Wildfire Flame and Smoke Detection Using Static Image Features and Artificial Neural Network

F M Anim Hossain[1], Youmin Zhang[2 *], Chi Yuan[3], and Chun-Yi Su[4]
Department of Mechanical, Industrial and Aerospace Engineering
Concordia University
Montreal, Quebec H3G 1M8, Canada
f.hossain@myport.ac.uk[1], youmin.zhang@concordia.ca[2], chiyuan996@gmail.com[3], chun-yi.su@concordia.ca[4]

***Abstract*—If forest fires are not contained quickly, they can spread wide very fast and cause devastating environmental, social and economic damages. The best method to minimize wildfire loss is to be able to detect it in its early stages for rapid containment and suppression. Fire comes with some distinguishable signatures such as flame, smoke and heat that can be used for early detection using computer vision based remote sensing techniques. Each signature has its own merits and demerits that vary under different environmental conditions and circumstances. Therefore, it is not always enough to form a detection algorithm based on a single signature. Keeping that in mind, this paper presents a novel algorithm that is capable of detecting both flame and smoke from a single image using block-based color features, texture features and a single artificial neural network (ANN). Such an algorithm is capable of providing reliable, rapid and continuous detection under any circumstances and can be incorporated into the existing unmanned aerial vehicle (UAV) based fire monitoring system.**

Keywords—Forest fire, flame detection, smoke detection, neural network, computer vision, remote sensing

I. Introduction

Forests keep our ecosystem balanced, climate change under control, provide shelter and they are a major source of food, medicine and jobs. It is evident that preserving the current forestry around the globe as well as maintaining a healthy growth of wildlands is of utmost importance for the world. However, the opposite is happening in reality and large-scale wildfire is one of the primary reasons behind it [1]. Over the past two decades, the occurrence of wildfire has become more frequent, affecting the lives of thousands of people and burning through millions of acres of lands every year [2]. As the ignition of forest fire is both a natural (lightning, volcanic eruption, meteor shower etc.) and man-made (bonfire, prescribed fire going wrong etc.) phenomenon, prevention is practically impossible [3]. The best course of action to fight forest fire is to detect fire at its early stage so that the authorities can be notified early and quickly.

Traditionally, fire has been detected by smoke detectors and thermal sensors. But these types of sensors are 'point sensors' that require smoke particles to reach the sensor or the temperature to rise significantly for a fire to be detected [4]. By the time that happens, the fire is already quite big. In addition, they fail to provide the exact location, size and propagation direction of the fire. Modern remote sensing using commercially-off-the-shelf CCD cameras can overcome all of the limitations of traditional methods through computer vision techniques [4]. Strategically positioned cameras attached in watch towers or payloads of UAVs can provide early detection, flexibility, large field of view, the exact location of the fire, size, rate of spread and propagation direction.

Three key signatures of fire that are useful for computer vision detection techniques are flame, smoke and heat [5]. Fire radiates heat, and according to Planck's law, the wavelength of radiation varies with temperature. As temperature due to the fire increases, the radiated electromagnetic wavelength gets smaller and at a certain temperature, it falls within the visible range and that visible portion is known as flame [5]. Flame has some distinct properties such as color, flickering, increasing in size over time and being illuminant. These make flame a strong signature for fire detection using computer vision. Conversely, in case of forest fire, the flame may be hidden under heavy forest canopy or smoke and may not be visible until it reaches the forest crown. That is not very effective for early detection. This limitation of flame can be addressed by using smoke signature because smoke becomes visible much earlier than flame and moves upwards very quickly. However, smoke color varies with fuel type and temperature, propagation depends on wind direction and speed. In addition, smoke's color and texture are too similar to other natural phenomena such as fog, clouds, steam etc. Moreover, smoke cannot be detected during nighttime operations. Heat detection is robust to illumination changes and forest canopy/smoke occlusion but it requires expensive infrared cameras and they are prone to false alarms due to solar reflection and other hot objects in nature. Furthermore, infrared cameras require the heat radiation to reach the sensors, therefore, often need to be in proximity to the fire source to be detected [4], [5].

