EdgeFireSmoke: A Novel Lightweight CNN Model for Real-Time Video Fire–Smoke Detection

Jefferson Silva Almeida, Chenxi Huang, Fabrício Gonzalez Nogueira, Surbhi Bhatia, and Victor Hugo C. de Albuquerque, *Senior Member, IEEE*

***Abstract*—The planet Earth is being affected by a series of wildfires, which have been steadily increasing over the last two decades. Forests have undergone deforestation due to natural forest fires and forest fires caused by man. These events are occurring on a global scale, and in Brazil, these wildfires are having an extreme impact on the Amazon forest as well as other forest biomes. This article proposes a novel lightweight convolutional neural network (CNN) model for wildfire detection through RGB images. This new method presents more advantages than the other methods used for the same task. Our CNN architecture can be used with aerial images from unmanned aerial vehicles and from video surveillance systems, combined with edge computing devices for image processing with a CNN. The proposed system is able to send wildfire alerts. The images do not have to be sent to a cloud computer as they can be processed in an edge device. However, it sends a string of alerts whenever a wildfire is detected. The evaluation of our proposed method showed that it required about 30 ms for the classification time, per image, and achieved an accuracy of 98.97% and an $F1$-score of 95.77%, which is a very promising result.**

***Index Terms*—Deep learning, edge devices, fire–smoke detection, Internet of Things (IoT).**

Manuscript received 14 September 2021; revised 20 October 2021 and 18 November 2021; accepted 22 December 2021. Date of publication 19 January 2022; date of current version 9 September 2022. This work was supported in part by the Brazilian Federal Agency for Support and Evaluation of Graduate Education (CAPES). The work of Jefferson Silva Almeida was supported in part by CAPES and in part by the Federal University of Ceará. Paper no. TII-21-3971. *(Corresponding author: Chenxi Huang.)*

Jefferson Silva Almeida and Fabrício Gonzalez Nogueira are with the Department of Electrical Engineering, Federal University of Ceará, Fortaleza 60020-181, Brazil (e-mail: jeffersonsilva@lapisco.ifce.edu.br; fnogueira@dee.ufc.br).

Chenxi Huang is with the School of Informatics, Xiamen University, Xiamen 361005, China (e-mail: supermonkeyxi@xmu.edu.cn).

Surbhi Bhatia is with the Department of Information Systems, College of Computer Science and Information Technology, King Faisal University, Al Hofuf 31982, Saudi Arabia (e-mail: sbhatia@kfu.edu.sa).

Victor Hugo C. de Albuquerque is with the Department of Teleinformatics Engineering, Federal University of Ceará, Fortaleza 60020-181, Brazil (e-mail: victor.albuquerque@ieee.org).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TII.2021.3138752.

Digital Object Identifier 10.1109/TII.2021.3138752

I. Introduction

The occurrence of forest fires registered on our planet in 2020 has been the greatest in scale and in CO_2 emissions over the past two decades [1]. According to the World Health Organization, wildfires and volcanic activities affected 6.2 million people between 1998 and 2017. Wildfires result in air pollution due to smoke and gas emissions and can cause serious damage to health, including respiratory problems, affecting vascular health, and even causing death. Wildfires can disrupt transportation, communications, power and gas services, and water supplies. They also lead to a deterioration of the air quality and loss of property, crops, resources, animals, and life. The National Centers for Environmental Information recorded a total of 58 258 fires in the United States in 2020. In Europe, according to the Annual Fire Reports made by the European Forest Fire Information System, there were approximately 65 627 fires in 2019. Satellite monitoring conducted by the Brazilian National Institute of Spatial Research (INPE) recorded a total of 223 000 fire points in Brazil in 2020 [2]. If we consider fire statistics on a global scale, these figures are significantly elevated. Today, although, monitoring forest fires worldwide by satellites is useful for large areas and specific geographical regions, there are no other monitoring methods. The Brazilian Institute of Environment and Renewable Natural Resources explains that the large occurrence of wildfires is expected to continue in Brazil [3].

Satellites in polar orbits can produce a set of images on a daily basis, whereas a geostationary satellite produces images on an hourly basis. Therefore, the monitoring of some areas in a country is not always in real time [3]. Currently, some forest fires cannot be detected by these means due to various natural factors in the environment, such as the following provided by the INPE:

1) fire in areas with less than 30 m^2;
2) fire on the dense forest floor, without affecting the treetops;
3) clouds covering a region;
4) fast forest fires occurring between the satellite images captured;
5) fire behind a mountain or outside the satellite field of view;
6) lack of precision in fire localizations.

Faced with these current and relevant problems, we have developed a new solution to detect forest wildfires to overcome the difficulties of the satellite monitoring systems. In this article, we propose a new method based on deep learning techniques that will automatically detect a forest fire and smoke immediately after the start of the wildfire using edge devices, such as closed-circuit television (CCTV) surveillance cameras, and aerial photographs from unmanned aerial vehicles (UAVs).

There are many research articles for fire and smoke detection in a forest environment and urban areas, for example, studies that involve image/video processing and analysis techniques [4]–[7] and others that use artificial intelligence [8]–[12]. Shi *et al.* [4] used the local binary patterns combined with a convolutional neural network (CNN) to detect fires. Preprocessing the image before classification with a CNN is a common practice. Generally, this image processing feature provides significant gains in accuracy in fire detection. In addition, this preprocessing can apply high-pass or low-pass filtering [5], as well as the removal of fog by means of a histogram equalization in the image [13]. However, these filters usually only perform well in specific and isolated situations. Another common strategy has been feature extraction based on color information present in the image [7]. However, this latter strategy is a very sensitive to variations that affect these attributes, such as the intensity of environmental lighting among others.

Furthermore, nowadays, we can see the emergence of studies that involve wildfire detection through smart devices, following the trend of Internet of Things (IoT) [14]–[16], and edge computing devices [17]–[20]. The use of UAV swarms empowered with edge–cloud computing has begun to be considered for forest fire detection [21]–[23], and some studies are even implementing strategies to monitor wildfires remotely [24]–[27].

The challenge of this work, and the work of others, is to detect a wildfire without the use of thermal cameras [28]. The difficulty of the detection is related to the appearance of the image. Wildfire detection using basic digital image processing techniques results in difficulties due to intrinsic image factors. There are approaches based on color feature statistics [7] and others that merge image processing with CNNs [29]. The challenge in computer vision inherently motivates the use of deep learning [30], [31], because the filters in deep learning models are automatically adjusted during the network training and produce feature maps automatically. Consequently, the use of deep learning is very effective for this study. Other works proved that it is possible to detect fire and smoke (hazy images in [13]) with conventional cameras [32]. This strategy here using conventional cameras is limited to operations during the day. However, the proposed system has advantages over existing solutions, as it does not need expensive cameras. However, as noted, all operations are limited to daytime. Some authors have developed algorithms to generate images of smoke that are very realistic [33]. We used a synthetic Hazy image dataset that can be found in the literature. The use of synthetic images facilitates the construction of a database with a large number of images that are similar to real situations. In this article, we searched for articles that used CNNs [29], [34] in order to compare and validate our EdgeFireSmoke method. Two methods were chosen: 1) Energy-Efficient [35] and 2) Edge-Intelligence [36].

The proposed fire and smoke identification and detection model, named EdgeFireSmoke method, was developed using a customized architecture of a CNN. The proposed EdgeFireSmoke method was trained and optimized to recognize fire and smoke patterns in outdoor environments. Thus, the proposed approach was implemented to complement the satellite information and contribute to the real-time detection of fire points in regions being monitored.

