Expert Systems With Applications 221 (2023) 119747

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

EdgeFireSmoke++: A novel lightweight algorithm for real-time forest fire detection and visualization using internet of things-human machine interface

Jefferson S. Almeida [a], Senthil Kumar Jagatheesaperumal [b], Fabrício G. Nogueira [a], Victor Hugo C. de Albuquerque [c,*]

[a] Department of Electrical Engineering, Federal University of Ceará, Fortaleza, 60440554, Ceará, Brazil
[b] Department of Electronics & Communication Engineering, Mepco Schlenk Engineering College, Sivakasi, 626005, Tamil Nadu, India
[c] Department of Teleinformatics Engineering, Federal University of Ceará, Fortaleza, 60455970, Ceará, Brazil

ARTICLE INFO

Keywords:
Forest fire detection
Machine Learning
Deep Learning
Internet of things
Human machine interface

ABSTRACT

Forest fires can have severe impacts on both the environment and human communities. They can cause soil erosion, loss of habitat and biodiversity, as well as the release of carbon dioxide and other pollutants into the atmosphere. In addition, they can cause damage to properties, displacement of residents, and put firefighters and other responders at risk. Forest fires can also contribute to climate change by releasing stored carbon into the atmosphere and altering ecosystems. In this work, we propose a novel algorithm capable of monitoring small areas of forest reserve environment through video streaming in real-time. It will complement the existing means of forest monitoring and surveillance and provide effective solutions faced in satellite-based monitoring. The proposed algorithm is an improvement of the EdgeFireSmoke method and uses an artificial neural network together with a deep learning method. The proposed EdgeFireSmoke++ algorithm was able to detect forest fires with 95.41% accuracy and 95.49% accuracy from the evaluated dataset. The real-time experiment performed very well for use with Internet Protocol cameras, reaching 33 frames per second. This test was superior concerning the evaluated methods in the literature.

1. Introduction

The forests across our planet are the primary sources of natural resources and provide a comfortable life by establishing a perfect balance of nature and its ecosystems (Karjalainen, Sarjala, & Raitio, 2010). As per the World Wild Life (WWF) organization, forests play a crucial role to purify the air we breathe, filtering the water we drink, and playing a significant part against global warming (Life, 2022). However, a potential risk has existed for decades is bushfires and forest fires, which are capable of destroying much of the fauna and flora in an entire region in addition it causes significant impacts on terrestrial dynamics.

The World Health Organization (WHO) reported from worldwide statistical data analyzed that forest fires and volcanic activities affected 6.2 million people worldwide between 1998–2017 (Organization, 2022). We currently know that the impacts are devastating and it goes beyond the human species and they affect other species and nature.

Considering recent statistics, during the period between 2019–2020 there were numerous outbreaks of forest fires in Brazil, the United States, and Europe. According to the Brazilian National Institute of Spatial Research (INPE), a total of 223,000 fire outbreaks were recorded in Brazilian territory in 2020 (INPE, 2021). The National Centers for Environmental Information (NCEI) recorded a total of 58,258 fires in the United States in the same year. In Europe, the numbers reported by the European Forest Fire Information System (EFFIS) showed the occurrence of approximately 65,627 outbreaks in 2019.

The Brazilian Institute of Environment and Renewable Natural Resources (IBAMA), which is one of the main environmental organizations in Brazil, stated that the frequent occurrence of forest fires will continue in Brazil for the next few years (IBAMA, 2017). This statement motivates us to increase the provision of technological tools to prevent, monitor, and fight forest fires, in order to complement existing technologies.

Currently, the monitoring of forests has been done through space satellites spread around the entire terrestrial globe. However, a few

* Corresponding author.
E-mail addresses: jeffersonsilva@lapisco.ifce.edu.br (J.S. Almeida), senthilkumarj@mepcoeng.ac.in (S.K. Jagatheesaperumal), fnogueira@dee.ufc.br (F.G. Nogueira), victor.albuquerque@ieee.org (V.H.C. de Albuquerque).

https://doi.org/10.1016/j.eswa.2023.119747
Received 14 November 2022; Received in revised form 19 February 2023; Accepted 22 February 2023
Available online 25 February 2023

limitations makes real-time monitoring of small areas in the forest reserve to be highly challenging, which are indicated as follows:

- Fire in areas with less than 30 square meters of coverage;
- Occurrence of fire in the dense forest floor, without affecting the treetops;
- Clouds covering a region of fire area;
- Fast forest fires occurring between the satellite images captured;
- Fire behind a mountain or outside the satellite field of view;
- Lack of accuracy and precision in fire localization.