Therefore, using a single signature may not yield the best result for early and reliable fire detection. Having multiple signatures makes the system robust to changes in illumination, allows continuous day/night operation and provides more confidence in the decision provided by the algorithm. Keeping this in mind, this paper introduces a novel algorithm that uses both flame and smoke to detect forest fire from a single static image using an artificial neural network (ANN). The following sections will highlight existing multi-signature fire detection methods, the methodology of the system and the preliminary results obtained from the developed algorithm.

II. Literature Review

Over the past two decades, fire detection with the help of computer vision has become popular due to the advances in electronics, computers and sensors. Detailed review on computer vision based fire detection are reported in [4]-[6]. Among all the reported works, only a handful of them have fused on flame and smoke signatures for fire detection.

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

The first fire detection method using both flame and smoke was reported in the year 2001 [7]. The authors developed the system to detect fire in airplane cargo bays. Their method measured the mean pixel value and the standard deviation of grayscale images with the idea that the presence of flame will increase the mean intensity and standard deviation of the image. To enhance the visibility of smoke, they subtracted the actual images from a reference image and superimposed the result on the actual image. This method showed the benefits of using multiple signatures to detect fire but it was developed for an indoor environment only.

Chen *et al* [8] developed a multi-signature algorithm using color rules for flame and smoke detection. They established color rules in RGB and HSI color space with empirically set thresholds followed by consecutive frame differencing for pixel growth monitoring. However, their report did not present any results of smoke detection. Additionally, they used RGB color space, where pixel values change under different illumination and red channel saturates when a flame is present, making the system prone to false alarms.

The flame and smoke detection model proposed in [9] is optimized for fire detection in tunnels where separate algorithms were used for flame and smoke. For flame detection, they thresholded a reference and input images and then subtracted the binary images to find the potential flame region.

Another empirically set rule-based thresholding algorithm is proposed in [10]. The authors used frame differencing, RGB-YCbCr color rules, corner angle variation of candidate region and variation in R color channel for flame detection. The subsequent part of the system used histogram variation, wavelet energy and correlation with background image for smoke detection.

While rule-based algorithms presented are computationally inexpensive, they are prone to misclassification. On the other hand, using machine learning algorithms such as support vector machine, neural network, Bayes classifier, fuzzy logic etc. can make the algorithms more robust under a wide set of scenarios. Ho [11] used a fuzzy logic classifier instead of empirically set threshold to detect flame and smoke. The overall algorithm uses a motion history image and HSI color space to identify a candidate region followed by flicker analysis in the candidate region. Additionally, after detection, the continuously adaptive mean shift (CAMSHIFT) algorithm is applied to track the detected fire region. The author reported 87% and 77% flame and smoke detection rate respectively and 20ms detection time. Fuzzy logic classifier was also used in [12] to detect flame in YCbCr color space and rule-based thresholding in RGB to detect smoke.

Yuan [13] used two separate models for flame and smoke. For flame, Gaussian mixture model in RGB color channel combined with frame differencing was used. For smoke detection, the sum of absolute difference (SAD) was used to compute motion orientation histogram and heuristic rules for color segmentation. A Bayes classifier was used for the final decision. This algorithm was trained and tested with both indoor and outdoor images and the results were satisfactory although the complexity of the system could make it difficult for real-time computation.

Another machine learning based model presented in [14] used ANN. They used two ANNs: one for flame and one for smoke and optimal mass transport-optical flow as features. Their report showed promising results but the classifiers were trained and tested with a limited dataset. ANN and optical flow method for smoke recognition are also applied in [15]. Their flame recognition method used a separate algorithm where candidate region was segmented using frame differencing and HSI color rules followed by dividing the frames into 8×8 blocks and analyzing the flicker frequency of each block. However, the use of temporal information in the form of optical flow makes these methods computationally expensive.