The main contributions of this article are as follows.

1) The fire and smoke detection methods evaluated in this article used the same datasets. However, the methods presented in [35] and [36] have a high computational cost problem. Therefore, to overcome this problem, we proposed a novel lightweight CNN, called EdgeFireSmoke.
2) The EdgeFireSmoke method reduces the detection time, as well as the GPU memory consumption. It is superior to current methods in use and has no significant loss of performance when compared with [35] and [36].
3) It is a novel conceptual model of wildfire detection that can be linked to CCTV systems and outdoor inspection of forests using UAVs. It is an easy-to-train and multiapplication model.
4) The EdgeFireSmoke method presents greater accuracy and shorter time of detection when compared with [35] and [36]. Consequently, our proposed method is promising, since it also needs less memory and requires less energy.
5) Finally, for the experimental procedures, we also proposed a novel dataset by collecting images from UAV videos using real aerial images of forest fires. This new dataset features a variety of 93 real and different environments and complements the existing data in the literature.

II. Methodology

In this section, the proposed method for wildfire detection is discussed, as well as the compared research methods. Finally, the metrics used in the statistical evaluation are described.

A. Datasets

Two datasets were used in this article. The first is obtained from the existing resources (public dataset; see [35]), and the second was created by the authors. A brief description is given as follows.

1) Dataset 1— CCTV Images of Smoke: This dataset was obtained from the work of Khan *et al.* [35] and consists of images recorded by CCTV cameras and organized into the following four classes:

1) smoke-free (ISF);
2) smoke (ICF);
3) smoke-free with fog (ISFN);
4) smoke with fog (ICFN).

From the classes ISF and ICF, Khan *et al.* [35] produced two additional classes synthetically, which include fog, and are

TABLE I
DESCRIPTION OF DATASET 1 WITH THE CLASSES: ISF, ICF, ISFN, AND ICFN

Groups	Images	Percent. (%)	ISF	ICF	ISFN	ICFN
Training	14402	20	3495	3706	3495	3706
Validation	21604	30	5242	5560	5242	5560
Test	36006	50	8737	9266	8737	9266
Total	**72012**	**100**	**17474**	**18532**	**17474**	**18532**

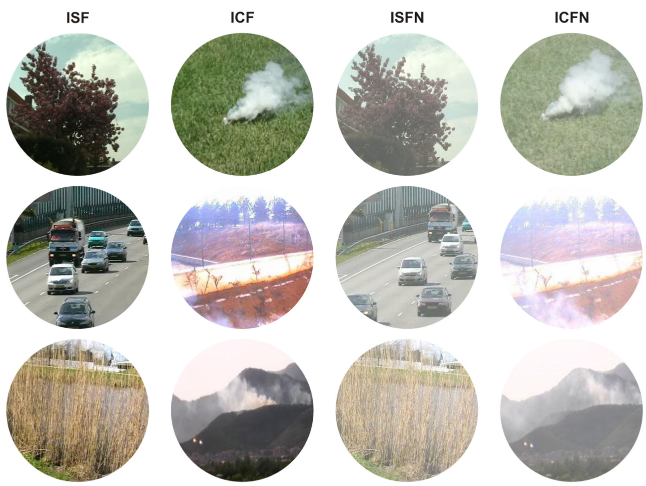


Fig. 1. Representative CCTV images of smoke in Dataset 1. They also contained synthetic Hazy images. Labels: ISF: smoke-free; ICF: smoke; ISFN: smoke-free with fog; and ICFN: smoke with fog.

labeled as ISFN and ICFN. In total, this dataset has 72 012 images, extracted from seven videos. The images have a resolution of 352 × 288 pixels and are in a .jpeg format. To evaluate our proposed method, we applied the same configurations as used by Khan *et al.* [35]: 20% of the images for CNN training, 30% for validation, and 50% for testing.

The images are organized in tenfold for the test set, randomly selected. Using the data distribution shown in Table I, our model is trained with 14 402 images. The images are divided into the four classes: 3495 of ISF, 3706 of ICF, 3495 of ISFN, and 3706 of ICFN.

Fig. 1 gives some examples of images from Dataset 1. This dataset has images of urban and forest areas.

2) Dataset 2— UAV Images of Wildfires: A new dataset (Dataset 2) to complete our conceptual model of wildfire and smoke detection from UAV images was built. In this dataset, public videos for fire detection from the UAVs were selected and processed to build the novel dataset. The videos recorded fire, smoke, burned areas, and green areas, mainly in forested areas, as well as in some urban areas. In total, 93 videos in the .mp4 format in an HD resolution of 1280 × 720 pixels were recorded.

The saved videos were converted to .jpeg images, and some frames were stored and then divided into the following four classes.

1) *Burned-area (IAQ):* The IAQ class has 9348 images after a fire. In general, the floor is black and the trees are without leaves.

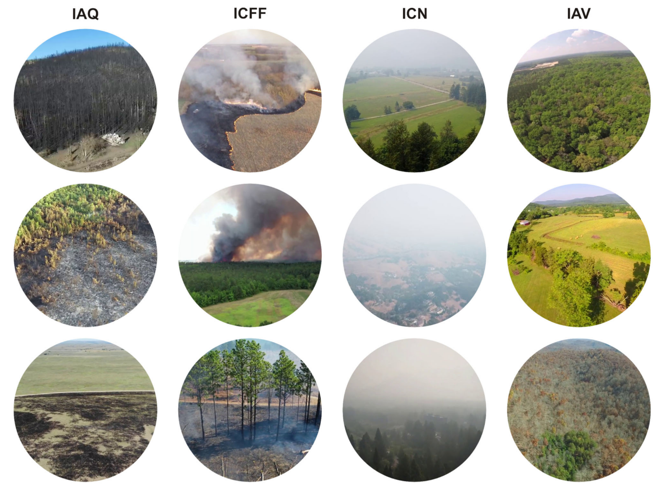


Fig. 2. Representative UAV images of wildfires in Dataset 2. It contains real wildfire images and real Hazy images. Labels: IAQ: Burned-area, ICFF: Fire-smoke, ICN: Fog-area; and IAV: Green-area.

TABLE II
DESCRIPTION OF DATASET 2 WITH THE CLASSES: IAQ, ICFF, ICN, AND IAV

Groups	Images	Percent. (%)	IAQ	ICFF	ICN	IAV
Training	9892	20	1870	3116	1953	2953
Validation	14833	30	2804	4673	2928	4428
Test	24727	50	4674	7790	4881	7382
Total	**49452**	**100**	**9348**	**15579**	**9762**	**14763**

2) *Fire-smoke (ICFF):* The ICFF class has 15 579 images captured when a fire started, usually showing fire, smoke, and the location, such as forested areas and urban areas, with dry vegetation or pasture. The smoke can be white or dark.
3) *Fog-area (ICN):* The ICN class has 9762 images of a foggy environment.
4) *Green-area (IAV):* Finally, we have 14 763 images in the IAV class. This last class has images of the forest, dry vegetation, and pasture areas without any fire, smoke, or fog.

Fig. 2 shows examples of images from Dataset 1. Table II gives a description of the data. Combined, this dataset has 49 452 samples and more than 93 different outdoor environments from 93 videos.