Therefore, we propose in this work an alternative tool that uses artificial intelligence to assist in the intelligent video monitoring of small areas of forest reserve. In this way, it is expected that the proposed algorithm will be able to overcome at least three of the problems listed above: (1) Detection of fire in small areas; (2) Detection of fast forest fires that form blind spots for satellites; (3) Detection of fire outside the field of view of satellites.

In the literature, we observe a significant number of works that involve Machine Learning and Deep Learning to detect forest fires. Some authors defend the use of object detection strategies based on regression, considering these algorithms to be efficient, as they identify the object of interest using bounding boxes. However, they present high computational costs, with high processing dependence on dedicated graphics cards (GPU) (Li, Shen, Li, & Xu, 2022; Tahir, Waqar, Khalid, & Usman, 2022). In other approaches, convolutional neural networks (CNNs) have been used because they present great performance with digital images and less dependence on GPU in relation to the methods mentioned above (Chatragadda et al., 2022; Oliver, Ashwanthika, & Aswitha, 2020; Rachana, Rajalakshmi, Bhat, Kaur, & Bimali, 2022; Silva, Huang, Nogueira, Bhatia, & Albuquerque, 2022).

The use of special or conventional RGB cameras, such as thermal and infrared cameras, has also been investigated, as they increase the possibility of detecting fires in adverse visibility conditions (Li, Xiaobo, Jun, & Ying, 2021; Sadi, Zhang, Xie, & Hossain, 2021; Tlig, Bouchouicha, Sayadi, & Moreau, 2022). The most prominent limitations of thermal and infrared cameras include limited sensing distance to a few tens of meters and high acquisition cost.

A technique that has been extensively investigated using CNNs has been Transfer Learning, which consists of evaluating the use of previously trained CNNs with ImageNet dataset to detect forest fires with images totally different from the original learning patterns (Agarwal & Jha, 2021). Therefore, in this work, we will evaluate this technique with some of the lightest modern CNNs methods found in the literature, such as VGG16 and VGG19 (Simonyan & Zisserman, 2014), MobileNet V2 (Sandler, Howard, Zhu, Zhmoginov, & Chen, 2018), Inception V3 (Szegedy, Vanhoucke, Ioffe, Shlens, & Wojna, 2016), Xception (Chollet, 2017) and DenseNet121 (Huang, Liu, Van Der Maaten, & Weinberger, 2017). We present in Table 1 a summary of popular articles selected from the literature that use CNNs for fire detection. The selection of these methods was considered by including methods presented in Table 1 that do not use bounding boxes.

Based on the expertise of the authors in the prescribed domain, there were works on smoke detection in various environments that were developed using edge intelligent models (Muhammad, Khan et al., 2019) and deep learning models (Khan et al., 2021, 2019). Additionally, energy-efficient fire monitoring solutions were also presented by the authors with intelligent networks (Muhammad, Rodrigues, Kozlov, Piccialli, & de Albuquerque, 2020) and smoke detection in IoT environments (Khan et al., 2019). Further, some noteworthy works on video summarization were developed, which were predominately meant for resource-effective summarization (Muhammad, Hussain, Del Ser, Palade and De Albuquerque, 2019), cost-effective summarization (Muhammad, Hussain, Tanveer, Sannino and de Albuquerque, 2019), and multiview IIoT applications (Hussain, Muhammad, Del Ser, Baik, & de Albuquerque, 2019). Very recently, an automatic fire extinguishing system was developed by the authors (Jagatheesaperumal, Muhammad, Saudagar, & Rodrigues, 2023). Such hardcore competencies of the authors give inspiration for the development of the novel EdgeFireSmoke++ approach proposed in this work.

Currently, emphasis has been given to lightweight algorithms, as they require less energy consumption, which is essential for embedded systems integrated with UAVs, and for processing in the CPU (Fan & Pei, 2021; Silva et al., 2022; Wang et al., 2021). However, most of these methods still have high computational costs. The use of meteorological information to predict the risk of forest fire has also been investigated (Omar, Al-zebari, & Sengur, 2021; Park, Jeon, Bak, Park, et al., 2022; Reddy & Kalyanasundaram, 2022). Therefore, we will address this issue of using climate data in order to make our contribution using the lightest modern CNNs.

The proposed algorithm constitutes an improvement on the work presented by Silva et al. (2022). Unlike the previous work, our algorithm is intended for use by video surveillance cameras with an internet connection, which could be loaded onto UAVs after some modifications.