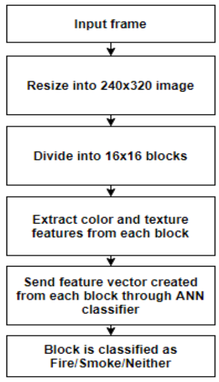


Fig. 1. Flow diagram of the proposed method

Recently, Frizzi *et al* [16] developed a flame/smoke detection algorithm using an emerging new power image classification tool called convolutional neural network (CNN). They have managed to achieve a remarkably high flame and smoke detection rate using this method. However, CNN training time is very high and it is unclear if their system detects fire and smoke using a single CNN or multiple CNNs. Nonetheless, this powerful image classification tool can have a significant impact on future researches in this area.

The relevant literature presented in this sector highlight the methods already implemented along with their strengths and weaknesses. A trend can also be noticed that over the last few years, the classification methods have leaned more towards machine learning from heuristic rules. Keeping with the trend, the proposed novel algorithm has the capability of detecting flame and smoke from the same image using a single multiclass ANN classifier that attempts to be robust under different scenarios and provides real-time classification.

III. Methodology

The overall system diagram is presented in Fig. 1. The input images will be first resized into 240×320 resolution for ease of computation because ultimately, the objective is to incorporate this algorithm into an unmanned aerial vehicle. The resized image will be divided into 16×16 blocks and features will be extracted from each block to create feature vectors. Each feature vector from the 300 blocks will go through an ANN and the results will display whether the blocks are fire, smoke or normal background blocks.

Different from existing methods, our proposed method aims to identify flame and smoke using a single image and a single algorithm. Therefore, it is important to find common features that can distinctly represent fire and smoke. To reduce computational complexity, temporal features such as flickering analysis, growth, and movement etc. have been ignored and only extract color and texture information.

A. *Extraction of Color Features*

Color is one of the most distinguishing features of flame. Flame color ranges from red to yellow and to white. Although RGB is the most popular color space for fire detection, it comes with a few limitations. RGB color channel does not spate chrominance from luminance making them prone to

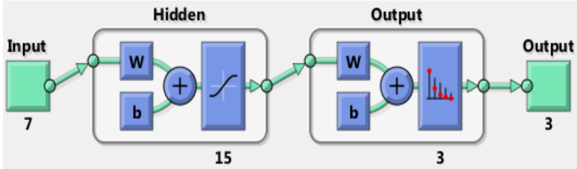


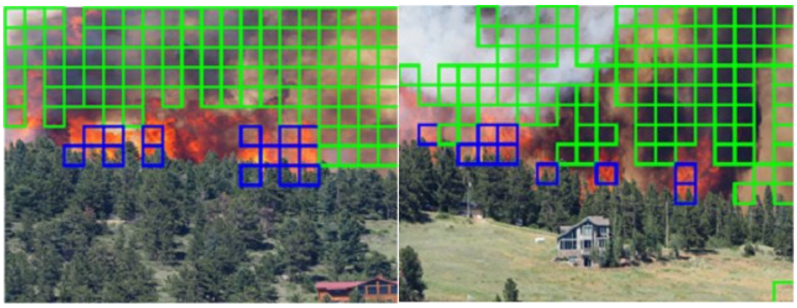
Fig. 2. Neural network structure of the system

(a) (b) (c)

(d) (e)

Fig. 3. Original test images

(a) (b) (c)



(d) (e)

Fig. 4. Result images: Green blocks represent detected smoke, blue blocks represent detected flame

changes under different illumination. Additionally, the red channel value saturates when the flame temperature increases making them indistinguishable from other high luminance objects in nature [17]. A comprehensive study on different color spaces show that flame color is most distinguishable in YCbCr and HSI color spaces [18]. However, smoke color has a wide range from white to gray and to black making it difficult to distinguish [15]. Many researchers have used HSI color space in their algorithms to segment smoke with considerable success [4], [8], [11]. Considering all these factors, YCbCr and HSI color spaces were chosen as they can both provide distinction among smoke, flame and background. For the purpose of the work, the mean values of Cb (and variance), Cr, H and S color channels of each block were extracted.