B. Proposed CNN Architecture for Wildfire and Smoke Detection

The proposed CNN was built to be used with edge devices. Initially, the idea was that the EdgeFireSmoke method would be computationally optimized to be embedded in a basic GPUs with a minimum of 1 GB of RAM. We optimized the method by removing excess convolutional layers, based on the CNN proposed in [35]. Based on computational experiments, we have seen that two convolutional layers are the minimum required for this four-class problem. We reduced the number of filters from 64 to 32 in each convolutional layer. Furthermore, a significant reduction in memory consumption was achieved by removing

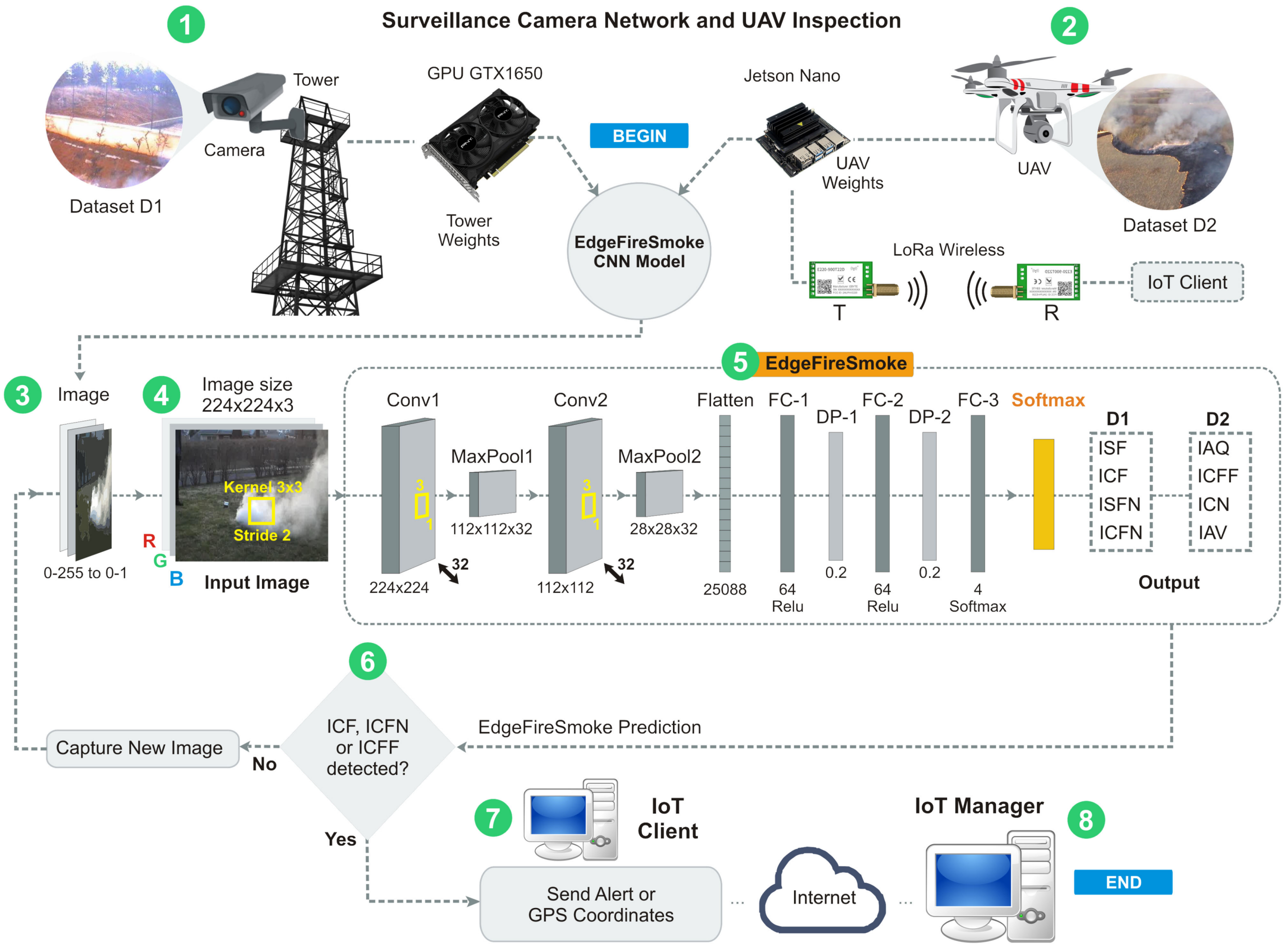


Fig. 3. Proposed conceptual model illustrated in eight steps. (1) and (2) describe two types of image acquisitions: CCTV in an observation tower and UAV monitoring; in each type, we trained the model from two datasets and save the weights separately; (3) and (4) show the preprocessing steps, normalization between 0 and 1, and then resizing the input image to 224 $\times$ 224 pixels; (5) is the proposed CNN architecture to detect wildfires with two convolutional layers, with 32 filters in each layer, two max pooling layers, a flattening layer, and finally a dense neural network with fully connected layers; The dense neural network has two hidden layers (FC-1 and FC-2) and two dropout layers (DP-1 and DP-20) to reduce overfitting. The output layer has four labels, and the model was evaluated in two datasets; (6) is a decision block for wildfire detection cases; in (7), the system has an IoT Client to send wildfire alerts; and (8) is an IoT Cloud Manager with a supervisory system.

excess convolutional layers and excess filters and using a stride equal to 4 in the Max Pooling 2 layer. The proposed method has the smallest number of parameters shown in Table IX. The major advantage of our methodology is that it can be used with a UAV and a Jetson Nano, and a GPU with a minimum of 1 GB of RAM. In addition, it can be linked up to a CCTV system to detect fires and send an alert via the Internet.

Thus, the proposed CNN was evaluated in two different situations: 1) the detection task from a forest observation tower or urban area with a CCTV system using an experimental simulation from Dataset 1 and 2) the detection task with aerial images of UAVs to classify wildfire points in forests simulated from Dataset 2.

Moreover, the two other current methods [35] and [36] were considered in Scenarios 1 and 2, under the same conditions and datasets. The edge configurations of the system were as follows:

1) a desktop with GPU, model GTX1650 with 896 Cuda cores;
2) a Jetson Nano with 128 Cuda cores;
3) additionally, the methods were evaluated in a Tesla T4 with 2560 Cuda cores.

The Edge configuration 1 used Tensorflow 1.14, CUDA 10.0, and CuDNN 7.5. For the Edge configuration 2, a JetPack 4.5, Tensorflow 1.15, CUDA 10.2, and CuDNN 8.2 were used. Finally, for the Edge configuration 3, CUDA 11.2, CuDNN 8.2, and TensorFlow-GPU 2.5 were tested.

Fig. 3 shows the eight steps for the proposed model as an effective tool of prevention and wildfire alert. In Steps 1 and 2, different image acquisition devices can be used; either CCTV systems or aerial images are captured by UAVs. Still, in Step 1, the concept of automatic wildfire surveillance and monitoring network by the CCTV system is defined. In this structure, the cameras can be installed in a surveillance tower. A tower needs a computer to capture images in real time. The EdgeFireSmoke can be run in this computer if it has a GPU. In this first case, this computer with a GPU will use the EdgeFireSmoke model embedded. However, the GPU does not necessarily need to be in the tower. In this second case, the computer without a GPU can send the images to be classified to an IoT manager in a cloud computer, for example, using Google Colaboratory with a Tesla T4. In this experiment, Dataset 1 has images of smoke captured by CCTV cameras and is used to train the CNN. To validate the proposed model, the weights for the CCTV system, called Tower Weights, are selected after the training.