The main difference presented here was the addition of an Artificial Neural Network (ANN), which precedes the CNN proposed by Silva et al. (2022). Moreover, with the enhancement in data visualization, we present a Human Machine Interface (HMI) that collects meteorological data from the internet to update the real-time risk of forest fire in certain monitored regions.

ANN will perform the pre-processing tasks such as, screening the input images and detecting the absence or presence of a forest environment. While CNN will classify the forest images into four groups: (1) Green area, (2) Burnt area, (3) Forest fire area, and (4) Fog area. Thus, it will make the EdgeFireSmoke algorithm more reliable and robust in its predictions (Silva et al., 2022). The new method we call EdgeFireSmoke++ will have enhanced features and improvements compared to its previous version.

The main contributions of this work were summarized as follows:

1. Inclusion of pre-processing stage based on the strategy proposed originally by Silva et al. (2022) for detection fire in forest environment.
2. Use of meteorological variables for forest fire risk calculation based on the inference in Lourenço, Gonçalves, and Loureiro (1997).
3. Implementation of a Human–Machine Interface (HMI) for visualization of detections and color scale of the forest fire risks.
4. Integration with the IoT using the OpenWeather API for meteorological data collection from a web server;
5. Video processing improvements observed in real-time with HD quality and reasonable frame rate from IP Cameras.
6. Discarding the dependency of dedicated graphics card (GPU) present in the CNN model proposed in Silva et al. (2022).
7. Maintenance on the capability of the model to train the proposed algorithm for new environments.

2. Methodology

In this work, the forest fire detection mechanism is carried out in two levels. The first level (ANN) is responsible for detecting the absence or presence of the forest environment. In the second level (CNN), the algorithm classifies the image in the four situations illustrated in the subsequent sections. The algorithm for wildfire detection will be discussed along with the comparisons of other research methods. Finally, the metrics used in the statistical evaluation are described.

2.1. Acquisition of images

In this work, two types of cameras were used for the acquisition and evaluation of forest fires. The first model is an IP camera with a WiFi connection, HD resolution, and remote access support via RTSP protocol. The second camera model was the webcam connected to a computer with VGA resolution. More specifications of the chosen camera are presented in Table 2.

Table 1
Summary of key observations from the literature on the model characteristics.

CNN model	Forest detection	Forest fire detection	Bounding boxes	Internet communication	Meteorological variables	Human machine interface
EdgeFireSmoke++	Yes	Yes	No	Yes	Yes	Yes
SRN-Yolo (Li et al., 2022)	No	Yes	Yes	No	No	Yes
YoloV5 (Tahir et al., 2022)	No	Yes	Yes	No	No	Yes
EdgeFireSmoke (Silva et al., 2022)	No	Yes	No	Yes	No	No
VGG16 (Khan, Muhammad, Mumtaz, Baik, & de Albuquerque, 2019; Simonyan & Zisserman, 2014)	No	Yes	No	No	No	No
VGG19 (Chatragadda et al., 2022; Simonyan & Zisserman, 2014)	No	Yes	No	No	No	No
MobileNetV2 (Muhammad, Khan, Palade, Mehmood and De Albuquerque, 2019; Sandler et al., 2018)	No	Yes	No	No	No	No
InceptionV3 (Agarwal & Jha, 2021; Szegedy et al., 2016)	No	Yes	No	No	No	No
Xception (Chollet, 2017)	No	No	No	No	No	No
DenseNet121 (Chatragadda et al., 2022; Huang et al., 2017)	No	Yes	No	No	No	No

Table 2
Description of imaging sensors used for acquisition of the forest fires.

Model		Resolution	FOV	Sensor	Connection
Cam1	IP camera	1280 × 720	111°	1/4	WiFi, RTSP
Cam2	PC camera	640 × 480	60°	1/4	USB, Serial

2.2. Climatological factors

Meteorological data are essential for us to be able to calculate the rate of deflagration and propagation of forest fires since they allow conditioning the state of dryness of the fuels (Lourenço et al., 1997). In the study carried out by Lourenço (1991), it was found that temperature and relative humidity are key variables for the outbreak of fires, while the wind is responsible for the rapid propagation. In this study, we defend the idea that the climate information considered through the OpenWeather API is sufficient for this monitoring because in an initial study, it manages to overcome the lack of a network of local meteorological stations in each region. The OpenWeather API is a free platform that provides weather data from all over the planet in real-time. Therefore, the key parameters used in the study are:

- Air temperature, in °C;
- Relative air humidity, in %;
- Wind speed, in km/h.