B. *Extraction of Texture Features*

While color provides a good general distinction among smoke, flame and most background, it is not enough to reliably classify flame and smoke because there are plenty of other objects in nature that has a similar color as flame and smoke. Red/orange/yellow flowers, setting sun, tree leaves in fall, red/orange/yellow cars are too similar to flame color. However, the texture of these other objects is usually plain and smooth giving them color consistency throughout. On the other hand, the texture of flame and smoke fluctuates, giving them fluctuating pixel values within a block.

One of the most popular texture extraction techniques is using a gray level co-occurrence matrix (GLCM) where the relative frequency of two neighboring pixel values, separated by a certain distance and angle, is represented in a matrix form [19]. The size of the matrix depends on the number of gray levels. For example, if the number of gray levels is 256, then there will be a 256×256 matrix where the first column of the first row will represent how many times in the region under observation, the pair with pixel value (0,0) has appeared. GLCM has been utilized to extract texture features for fire detection in a few researches [20]-[23]. From GLCM, numerous texture statistics can be computed that described the texture of the object under observation such as entropy, contrast, variance, dissimilarity etc. [24].

In our work, the GLCM was computed by first converting the image to grayscale image, using 256 gray levels and creating the matrix using a distance of 1 pixel to the right and in 90º direction. From the GLCM, contrast and variance of each block was extracted as texture features.

C. *Artificial Neural Network*

ANN is a machine learning algorithm that tries to mimic the performance of the human brain and has become very popular over the last two decades due to the emergence of powerful computers. ANN allows parallel computation, can be implemented in analog VLSI, makes no assumption on input pattern classes, can simulate convoluted decision boundaries, input data units can be sparse, can automatically integrate the relationship between the dependent and independent variables and has the robustness to multicollinearity [25], [26]. These advantages make ANN a very powerful image classification tool that works with satisfactory computational efficiency.

The general structure of ANN has three key components, an input layer, hidden layers and an output layer. Within the layers, there are numbers of nodes or 'neurons'. The number of the neurons in the input and output layers are the same as the input features and output classes respectively while the hidden layers' number of nodes are user-defined. The neurons of the input layer hold the mathematical representation of each input feature and afterward, neurons in each layer obtain information from the previous layer, process them and propagate it to the next layer. This process is called forward propagation. During forward propagation, the hidden layers and the output layer perform a linear operation as follows:

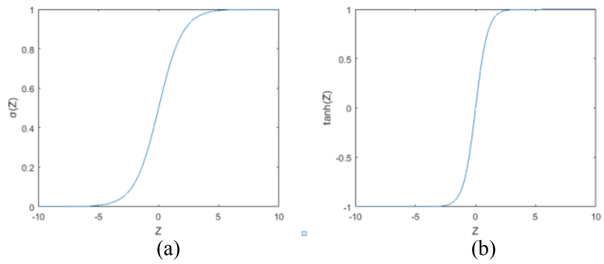


Fig. 5. (a) output of sigmoid activation, (b) output of tanh activation

$$Z^L = W^L A^{L-1} + b^L \tag{1}$$

Equation (1) is represented in a vectorized form for an entire layer 'L'. Here, 'W' is a matrix, 'b' is a vector and they are parameters of the neural network known as weights and biases, respectively. A^{L-1} is a column vector that brings information from the previous layer, *L-1*. However, such linear operation can only result in a linear decision boundary, which is not enough when the relationship between input features and output classes is non-linear. Therefore, the neurons usually perform a non-linear operation on 'Z' following the linear operation. This non-linear operator is called an 'activation function'. There are different types of activation functions such as sigmoid, hyperbolic tangent (tanh), softmax, ReLU etc. The sigmoid function is one of the most common activation functions and performs the following operation [27]:

$$A^L(Z) = \sigma(Z^L) = \frac{1}{1+e^{-Z^L}} \tag{2}$$

From understanding the nature of (2) and from Fig. 5(a), it can be understood that the output of a layer L, A^L will be within 0 and 1. However, for a neural network to learn faster and perform better, it is better to have zero mean shifted inputs [28]. A variation of the sigmoid activation, the hyperbolic tangent (tanh) activation function performs the non-linearity as follows [27]:

$$A^L(Z) = tanh(Z^L) = \frac{e^{Z^L}-e^{-Z^L}}{e^{Z^L}+e^{-Z^L}} \tag{3}$$

Which results in an output that has a mean of 0 and ranges between -1 and +1 (Fig. 5(b)), making it a better choice for the hidden layers. As the output of sigmoid activation ranges between 0 and 1 it is ideal for output layers. However, in multi-class classification problems, sigmoid activation output can predict more than 0.5 probability for multiple classes within the same observation, creating confusion about the prediction. Softmax activation removes this problem by performing the following operation [29]:

$$A_i^L(Z_i^L) = \frac{e^{Z_i^L}}{\sum_{j=1}^{n} e^{Z_j^L}} \tag{4}$$

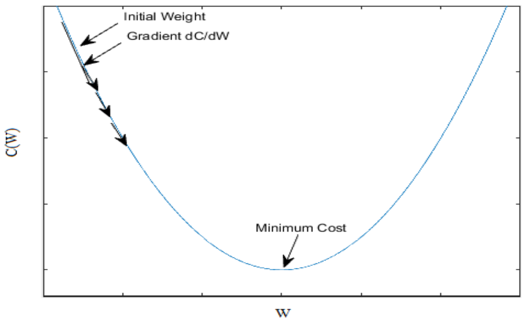


Fig. 6. Gradient descent intuition

Here, A_i^L is the output of i-th unit in layer L. Z_i^L is the i-th value of Z in layer L and 'n' is the total number of nodes in layer L. it can be understood from (3) that softmax creates a probability distribution of the results that sum up to 1.

Before the neural network can predict outputs from test dataset using forward propagation, it needs to be trained first. The procedure of training a neural network is called backpropagation, where contrary to forward propagation, the process starts at the output layer and propagates back to end at the first layer. At the beginning of the training process, all the weights and biases of the neural network need to be randomly initialized. The objective of the training process is to find weights and biases that minimize the cost function. Cost function is a mathematical operator like mean square error or cross-entropy that determines the accuracy of the neural network by comparing the predicted output with the ground truth. For the purpose of classification, cross-entropy is a more effective method compared to mean square error. The mathematical operation of cross-entropy is as follows:

$$C(y^p, y^d) = \frac{1}{2m}\sum_{m=1}^{n} -y_m^d \log(y_m^p) - (1 - y_m^d)\log(1 - y_m^p) \quad (5)$$

where y^d is the desired output, y^p is the predicted output of the network and m is the total number of observations. It can be noted that as y^p is a function of W and b, we can write the cost function as $C(W, b)$. One of the most popular methods of finding the weights and biases that minimize the cost function is using gradient descent. The general formula for gradient descent is as follows [28]:

$$W = W_c - \mu\frac{\partial C}{\partial W_c} \quad (6)$$

Fig. 6 provides a simple overview of gradient descent. Assuming that there is only one neuron in the network, only one weight and a convex cost graph, a weight vs cost function graph like Fig. 5 can be imagined. Using gradient descent, moving along the direction of the gradient of current weight W_c, eventually, the algorithm will learn the weight that generates the minimum cost function. The speed of convergence depends on another hyperparameter, μ, which is the learning rate of the algorithm. If it is set too small, its convergence will take longer and if it is set too large, then W_c will keep oscillating and may never reach convergence [28].