In Step 2, the model of detection that uses UAV images is also evaluated. The proposal is to embed the EdgeFireSmoke method in a Jetson Nano and integrate it with the UAV system. In this way, the model can detect wildfire patterns in real time and send the exact location of the incident to a management system. The goal is that the UAV follows an autonomous scheduled route and sends an alert via radio of the possible occurrence of the wildfire reference point. In the conceptual model, the UAV adopts a communication module of radio frequency to send the alert. The radio receptor would be in a mobile terrestrial base that has a connection to the Internet. Dataset 2 is used for this experiment. To validate our proposed EdgeFireSmoke model, the weights for the UAVs, called the UAV Weights, are selected, after the training.

Steps 3 and 4 describe the preprocessing applied to the input image. Step 3 consists of image normalization, and Step 4 consists of the resizing of the RGB images to the new size of 224×224 pixels.

Step 5 describes the architecture of the proposed EdgeFireSmoke method. The proposed CNN network has two convolutional layers with 32 filters in each layer. The first receives an input image from Step 3. The Max Pooling 1 step reduces the size of activation maps to $112 \times 112 \times 32$ and, consequently, reduces the memory consumption. The second receives these reduced activation maps and carries out convolutional operations using 32 filters. Max Pooling layer 2 is responsible to reduce the activation maps by four, which will then have $28 \times 28 \times 32$ pixels. Finally, the network flattens in the activation maps, which will have a size of 25 088 pixels. The data of the flattened layer are the input data of a dense neural network. This dense neural network has an input layer, two hidden layers fully connected with 64 neurons (FC-1 and FC-2), two Dropout layers of 20% to reduce overfitting, and it has an output layer with four neurons. The Softmax function of activation is used. The Adam optimizer was used during the training of this network and its leaning rate was 0.01. The output of the trained CNN will depend on the weights loaded from Dataset 1 or Dataset 2. The model was implemented using Tensorflow-GPU and Python 3.6.

To arrive at the proposed CNN architecture, we performed several computational experiments starting from the configuration used by Khan *et al.* [35]. In these experiments, we gradually reduced the number of convolutional layers and filters and modified the number of neurons in the two hidden layers using multiples of 2, until reaching the minimum possible value without any significant loss of performance in detecting forest fires. This optimization was designed to build a CNN that needs less RAM compared to the methods in the literature.

In Step 6, the embedded CNN can send a wildfire alert. After detecting the classes ICF, ICFN, or ICFF, the edge devices will send an alert through the IoT Client (Step 7). This device has Internet communication with the IoT Manager. Finally, in Step 8, the IoT Manager Device has a remote computer with a system that monitors wildfire occurrences.

The source code of the proposed method and the database is available in the repository.[1]

C. Energy-Efficient and Edge-Intelligence Methods

Khan *et al.* [35] proposed the Energy-Efficient method that is based on the VGG16 model, trained from ImageNet. The size of the input image is fixed at $224 \times 224 \times 3$ pixels. Each image is passed through five different convolution layers. The first convolution comprises two convolutional layers with an input size of 224×224 using 64 kernels of size 3×3 with stride 1. The result is then forward propagated to the max pooling layer with 22 kernels and stride 2 to get the maximum activations from the feature maps. The second convolution consists of two convolution layers with an input size of 112×112 followed by a max pooling. The third convolution consists of three convolutional layers with 256 kernels of size 3×3 with stride 1 and an input size 56×56 followed by a max pooling. The next convolution has an input size of 28×28. A stack of these convolutional layers is followed by three fully connected layers. The first and second fully connected layers have 4094 channels, while the third fully connected layer is modified from 1000 to 4 channels. The authors used transfer learning and fine-tuning techniques to train this model. The last layer is customized to the four classes of Dataset 1.

In [36], the Edge-Intelligence method is based on MobileNetV2 and trained in ImageNet. The basic building block of the standard MobileNet V2 architecture is a bottleneck with residuals. In a residual bottleneck, the start and the end of a convolutional block are connected to each other with a skip connection. Using these states, the model has the opportunity of retrieving previous activations that are not updated in the convolutional block. The architecture of the MobileNetV2 initiates a convolutional layer followed by 19 residual bottlenecks. After the bottlenecks, there is a convolutional and pooling layer. These layers are followed further by another convolutional layer. The authors considered the same techniques to train this method. The last layer is customized to four classes also in Dataset 1.

In summary, we are evaluating two state-of-the-art methods that have been proposed for smoke detection. These methods have a high computational cost compared to the method proposed in this work. This discussion will follow later.

D. Training Algorithm

To evaluate our method, as well as the other two methods, Algorithm 1 in Table III was used. The abbreviations adopted in this algorithm are described in Table IV.

Algorithm 1 was used to train and test the CNN models using Datasets 1 and 2. In experiment A, only the CCTV images of Dataset 1 were evaluated. In experiment B, only the UAV images (Dataset 2) were used.

[1][Online]. Available: https://github.com/jefferson2021ufc/EdgeFireSmoke.git

TABLE III
TRAINING AND TESTING ALGORITHM

Algorithm 1: Training and testing of the proposed approach

Input Data: Dataset D1 or D2
Initialization: split = [0.2, 0.3, 0.5],
training parameters = [epochs=30, learn rate=0.001, batch=16, optimizer = Adam]
1. $[\alpha, \beta, \gamma]$ = data preparation (Input$_{D}ata, split$)
$(Data_{train}, Data_{evaluation}, Data_{test}$ = (split[0], split[1], split[2])
α = random(D, $Data_{train}$)
β = random(D, $Data_{evaluation}$)
γ = random(D, $Data_{test}$)
return $[\alpha, \beta, \gamma]$
2. $[\Omega]$ = $W(\alpha$, β, training parameters)
3. split$_{test}$ = [0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1]
4. $[\delta, \epsilon, \zeta, \eta, \theta, \tau, \phi, \nu, \rho, \psi]$ = data test preparation(γ, split$_{test}$)
δ = random(γ, batch1)
ϵ = random(γ, batch2)
ζ = random(γ, batch3)
η = random(γ, batch4)
θ = random(γ, batch5)
τ = random(γ, batch6)
ϕ = random(γ, batch7)
ν = random(γ, batch8)
ρ = random(γ, batch9)
ψ = random(γ, batch10)
5. [A, P, R, F, M, T] = $\Omega(\delta, \epsilon, \zeta, \eta, \theta, \tau, \phi, \nu, \rho, \psi)$
6. Output: [A, P, R, F, M, T]

TABLE IV
DESCRIPTION OF THE ACRONYMS ADOPTED IN THE PSEUDOCODE OF THE PROPOSED METHOD WITH ALGORITHM 1

α	Training data	A	Accuracy
β	Validation data	P	Precision
γ	Test data	R	Recall
Ω	Trained model	F	F1-score
D1	Dataset of Smoke from CCTV images	M	Confusion matrix
D2	Dataset of Wildfire from UAV images	T	Mean prediction time

In experiments A and B, the images are randomly split: 20% for training, 30% for validation, and 50% for testing. Subsequently, the test data were divided into ten batches and used in the test step with ten iterations. The input of Algorithm 1 was the images of individual datasets, D1 or D2, and the output was the numerical results obtained from the evaluation metrics.

To train the proposed EdgeFireSmoke method, ten epochs and a learning rate of 0.001 were adopted in all datasets, because this model is a compact CNN and requires fewer epochs to solve the problem.