2.3. ANN architecture for florestal image recognition

The lack of pre-processing stages in the method proposed by Silva et al. (2022) imparts challenges in identifying whether the input image is really a forest environment. Therefore, by including ANN in this work, it will be responsible for classifying the input image into two groups: absence or presence of a forest environment. In this way, we could considerably reduce errors in the detection step with CNN. A brief summary of the pre-processing steps and their significant contributions are presented in Table 3.

The sequence of mathematical steps involved in the pre-processing stages is described in the following sections.

2.3.1. Color transformation

The first technique used in the proposed method is the color transformation of an image from the RGB standard to the HSV color space. The color transformations formulated by González-Moles, Ramos-García, and Warnakulasuriya (2021) are expressed in a generic way as shown in Eq. (1):

$$g(x, y) = T[f(x, y)] \tag{1}$$

where $f(x, y)$ is a color input image, $g(x, y)$ is an output image after transformation, and T represents a transformation operator on image f.

The RGB (Red, Green, black) color model is the most common standard found in digital displays such as computer monitors, televisions, and digital camera sensors. This color pattern was elaborated from studies of the human eye and how it is able to capture light and color information. Therefore, the RGB model was inspired by the characteristics of the human eye and chosen as the most suitable model for capturing and viewing digital images due to its ability to better sensitize the human eye.

Eventually, it may be necessary to carry out a transformation from one color system to another, such as converting from RGB to HSV (Hue, Saturation, Value). While the RGB color model is hardware-oriented and aimed only at viewing digital images, the HSV standard proved to be more suitable for processing in sensitive situations.

$$V \leftarrow max(R, G, B) \tag{2}$$

The expression (2) allows us to calculate the S, saturation channel:

$$S \leftarrow \frac{V - min(R, G, B)}{V} \tag{3}$$

The expression (3) is intended for the calculation of the H channel, of hue:

$$H \leftarrow \begin{cases} \frac{60(G-B)}{V-min(R,G,B)}, & if\ V = R \\ \frac{120+60(B-R)}{(V-min(R,G,B))}, & if\ V = G \\ \frac{240+60(R-G)}{(V-min(R,G,B))}, & if\ V = B \\ 0, & if\ R = G = B \end{cases} \tag{4}$$

Additionally, it is necessary to consider some significant features of this transformation. If the value of H is less than zero, calculate:

$$If\ H < 0,\ then\ H \leftarrow H + 360 \tag{5}$$

In general, encompassing the three channels H, S, and V, the values calculated at the output must be within this range of values:

$$0 \le V \le 1.0 \le S \le 1.0 \le H \le 360 \tag{6}$$

To finish the T transformation, the values for an 8-bit image f are converted per channel following the expressions:

$$V \leftarrow 255V,\ S \leftarrow 255S,\ H \leftarrow \frac{H}{2}(to\ adjust\ from\ 0\ to\ 255) \tag{7}$$

Table 3
A brief summary and description of the pre-processing steps.

Color transformation
Conversion of input images from RGB to HSV. The HSV standard proved to be more suitable for processing in forest fire situations.
Histogram and its normalization
This technique allows for transforming the digital image into a simplified statistical representation.
Histogram comparison
This technique allows calculating the similarity between the input image and the patterns of interest. Its output generates a vector of coefficients for each compared pattern.
Artificial neural network
Applying this ANN corrects some misclassifications of the histogram comparison step. Its output allows for detecting the presence or absence of a forest environment.

2.3.2. Histogram and its normalization

The second technique used in the proposed method is histogram calculation. The histogram calculation of a digital image is one of the fundamental techniques among several others in the domain of image processing. It provides statistical information about a digital image and is widely used for real-time image processing. It has the intensity levels in the range $[0,\ L-1]$ is a discrete function expressed by (8):

$$h(r_k) = n_k \tag{8}$$

where r_k represents the kth value of intensity and the number of pixels of the image with intensity r_k is designated by n_k.

Considering an image f of dimensions $M \times N$, where M and N are the dimensions of the width and height of f, it is customary to normalize a histogram by dividing each of these components by the total number of image pixels, expressed by the product MN. In this way, normalizing histogram results as per the expression (9):

$$p(r_k) = \frac{r_k}{MN},\ to\ k = 0, 1, 2, 3, \ldots, L-1. \tag{9}$$

In other words, it can be expressed that $p(r_k)$ is an estimate of the probability of occurrence of the intensity level r_k in an image f. Therefore, in a normalized histogram, the sum of all its components will be equal to 1.