In this algorithm, there is a 7 nodes input layer, 3 nodes output layer (because the system output will show whether the block under consideration is fire, smoke or neither) and 15 nodes for the single hidden layer giving it a 7-15-3 neural network structure. The illustration of the network architecture is presented in Fig. 2. Activation functions of the hidden layer and the output layer are tanh and softmax, respectively. The network has been trained using cross-entropy loss function of (5) and a variation of gradient descent called scaled conjugate descent (SGD), which generally has a faster convergence rate than standard gradient descent [30].

IV. Experiments and Results

The system was trained with 20 images containing images of different scenarios such as flames of different colors and sizes, smokes with different colors and sizes, images containing both flame and smoke, normal background images, flame aliases such as yellow/red flowers and cars etc. Each of the 240×320 resolution images has 300 16×16 blocks. Therefore, a total of 6,000 blocks were used to train the images using backpropagation. MATLAB 2015b was used to design the neural network, perform training and testing of the algorithm using its built-in functions. Training and testing images were gathered from the internet.

Sample test images are presented in Fig. 3 with flame and smoke under different scenarios, illumination, distance, color and shape. The corresponding results are presented in Fig. 4 where the green blocks represent the blocks classified as smoke and the blue ones are classified as flame.

In Fig. 3(a), the image contains a yellow flame that is partially occluded by trees. The subsequent result image presented in Fig. 4(a) shows that the system has successfully detected the flame blocks around the trees including the isolated smaller ones. The thick black part of the smoke is detected and the black road is successfully classified as normal background.

A different scenario has been presented in Fig. 3(b) as a night-time fire without smoke. The illumination from the fire gives the surrounding a similar red-orange glow. The result of this test image in Fig. 4(b) shows that the fire blocks have been correctly detected and fire aliases such as the glowing surrounding and the yellow road sign have been successfully detected as normal background and no block has been classified as smoke. However, three blocks have been misclassified as flame.

Fig. 3(c) displays a small fire from a fairly short distance under low illumination and white-black smoke. The results in Fig. 4(c) show that the fire blocks, the thick black smoke on top the fire source and the thick white smoke on the left side of the image have been successfully detected but the transparent smoke through which the background can be seen could not be detected. No background blocks have been misclassified as flame/smoke.

Both test images in Fig. 3(d) and Fig. 3(e) are taken from a distance that displays a crown fire with partially visible flame and heavy smoke. Both flame and smoke display a wide range of color in these images. The corresponding results in Fig. 4(d) and Fig. 4(e) show that most of the flame and smoke blocks, present in the images, have been properly classified. This type of image can properly represent a real scenario taken from a UAV where the flame is occluded by the canopy. The advantage of our algorithm is evident in such a case where the UAV can raise a flag from the initial detection of smoke and confirm the presence of fire by detecting flame as it becomes visible.

Analysis of all the test images shows that the algorithm has a 100% detection rate when fire, smoke or both are present in the image. This means that if flame and smoke are present in the image, it has been able to accurately classify at least one block as flame or smoke. If every block in the image is considered, the rate of correctly classified blocks, commonly known as the true positive rate, is 89% for flame, 80.7% for smoke and 87.4% for background blocks. The average accuracy of the system is 84.8%.

V. Conclusions and Future Works

A novel algorithm that uses a single neural network with multi-class detection ability to detect flame and smoke has been presented in this paper. The preliminary results of the algorithm show promising results that can successfully detect fire through flame and smoke identification. Additionally, the lack of temporal feature in this algorithm means that it does not need to process multiple frames for detection, saving computation and storage needs. This algorithm can be incorporated into any existing fire monitoring platform such as watch tower, manned and unmanned aerial vehicles. The use of multiple fire signatures makes it possible to detect fire as early as possible, makes the system robust to illumination changes and can provide continuous monitoring throughout day and night.