The other two methods demanded different strategies for training. In this case, two techniques were used: transfer learning and fine-tuning. For the Energy-Efficient method, 100 epochs using transfer learning and two epochs using fine-tuning were used. However, for the Edge-Intelligence method, eight epochs using transfer learning and two epochs using fine-tuning with a learning rate of 0.0001 were used.

E. Evaluation Metrics

In this article, five statistical metrics were evaluated from the Sci-Kit Learn library: accuracy, recall, precision, F-score, and Hamming loss.

Accuracy is the number of hits made among all predicted values, given by

$$\text{Acc}\ (\%) = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FN} + \text{FP} + \text{TN}} \times 100. \quad (1)$$

Recall is the ratio of the correct values among the values that were correctly identified, given by

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \quad (2)$$

Precision is the ratio of correct values among values defined as

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \quad (3)$$

$F1$-score is a measure of balance between precision and recall among unbalanced classes, given by

$$F1\text{−score}\ (\%) = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \times 100. \quad (4)$$

The Hamming loss computes the average Hamming loss or Hamming distance between two sets of samples. If $\hat{y}_j$ is the predicted value for the jth label of a given sample, y_j is the corresponding true value, and n_{labels} is the number of classes or labels, then the Hamming loss L_{Hamming} between two samples is defined as

$$L_{\text{Hamming}}(y, \hat{y}) = \frac{1}{n_{\text{labels}}} \sum_{j=0}^{n_{\text{labels}}-1} 1(\hat{y}_j \neq y_j). \quad (5)$$

F. Real Application

The proposed approach can be applied in the real world using images captured by drones or video surveillance cameras. The drone would make a programmed route, and the surveillance tower is static and of known location. It is also possible to monitor satellite images with the proposed method, only requiring small adjustments in the preprocessing stages. The drone and tower applications are aimed at detecting small- or large-scale forest fires. The satellite image application is only intended to detect large-scale forest fires. The proposed model foresees a network of devices interconnected to a real-time web supervision system. Communication between the customer's device and the supervision system will be done through long-range radio frequency modules and the Internet. The static or mobile video monitoring device must have a battery to detect the images and send information, and it must have a solar panel, battery charging circuit, and a dc–dc converter to power the Jetson Nano board, camera, and radio modules.

III. EXPERIMENTAL RESULTS AND DISCUSSION

In this section, the results obtained using the proposed method and other two approaches are presented. The methods were evaluated in the following three computational platforms using TensorFlow library and Python language:

1) Platform 1: Google Colaboratory using GPU Tesla T4 and 16 GB of RAM;
2) Platform 2: A desktop with GPU GTX 1650 and 4 GB of RAM;

TABLE V
RESULTS OF TEST PERFORMANCE WITH TEN ITERATIONS ON GPU GTX-1650

Dataset	Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	Hamming Loss (%)
CCTV Smoke images	Energy-Efficient	94.92+/-0.45	95.06+/-0.39	94.91+/-0.42	94.93+/-0.41	05.08+/-0.42
	Edge-Intelligence	88.85+/-0.47	90.29+/-0.40	88.52+/-0.49	88.63+/-0.49	11.15+/-0.47
	EdgeFireSmoke	**98.97+/-0.15**	**99.00+/-0.14**	**98.96+/-0.15**	**98.97+/-0.15**	**01.03+/-0.15**
UAV Wildfire images	Energy-Efficient	80.27+/-0.87	80.22+/-0.88	79.42+/-0.88	80.13+/-0.87	19.72+/-0.87
	Edge-Intelligence	65.27+/-0.31	73.32+/-0.0065	62.30+/-0.87	63.55+/-0.93	34.72+/-0.93
	EdgeFireSmoke	**95.77+/-0.31**	**96.25+/-0.35**	**95.84+/-0.35**	**95.76+/-0.31**	**03.96+/-0.37**

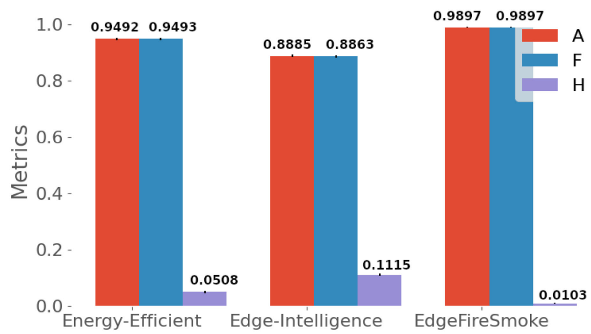


Fig. 4. Accuracy (A), $F1$-score (F), and Hamming loss (H) recorded from CCTV images in Dataset 1. Each color bar represents a CNN method evaluated in this article and its respective results.

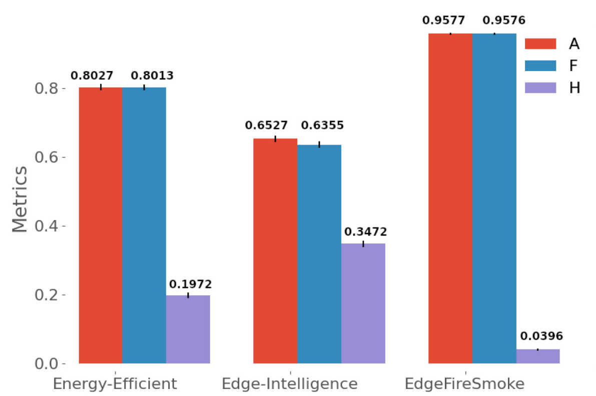


Fig. 5. Accuracy (A), $F1$-score (F), and Hamming loss (H) registered from UAV images in Dataset 2. Each color bar represents a CNN method evaluated in this article and its respective results.

TABLE VI
RESULTS OF TEST TIME (IN SECONDS) REGISTERED FROM TESLA T4 GPU

Model	Mean	Median	Maximum	Minimum
Energy-Efficient	0.04+/-0.01	0.04	0.05	0.03
Edge-Intelligence	0.03+/-0.02	0.03	0.03	0.03
EdgeFireSmoke	0.03+/-0.01	0.03	0.05	0.02

TABLE VII
RESULTS OF TEST TIME (IN SECONDS) REGISTERED FROM GTX-1650 GPU

Model	Mean	Median	Maximum	Minimum
Energy-Efficient	0.10+/-0.01	0.10	0.15	0.08
Edge-Intelligence	1.00+/-0.03	1.01	1.08	0.94
EdgeFireSmoke	0.06+/-0.01	0.06	0.09	0.05

TABLE VIII
RESULTS OF TEST TIME (IN SECONDS) REGISTERED FROM JETSON NANO

Model	Mean	Median	Maximum	Minimum
Energy-Efficient	no supported	-	-	-
Edge-Intelligence	3.53+/-0.03	3.53	4.22	3.32
EdgeFireSmoke	0.21+/-0.01	0.21	0.26	0.18

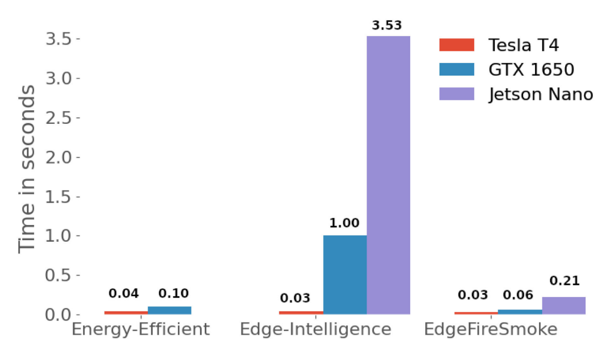


Fig. 6. Time of classification obtained from each CNN using three different Edges-GPU.