2.3.3. Histogram comparison

The third technique used was the comparative analysis of histograms. Let $H1$ and $H2$ be the histograms of two images f_1 and f_2, respectively. The comparison between these two histograms can be done using a similarity measure. In this work, we use the measure of the correlation between two images, which is expressed by Eq. (10):

$$d(H1, H2) = \frac{\sum_I (H_1(I) - \bar{H}_1)(H_2(I) - \bar{H}_2)}{\sqrt{\sum_I (H_1(I) - \bar{H}_1)^2 \sum_I (H_2(I) - \bar{H}_2)^2}} \tag{10}$$

where,

$$\bar{H}_k = \frac{1}{2} \sum_J H_k(J) \tag{11}$$

where N represents the total number of bins used in the histogram calculation.

In this way, similarity measures are assessed by comparing any two histograms, where we obtain a result as a single numerical value expressed on a scale $[-1.0, +1.0]$ and how two similar histograms are related, with -1 specifies no similarity and $+1$ maximum similarity.

2.3.4. Artificial neural network

The fourth technique used in this work was the use of an ANN (Murtagh, 1991). ANNs are motivated by the activities of the biological neurons of the human brain. We use a two-layer feedforward neural network, containing an intermediate layer. The intermediate layer gives the ANN a greater capacity for computation with the approximation of continuous functions. The middle layer increases the learning capacity of the neural network when compared to a single-layer feedforward network. As it is a multilayer neural network, to train it with the supervised learning technique, it was necessary to use the generalized form of the Delta rule, known as the error backpropagation (Rumelhart, Hinton, & Williams, 1986).

In this work, the offline form of supervised learning was used, in which the selected training data remains unchanged. During the learning process, the model holds its synaptic weights adjusted with feedback from the error metrics measured at the output of the network. Further, the synaptic weights of the model remain fixed once its solution is estimated.

During the design of the ANN for classification purposes, we consider its structure as one of the depend on several factors, such as the nature and complexity of the problem, the size of the input data, prior knowledge of the problem, and how the data is represented, whether through features or in its original form. Therefore, to model a minimalist network, aiming to work with enhanced performance on a limited range of hardware architecture, it is necessary to investigate these factors from the chosen set of forest fire images.

The error functions estimated from the single-layer networks are calculated from the difference between the desired output and the actual output of the network. However, for MLP networks this procedure can be used only in the output layer since there are no desired outputs for its intermediate layers.

The backpropagation algorithm uses the gradient descent calculation to estimate the error of the intermediate layers considering the effect they have on the error at the output layer. In this way, the weights of the intermediate layers are proportionally adjusted by the feedback of the error in the output through the connections between the layers.

The use of multiple layers contributes to a transformation of the input data, in order to make it less complex along the intermediate layers to a tractable level at the output. Thus, a problem of a non-linear nature can turn into a linear approach at the output of the MLP.

Thus, an MLP network of a middle layer behaves through two successive transformations, expressed mathematically by $H(x; w_H)$, for the middle layer, and as $Y(H(x, w_H); w_s)$, relative to the output layer, where w_H corresponds to the weight vector of the middle layer and w_s represents the weight vector of the output layer.

Once the demand for using a multilayer neural network is identified, it becomes necessary to determine the activation functions of the intermediate and output layers, and the number of neurons in each layer. In this work, we used the sigmoidal activation function in the intermediate layers and the *argmax* function in the output layer.

In order to produce the output h of the neurons at the middle layer from the weight vectors $w1 = (w1_1, w1_2, \ldots, w1_n)^T$ and input vectors $x = (x_1, x_2, \ldots, x_n)$ we adopt a fully differentiable activation function, the sigmoidal activation function:

$$g(h) = \frac{1}{1 + e^{-h}} \tag{12}$$

In the output layer, we use the activation function *argmax* to generate the output y from the weight vectors $w2 = (w2_1, w2_2, \ldots, w2_n)^T$

Table 4
A summary of the ANN parameters.

Topology	Learn rate	Training algorithm	Activation function
3 layers (input, hidden, output) 10/128/10	0.01	Backpropagation	Sigmoid in the hidden layer and Argmax in the output

and output from the middle layer $h = (h_1, h_2, \ldots, h_n)$. This function is used to identify the maximum argument in the y output array. In other words, the $argmax$ function allows finding the class with the highest predicted probability and can be expressed as shown in Eq. (13):

$$argmax(\left[z_1, z_2, \ldots, z_n\right]) = \left[y_1, y_2, \ldots, y_n\right] = [0, 1, \ldots, 0] \tag{13}$$

where $y_i = 1$, if it is the only maximum value in the $\bar{z}$ vector.