The results, achieved in the preliminary stage, indicate that it is worth investigating further. One of the future works will involve training the algorithm with more images under different conditions. Eventually, the objective is to increase the number of classes to differentiate between flame and flame alias. Potential future work also involves improving the results by experimenting with different spatial and texture features. Once the algorithm is optimized, it will be incorporated into a UAV and tested it in real time in a laboratory setup. Improving smoke detection will be one of the primary focuses to increase the overall detection rate. Modifying this method with other machine learning algorithms such as support vector machine and convolutional neural network is also under consideration, to find the optimal algorithm for reliable and real-time fire detection. Such a system can have a significant impact in the ongoing fire detection research using autonomous vehicles in terms of accuracy, computation and expense.

References

[1] G. M. Budd *et al.*, "Project Aquarius 4: experimental bushfires, suppression procedures, and measurements," *Int. J. Wildland Fire*, vol. 7, no. 2, pp. 99–104, 1997.

[2] S. H. Doerr. and C. Santín, "Global trends in wildfire and its impacts: perceptions versus realities in a changing world," *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 371, no. 1696, p. 20150345, Jun. 2016.

[3] P. A. Thomas and R. S. McAlpine, "Fire in the wild landscape," in *Fire in the Forest*, 1st ed., Cambridge, United Kingdom: Cambridge University Press, 2010, p. 54.

[4] A. E. Çetin *et al.*, "Video fire detection – Review," *Digital Signal Processing*, vol. 23, no. 6, pp. 1827–1843, Dec. 2013.

[5] R. S. Allison, J. M. Johnston, G. Craig, and S. Jennings, "Airborne optical and thermal remote sensing for wildfire detection and monitoring," *Sensors*, vol. 16, no. 8, p. 1310, Aug. 2016.

[6] C. Yuan, Y. Zhang, and Z. Liu, "A survey on technologies for automatic forest fire monitoring, detection, and fighting using unmanned aerial vehicles and remote sensing techniques," *Can. J. For. Res.*, vol. 45, no. 7, pp. 783–792, Mar. 2015.

[7] T. Wittkopp, C. Hecker, and D. Opitz, "Cargo Fire Monitoring System (CFMS) for the visualisation of fire events in aircraft cargo holds," in the *Proceedings of the 12th International Conference on Automatic Fire Detection* (AUBE'01), Gaithersburg, MD, USA, 2001.

[8] T.-H. Chen, P.-H. Wu, and Y.-C. Chiou, "An early fire-detection method based on image processing," in the *2004 International Conference on Image Processing, 2004. ICIP '04.*, 2004, vol. 3, pp. 1707-1710.

[9] D. Han and B. Lee, "Development of early tunnel fire detection algorithm using the image processing," in *Advances in Visual Computing*, 2006, pp. 39–48.

[10] C. L. Lai, J. C. Yang, and Y. H. Chen, "A real time video processing based surveillance system for early fire and flood detection," in the *2007 IEEE Instrumentation & Measurement Technology Conference IMTC 2007*, pp. 1–6, 2007.

[11] C.-C. Ho, "Machine vision-based real-time early flame and smoke detection," *Meas. Sci. Technol.*, vol. 20, no. 4, p. 045502, Mar. 2009.

[12] T. Çelik, H. Özkaramanli, and H. Demirel, "Fire and smoke detection without sensors: Image processing based approach," *2007 15th European Signal Processing Conference*, pp. 1794–1798, 2007.

[13] F. Yuan, "An integrated fire detection and suppression system based on widely available video surveillance," *Machine Vision and Applications*, vol. 21, no. 6, pp. 941–948, Oct. 2010.