3) Platform 3: A Jetson Nano with 4 GB of RAM.

Table V shows the performance of the proposed method, which achieves satisfactory results in Dataset 1. The proposed method achieves an accuracy of 98.71% and the $F1$-score of 98.97%. It records the smallest loss of 1.03% compared with other two methods. When compared with the second-best method, the difference between accuracy was 4.05%. The proposed method presents the smallest standard deviation. In Dataset 2, it achieved an accuracy of 95.77%, and 96.25% for precision, 95.84% for recall, and $F1$-score of 95.76%, as shown in Figs. 4 and 5.

The classification time was registered with all the GPU models evaluated in this article. The proposed method had the shortest classification time, as shown in Tables VI–VIII, as well as in Fig. 6. However, it was impossible to evaluate the Energy-Efficient method using the Jetson Nano because this method needed more RAM.

Comparing the results shown in Table IX, in terms of RAM consumption on the GPU, the proposed method had some advantages, such as: 1) it only needed 850 MB of RAM, while the others evaluated methods need more RAM, 3798 MB and 2348 MB, respectively and 2) the lowest consumption by the proposed CNN can be justified due to fewer convolution layers and, in general, fewer parameters. Table IX presents the lowest energy consumption during the training phase and the second

TABLE IX
ADVANTAGES OF EACH CNN METHOD

Model	Params	GPU Memory (MB)	Mean Power Train (W)	Mean Power Test (W)	Test Time (ms)	F1-Score (%)
Energy-Efficient	33,613,636	3795	38.9	15.6	101.727	94.93
Edge-Intelligence	3,542,988	2348	44.6	**13.5**	1002.544	88.63
EdgeFireSmoke	**1,620,260**	**850**	**33.0**	14.5	**65.073**	**98.97**

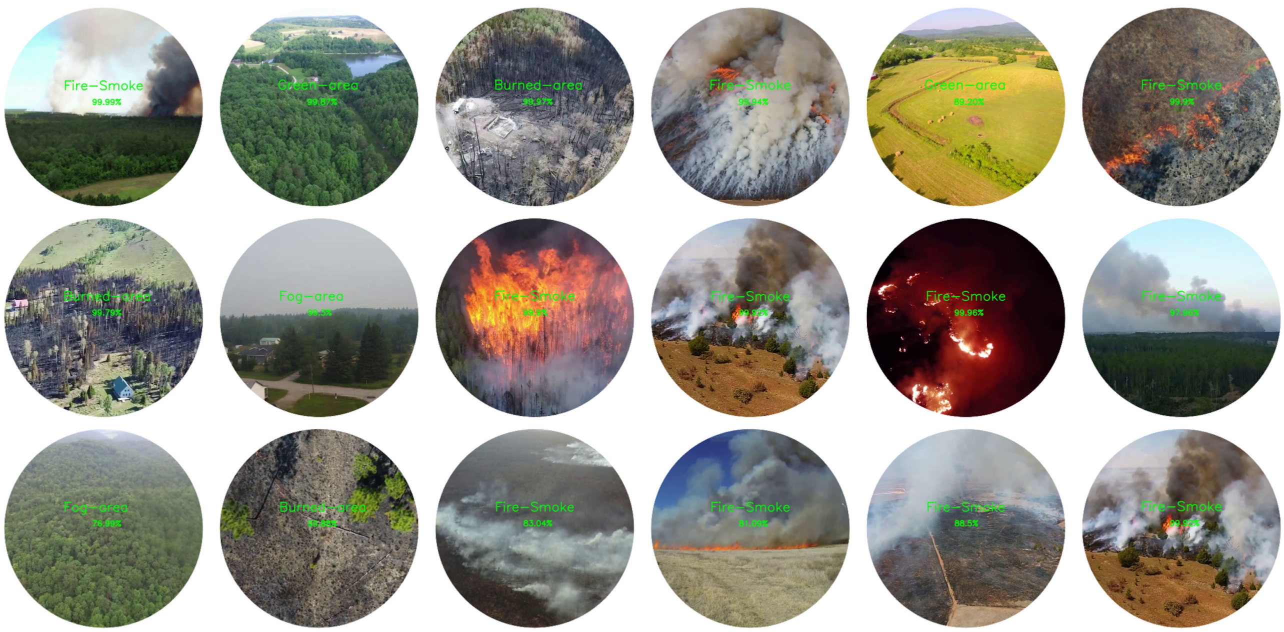


Fig. 7. Video results after the detection by EdgeFireSmoke using Dataset 2. The classification result by the proposed CNN is indicated in the middle of the image in green. The percentage value represents the detection certainty level and its respective class.

lowest consumption during the testing phase, when it achieved 33 and 14.5 W, respectively.

Table IX illustrates the following main advantages of the proposed CNN.

1) a fewer number of parameters;
2) the lowest consumption of RAM;
3) the second lowest consumption of electrical energy during the classification test;
4) the lowest time to classify a sample;
5) the highest accuracy.

Furthermore, the existing forest fire detection models can be run on our custom image dataset. However, such models must be set up for use with RGB images. As limitations, the proposed method is limited to work with natural or artificial light illuminating the environment.

Fig. 7 offers the results of wildfire detection using the proposed method. These images are examples of detection in real time from the Dataset 3 videos. The classification results with the class label predicted by the proposed EdgeFireSmoke method are shown in the center of the images. The confidence level of the detection is given below the label. The confidence level can be adjusted to the desired value to reduce false detections. In this study, the classes of interest are fire and smoke detection: ICF, ICFN, and ICFF. In general, the proposed EdgeFireSmoke method detected the classes of the occurrence of wildfires: ICF, ICFN, and ICFF with high accuracy.

In addition, it is possible to say that the best model is the one that can detect the occurrence of wildfires more accurately, i.e., the proposed method. However, it is necessary to consider the computational costs of all models.

When the goal is a model that can be used with an UAV, the factors of electrical energy consumption and time of classification are relevant. These factors can affect the UAV flying time. While using desktop applications, the time factor and consumption of electrical energy can be used with a lower priority, because we can use powerful GPUs and adopt the best model independent of the classification time.

Another relevant aspect is the form of communication established between the UAV and a monitoring center. In an imaginary situation, we can think of a network of UAVs that can exchange information and geolocate locations with fire points in real time. It will be interesting in future works to evaluate, from an energy point of view, the advantages and disadvantages of using edge or cloud processing.

Analyzing the confusion matrix shown in Table X, the true and predicted classes are defined. In general, the proposed method classified the ISF and ISFN samples more accurately. The values are highlighted in bold. The Edge-Intelligence method showed a better ability to classify ICF and ICFN on Dataset 1.

