2000. Vol. 3, pp. 935–938. doi: 10.1109/ICPR.2000.903698.

Transport Canada. 18 October 2018. Flying your drone safely and legally. Available from https://www.tc.gc.ca/en/services/aviation/drone-safety/flying-drone-safely-legally.html [accessed 7 May 2020].

Vardoulakis, S., Marks, G., and Abramson, M.J. 2020. Lessons learned from the Australian Bushfires: Climate change, air pollution, and public health. JAMA Intern. Med. **180**(5): 635–636. doi: 10.1001/jamainternmed.2020.0703. PMID: 32108854.

Vicente, J., and Guillemant, P. 2002. An image processing technique for automatically detecting forest fire. Int. J. Therm. Sci. **41**(12): 1113–1120. doi: 10.1016/S1290-0729(02)01397-2.

Wang, X., Parisien, M.-A., Taylor, S.W., Candau, J.-N., Stralberg, D., Marshall, G.A., et al. 2017. Projected changes in daily fire spread across Canada over the next century. Environ. Res. Lett. **12**(2): 025005. doi: 10.1088/1748-9326/aa5835.

Wardihani, E.D., Ramdhani, M., Suharjono, A., Setyawan, T.A., Hidayat, S.S., Helmy, S.W., et al. 2018. Real-time forest fire monitoring system using unmanned aerial vehicle. J. Eng. Sci. Technol. **13**(6): 1587–1594.

Wittkopp, T., Hecker, C., and Opitz, D. 2001. The Cargo Fire Monitoring system (CFMS) for the visualisation of fire events in aircraft cargo holds. NIST Special Publication (SP). pp. 665–676.

Wong, S.D., Broader, J.C., and Shaheen, S.A. 2020. Review of California wildfire evacuations from 2017 to 2019. UC Office of the President: University of California Institute of Transportation Studies. doi: 10.7922/G29G5K2R.
Yu, C., Mei, Z., and Zhang, X. 2013. A real-time video fire flame and smoke detection algorithm. Procedia Eng. **62**: 891–898. doi: 10.1016/j.proeng.2013.08.140.
Yu, Z.Q., Zhang, Y.M., Liu, Z.X., Qu, Y., Su, C.-Y., and Jiang, B. 2019. Decentralized finite-time adaptive fault-tolerant synchronization tracking control for multiple UAVs with prescribed performance. J. Frankl. Inst. doi: 10.1016/j.jfranklin.2019.11.056.
Yuan, C., Zhang, Y.M., and Liu, Z.X. 2015. A survey on technologies for automatic forest fire monitoring, detection, and fighting using unmanned aerial vehicles and remote sensing techniques. Can. J. For. Res. **45**(7): 783–792. doi: 10.1139/cjfr-2014-0347.
Yuan, C., Liu, Z.X., and Zhang, Y.M. 2017. Aerial images-based forest fire detection for firefighting using optical remote sensing techniques and unmanned aerial vehicles. J. Intell. Robot. Syst. **88**(2–4): 635–654. doi: 10.1007/s10846-016-0464-7.
Yuan, C., Liu, Z.X., and Zhang, Y.M. 2019. Learning-based smoke detection for unmanned aerial vehicles applied to forest fire surveillance. J. Intell. Robot. Syst. **93**(1–2): 337–349. doi: 10.1007/s10846-018-0803-y.
Yuan, F. 2010. An integrated fire detection and suppression system based on widely available video surveillance. Mach. Vis. Appl. **21**(6): 941–948. doi: 10.1007/s00138-010-0276-x.
Zeiler, M.D., and Fergus, R. 2014. Visualizing and understanding convolutional networks. *In* ECCV: European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014. *Edited by* D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars. Springer International Publishing, Cham, Switzerland. pp. 818–833. doi: 10.1007/978-3-319-10590-1_53.
Zhang, Y., Liang, P., and Charikar, M. 2017. A hitting time analysis of stochastic gradient Langevin dynamics. *In* Conference on Learning Theory, Amsterdam, the Netherlands, 7–10 July 2017. pp. 1980–2022. Available from http://proceedings.mlr.press/v65/zhang17b.html [accessed 1 May 2020].
Zhang, Y.M., and Li, X.R. 1999. A fast U-D factorization-based learning algorithm with applications to nonlinear system modeling and identification. IEEE Trans. Neural Netw. **10**(4): 930–938. doi: 10.1109/72.774266. PMID: 18252590.
Zhao, Y., Ma, J., Li, X., and Zhang, J. 2018. Saliency detection and deep learning-based wildfire identification in UAV imagery. Sensors, **18**(3): 712. doi: 10.3390/s18030712. PMID: 29495504.