[14] I. Kolesov, P. Karasev, A. Tannenbaum, and E. Haber, "Fire and smoke detection in video with optimal mass transport based optical flow and neural networks," in the *2010 IEEE International Conference on Image Processing*, 2010, pp. 761–764.

[15] C. Yu, Z. Mei, and X. Zhang, "A real-time video fire flame and smoke detection algorithm," *Procedia Engineering*, vol. 62, pp. 891–898, Jan. 2013.

[16] S. Frizzi, R. Kaabi, M. Bouchouicha, J. Ginoux, E. Moreau, and F. Fnaiech, "Convolutional neural network for video fire and smoke detection," in the *42nd Annual Conference of the IEEE Industrial Electronics Society*, 2016, pp. 877–882.

[17] T. Çelik and H. Demirel, "Fire detection in video sequences using a generic color model," *Fire Safety Journal*, vol. 44, no. 2, pp. 147–158, Feb. 2009.

[18] S. Rudz, K. Chetehouna, A. Hafiane, O. Sero-Guillaume, and H. Laurent, "On the evaluation of segmentation methods for wildland fire," in *Advanced Concepts for Intelligent Vision Systems*, 2009, pp. 12–23.

[19] D. J. Marceau, P. J. Howarth, J. M. M. Dubois, and D. J. Gratton, "Evaluation of the grey-level co-occurrence matrix method for land-cover classification using SPOT imagery.," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 28, no. 4, pp. 513–519, 1990.

[20] Y. Cui, H. Dong, and E. Zhou, "An early fire detection method based on smoke texture analysis and discrimination," in the *2008 Congress on Image and Signal Processing*, 2008, vol. 3, pp. 95–99.

[21] A. Chenebert, T. P. Breckon, and A. Gaszczak, "A non-temporal texture driven approach to real-time fire detection," in the *18th IEEE International Conference on Image Processing*, 2011, pp. 1741–1744.

[22] J. Zhao, Z. Zhang, S. Han, C. Qu, Z. Yuan, and D. Zhang, "SVM based forest fire detection using static and dynamic features," *ComSIS*, vol. 8, no. 3, pp. 821–841, 2011.

[23] Y. Chunyu, Z. Yongming, F. Jun, and W. Jinjun, "Texture analysis of smoke for real-time fire detection," in *2009 Second International Workshop on Computer Science and Engineering*, 2009, vol. 2, pp. 511–515.

[24] D. A. Clausi, "An analysis of co-occurrence texture statistics as a function of grey level quantization," *Canadian Journal of Remote Sensing*, vol. 28, no. 1, pp. 45–62, Jan. 2002.

[25] D. Bejou, B. Wray, and T. N. Ingram, "Determinants of relationship quality: An artificial neural network analysis," *Journal of Business Research*, vol. 36, no. 2, pp. 137–143, Jun. 1996.

[26] K. K. Mohanty and T. J. Majumdar, "An artificial neural network (ANN) based software package for classification of remotely sensed data," *Computers & Geosciences*, vol. 22, no. 1, pp. 81–87, Feb. 1996.

[27] B. Karlik and A. V. Olgac, "Performance analysis of various activation functions in generalized MLP architectures of neural networks," *International Journal of Artificial Intelligence and Expert Systems*, vol. 1, no. 4, pp. 111–122, 2011

[28] Y. A. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller, "Efficient BackProp," in *Neural Networks: Tricks of the Trade: Second Edition*, G. Montavon, G. B. Orr, and K.-R. Müller, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 9–48.

[29] S. Kanai, Y. Fujiwara, Y. Yamanaka, and S. Adachi, "Sigsoftmax: Reanalysis of the Softmax Bottleneck," in *Advances in Neural Information Processing Systems 31*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds. Curran Associates, Inc., 2018, pp. 286–296.

[30] M. F. Møller, "A scaled conjugate gradient algorithm for fast supervised learning," *Neural Networks*, vol. 6, no. 4, pp. 525–533, Jan. 1993.