TABLE X
CONFUSION MATRIX ASSOCIATED WITH ALL CLASSIFIERS AFTER EVALUATION IN DATASET 1

True class	Classified as	Energy-Efficient	Edge-Intelligence	EdgeFireSmoke
ISF	ISF	842.0	646.5	**861.1**
	ICF	30.20	227.0	12.40
	ISFN	1.300	0.000	0.000
	ICFN	0.000	0.000	0.000
ICF	ISF	25.10	0.200	1.200
	ICF	878.9	**924.1**	918.7
	ISFN	0.900	0.000	0.000
	ICFN	21.60	2.200	6.600
ISFN	ISF	8.000	58.70	0.000
	ICF	0.300	4.700	91.80
	ISFN	807.5	712.4	**861.6**
	ICFN	57.70	97.70	11.30
ICFN	ISF	0.000	0.000	0.000
	ICF	25.90	4.300	0.700
	ISFN	11.70	0.100	4.100
	ICFN	888.9	**922.1**	921.7

TABLE XI
CONFUSION MATRIX ASSOCIATED WITH ALL CLASSIFIERS AFTER EVALUATION IN DATASET 2

True class	Classified as	Energy-Efficient	Edge-Intelligence	EdgeFireSmoke
IAQ	IAQ	315.3	377.1	**433.9**
	ICFF	39.3	56.3	16.2
	ICN	21.1	0.2	1.2
	IAV	91.3	33.4	15.7
ICFF	IAQ	32.1	107.3	6.9
	ICFF	646.3	604.4	**738.7**
	ICN	57.4	6.8	10.1
	IAV	43.2	60.5	23.3
ICN	IAQ	1.5	317.6	0.5
	ICFF	28.2	3.6	6.6
	ICN	422.8	77.3	**478.0**
	IAV	35.5	89.5	2.9
IAV	IAQ	38.6	138.3	3.6
	ICFF	58.2	41.2	9.2
	ICN	39.7	0.3	2.3
	IAV	601.5	558.2	**722.9**

In the confusion matrix of Table XI, the proposed method shows a better ability to classify all classes. The values are highlighted in bold. The EdgeFireSmoke method detects the occurrence of wildfires, which is the most important class, more accurately than the other models.

The idea of building a CNN with a basic and functional architecture came precisely from the idea that there is no need for parameterization in classical image processing techniques. Therefore, the use, in part, of other techniques used in the literature would be of little benefit. A model based on deep learning has, as its main advantage, the automatic adjustment of filters internally in the convolutional layers of the network. These filters are self-adjusting during network training. Therefore, the generalizability of the model will mainly depend on the diversity of the images and situations presented and on the correct training of the proposed model.

Finally, we show that this research has a relevant social role for the planet, in the conservation of forests, human health, and the preservation of animal life, and many other advantages of the proposed method that were mentioned highlighting the efficient monitoring of the occurrence of forest fires. However, like every method, it has some disadvantages. As a disadvantage of this research, we highlight the fact that deep learning methods, in this case the CNNs, present better performance of time using a graphic card GPU, and it is slower with CPU. A second disadvantage comes from the use of conventional RGB cameras for fire detection, which limits the use of the system during the daytime with the presence of sunlight, whereas at night, it would need long-range artificial lighting.

IV. CONCLUSION

In this article, we presented a novel CNN model capable of detecting and classifying the occurrence of forest fires seen in aerial photographs from UAV systems and CCTV systems.

The proposed method proved to be effective and efficient in both datasets evaluated. In addition, it presented advantages in relation to computational cost, classification time, and sufficient accuracy for good performance.

The proposed method performed well when detecting forest fires, with 98.97% of accuracy in Dataset 1 and 95.77% of accuracy in Dataset 2. The shortest classification time per sample recorded by the proposed method was 30 ms in Tesla T4, 60 ms in the GTX 1650, and 210 ms in the Jetson Nano.

The main objectives were to prove that it is possible to embed the proposed computational model and to complement the information provided by satellites to detect fires, through the use of images from a video monitoring tower and inspection by UAVs.

As a future work, field tests will be carried out with a UAV and the proposed computational model, in order to evaluate its performance in an outdoor controlled environment.

REFERENCES

[1] BBC News Brazil, “Forest fires around the world are the biggest in scale and CO_2 emissions in 18 years,” 2020. [Online]. Available: https://www.bbc.com/portuguese/geral-54202546

[2] Brazilian National Institute of Spatial Research, “Monitoring of active spots by countries,” 2021. [Online]. Available: https://queimadas.dgi.inpe.br/queimadas/portal-static/estatisticas_paises/

[3] Brazilian Institute of Environment and Renewable Natural Resources, “Burn monitoring in satellite images,” 2017. [Online]. Available: http://www.ibama.gov.br/consultas/incendios-florestais/consultas-monitoramento-de-queimadas/monitoramento-de-focos-de-queimadas-em-imagens-de-satelites

[4] J. Shi, W. Wang, Y. Gao, and N. Yu, “Optimal placement and intelligent smoke detection algorithm for wildfire-monitoring cameras,” *IEEE Access*, vol. 8, pp. 72326–72339, 2020.

[5] X. Ye, F. Yu, C. Zhou, and M. Jiang, “A blurry image recognition method for straw burning detection,” in *Proc. IEEE 5th Int. Conf. Signal Image Process.*, 2020, pp. 323–327.

[6] S. Wang, F. Yu, C. Zhou, and M. Jiang, “Straw burning detection method based on improved frame difference method and deep learning,” in *Proc. IEEE 5th Int. Conf. Image, Vis. Comput.*, 2020, pp. 29–33.

[7] W. Li, S. Xiaobo, C. Jun, and L. Ying, “Research on forest fire image recognition algorithm based on color feature statistics,” in *Proc. 6th Int. Conf. Intell. Comput. Signal Process.*, 2021, pp. 346–349.

[8] X. Li, Z. Chen, Q. J. Wu, and C. Liu, “3D parallel fully convolutional networks for real-time video wildfire smoke detection,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 1, pp. 89–103, Jan. 2020.

[9] H. Tao and X. Lu, “Smoke vehicle detection based on spatiotemporal bag-of-features and professional convolutional neural network,” *IEEE Trans. Circuits Syst. VideoTechnol.*, vol. 30, no. 10, pp. 3301–3316, Oct. 2020.

[10] W. Thomson, N. Bhowmik, and T. P. Breckon, “Efficient and compact convolutional neural network architectures for non-temporal real-time fire detection,” in *Proc. 19th IEEE Int. Conf. Mach. Learn. Appl.*, 2020, pp. 136–141.

[11] X. Shi, N. Lu, and Z. Cui, “Smoke detection based on dark channel and convolutional neural networks,” in *Proc. IEEE 5th Int. Conf. Big Data Inf. Analytics*, 2019, pp. 23–28.

[12] Y. Zhang and Y. Hu, “Video smoke detection based on convolution neural network,” in *Proc. Int. Comput., Signals Syst. Conf.*, 2018, pp. 123–127.

[13] H. Ullah, K. Muhammad, M. Irfan, B. Z. Muhammad, and S. Anwar, "Light-DehazeNet: A novel lightweight CNN architecture for single image dehazing," *IEEE Trans. Image Process.*, vol. 30, pp. 8968–8982, Oct. 2021.

[14] G. Roque and V. S. Padilla, "LPWAN based IoT surveillance system for outdoor fire detection," *IEEE Access*, vol. 8, pp. 114900–114909, 2020.

[15] S. Verma, S. Kaur, D. B. Rawat, C. Xi, L. T. Alex, and N. Zaman Jhanjhi, "Intelligent framework using IoT-based WSNs for wildfire detection," *IEEE Access*, vol. 9, pp. 48 185–48 196, 2021.

[16] K. Kaur, S. Garg, G. Kaddoum, S. H. Ahmed, and M. Atiquzzaman, "KEIDS: Kubernetes-based energy and interference driven scheduler for industrial IoT in edge-cloud ecosystem," *IEEE Internet Things J.*, vol. 7, no. 5, pp. 4228–4237, May 2020.

[17] J. Zhang, C. Xu, Z. Gao, J. J. Rodrigues, and V. H. C. de Albuquerque, "Industrial pervasive edge computing-based intelligence iot for surveillance saliency detection," *IEEE Trans. Ind. Informat.*, vol. 17, no. 7, pp. 5012–5020, Jul. 2020.

[18] Z. Gao, C. Xu, H. Zhang, S. Li, and V. H. C. de Albuquerque, "Trustful Internet of surveillance things based on deeply represented visual co-saliency detection," *IEEE Internet Things J.*, vol. 7, no. 5, pp. 4092–4100, May 2020.

[19] D. Połap and M. Woźniak, "Meta-heuristic as manager in federated learning approaches for image processing purposes," *Appl. Soft Comput.*, vol. 113, 2021, Art. no. 107872.

[20] Z. Gao *et al.*, "Salient object detection in the distributed cloud-edge intelligent network," *IEEE Netw.*, vol. 34, no. 2, pp. 216–224, Mar./Apr. 2020.

[21] Y. Chen *et al.*, "UAV image-based forest fire detection approach using convolutional neural network," in *Proc. 14th IEEE Conf. Ind. Electron. Appl.*, 2019, pp. 2118–2123.

[22] O. M. Bushnaq, A. Chaaban, and T. Y. Al-Naffouri, "The role of UAV-IoT networks in future wildfire detection," *IEEE Internet Things J.*, vol. 8, no. 23, pp. 16984–16999, Dec. 2021.

[23] S. Dutta and S. Ghosh, "Forest fire detection using combined architecture of separable convolution and image processing," in *Proc. 1st Int. Conf. Artif. Intell. Data Analytics*, 2021, pp. 36–41.

[24] M. Y. Arafat and S. Moh, "Bio-inspired approaches for energy-efficient localization and clustering in UAV networks for monitoring wildfires in remote areas," *IEEE Access*, vol. 9, pp. 18649–18669, 2021.

[25] A. Viseras, M. Meissner, and J. Marchal, "Wildfire front monitoring with multiple UAVs using deep Q-learning," *IEEE Access*, to be published, doi: 10.1109/ACCESS.2021.3055651.

[26] K. Shrestha, R. Dubey, A. Singandhupe, S. Louis, and H. La, "Multi objective UAV network deployment for dynamic fire coverage," in *Proc. IEEE Congr. Evol. Comput.*, 2021, pp. 1280–1287.

[27] J. Jeyavel, A. A. Prasad, K. M. Shelke, P. D. Sargade, and U. V. Thoke, "Survey on fire fighting techniques using unmanned aerial vehicles," in *Proc. Int. Conf. Adv. Comput. Innov. Technol. Eng.*, 2021, pp. 239–241.

[28] M. M. Valero *et al.*, "Thermal infrared video stabilization for aerial monitoring of active wildfires," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 2817–2832, Feb. 2021.

[29] P. Ma, F. Yu, C. Zhou, and M. Jiang, "An integrated smoke detection method based on convolutional neural network and image processing," in *Proc. IEEE 8th Int. Conf. Comput. Sci. Netw. Technol.*, 2020, pp. 32–36.

[30] S. Garg, K. Kaur, N. Kumar, G. Kaddoum, A. Y. Zomaya, and R. Ranjan, "A hybrid deep learning-based model for anomaly detection in cloud datacenter networks," *IEEE Trans. Netw. Service Manage.*, vol. 16, no. 3, pp. 924–935, Sep. 2019.

[31] S. Garg, K. Kaur, N. Kumar, and J. J. Rodrigues, "Hybrid deep-learning-based anomaly detection scheme for suspicious flow detection in SDN: A social multimedia perspective," *IEEE Trans. Multimedia*, vol. 21, pp. 566–578, Jan. 2019.

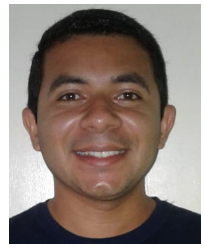
[32] T. Gupta, H. Liu, and B. Bhanu, "Early wildfire smoke detection in videos," in *Proc. 25th Int. Conf. Pattern Recognit.*, 2021, pp. 8523–8530.

[33] C. Xie and H. Tao, "Generating realistic smoke images with controllable smoke components," *IEEE Access*, vol. 8, pp. 201418–201427, 2020.

[34] D. Sheng, J. Deng, and J. Xiang, "Automatic smoke detection based on SLIC-DBSCAN enhanced convolutional neural network," *IEEE Access*, vol. 9, pp. 63933–63942, 2021.

[35] S. Khan, K. Muhammad, S. Mumtaz, S. W. Baik, and V. H. C. de Albuquerque, "Energy-efficient deep CNN for smoke detection in foggy IoT environment," *IEEE Internet Things J.*, vol. 6, no. 6, pp. 9237–9245, Dec. 2019.

[36] K. Muhammad, S. Khan, V. Palade, I. Mehmood, and V. H. C. De Albuquerque, "Edge intelligence-assisted smoke detection in foggy surveillance environments," *IEEE Trans. Ind. Informat.*, vol. 16, no. 2, pp. 1067–1075, Feb. 2020.

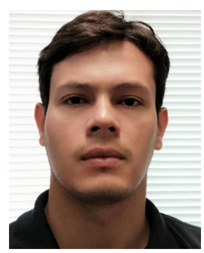


Jefferson Silva Almeida received the bachelor's degree in industrial maintenance and industrial automation and the master's degree in computer science from the Federal Institute of Ceará, Limoeiro do Norte, Brazil, in 2017 and 2019, respectively. He is currently working toward the Ph.D. degree in electrical engineering with the Federal University of Ceará, Fortaleza, Brazil.

His research interests include embedded systems and mobile robotics.

Chenxi Huang received the Ph.D. degree in computer science from the College of Electronics and Information Engineering, Tongji University, Shanghai, China, in 2019.

Since 2019, he has been with the School of Informatics, Xiamen University, Xiamen, China, where he is currently an Assistant Professor. His current research interests include deep learning, object detection, and image analysis.

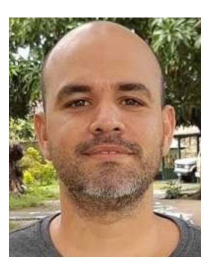


Fabrício Gonzalez Nogueira received the B.Sc. degree in computer engineering and the M.Sc. and Ph.D. degrees in electrical engineering from the Federal University of Pará, Belém, Brazil, in 2007, 2008, and 2012, respectively.

Since 2013, he has been a Professor with the Department of Electrical Engineering, Federal University of Ceará, Fortaleza, Brazil. His main research interests include adaptive and robust control, robust stability, robust linear matrix inequality, identification and control of linear parameter-varying systems, and embedded systems.

Surbhi Bhatia received the Ph.D. degree in computer science and engineering from Banasthali Vidyapith, Jaipur, India, in 2018.

She is currently an Assistant Professor with the Department of Information Systems, College of Computer Sciences and Information Technology, King Faisal University, Al Hofuf, Saudi Arabia. Her research interests include data mining, machine learning, and information retrieval.



Victor Hugo C. de Albuquerque (Senior Member, IEEE) received the B.S.E. degree in mechatronics engineering from the Federal Center of Technological Education of Ceará, Paraná, Brazil, in 2006, the M.Sc. degree in teleinformatics engineering from the Federal University of Ceará (UFC), Fortaleza, Brazil, in 2007, and the Ph.D. degree in mechanical engineering from the Federal University of Pará, Belém, Brazil, in 2010.

He is currently a Professor and Senior Researcher with the Department of Teleinformatics Engineering, UFC. His research interests include image data science, Internet of Things, machine/deep learning, pattern recognition, automation and control, and robotics.