An additional stage has been added to the network output to allow an adjustment that we will assess the detection threshold, represented by Ω. This threshold allows establishing a probability outcome at which the algorithm provides a reliable output, and is expressed in Eq. (14):

$$\Omega = \begin{cases} 1, & if\ p \geq 0.6 \\ 0, & if\ p < 0.6 \end{cases} \tag{14}$$

where p is the value associated with the class with the highest predicted probability.

The threshold value Ω was chosen based on the experimental trials carried out in the laboratory. Subsequently, it was normalized between the range of 0 to 1 to suit the environmental conditions of deploying the CNN model. Such normalized thresholds significantly contribute to the reduction of false fire detection.

Below, we present in Table 4 a summary of the parameters configured for the ANN model.

2.4. CNN architecture for wildfire detection

The CNN proposed by Silva et al. (2022) was built to be used with edge devices, where the EdgeFireSmoke method was computationally optimized to be embedded in 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 the work of Khan et al. (2019). Based on computational experiments, we have seen that two convolutional layers are the minimum requirement for solving this 4-class problem. We reduced the number of filters from 64 to 32 filters in each convolutional layer. Furthermore, a significant reduction in memory consumption was achieved by removing excess convolutional layers, and excess filters, and using a stride equal to 4 in the max pooling 2 layers. The proposed method has the smallest number of parameters. In addition, it can be linked to a CCTV system to detect fires and alert users through the internet.

In this work, we use the CNN architecture proposed in Silva et al. (2022) to detect forest fires from an image dataset. However, the method relies on a previous step that will guarantee the image to be processed was actually captured in a forest environment. Thus, the idea was to reduce the occurrence of possible false alarms when detecting fire in a non-forest environment. Furthermore, we evaluated the proposed ANN combined with other lightweight CNNs available in the literature.

2.5. Forest Fire Risk Index (FFRI)

The calculation of the risk of forest fire was formulated by Lourenço et al. (1997), in which they came to the conclusion that the risk of forest fire depends on three meteorological variables: temperature and relative humidity, as these contribute to the dryness of the combustibles and the outbreak of fire; and the wind speed, which contributes to the spread of fire over vegetation. Mathematically, this index is expressed as in Eq. (15):

$$FFRI = \frac{T}{H} + \frac{W}{100} \tag{15}$$

Table 5
Definition of risk classes of forest fires.
Source: Adapted from Lourenço (1991).

Risk	Degree of risk	Interval	Color scale
1	Reduced	$0 < FFRI \leq 0.5$	Black
2	Moderate	$0.5 < FFRI \leq 1.0$	Green
3	High	$1.0 < FFRI \leq 1.5$	Yellow
4	Very high	$1.5 < FFRI \leq 2.0$	Orange
5	Extreme	$FFRI > 2.0$	Red

T - Air temperature, in °C;
H - Relative air humidity, in %;
W - Wind speed, in km/h;

In this work, we adapt the nomenclature defined by Lourenço (1991), which expresses, through intervals, a scale of the risk of deflagration and propagation of forest fire, which we will call the Risk of Deflagration and Propagation Scale (RDPS). This scale will be useful to present an estimate of the fire risk to the system operator, specifically through visual feedback on the human–machine interface (HMI) (see Table 5).

In order to understand the degree of risk, we briefly illustrate it further on the basis of the theory formulated by Lourenço (1991).

1. **Reduced Risk:** Small outbreaks that normally do not cause a fire may appear.
2. **Moderate Risk:** Hotspots may appear and the fire progresses slowly in a green forest area. In dry areas, it can progress quickly with the wind.
3. **High Risk:** Combustible materials ignite quickly. Fires can be uncontrollable if not dealt with in time. Aerial surveillance and alertness are recommended.
4. **Very High Risk:** After ignition the fire is established quickly, with rapid propagation and intensity increase. It is suggested the action of brigades with intervention at strategic points to contain the advance of the fire.
5. **Extreme Risk:** Easy ignition, immediate propagation with great speed and intensity. It is a situation of maximum gravity, which must involve all available means of monitoring and combat.

2.6. Real-time data visualization (IoT-HMI)

We will need a fast method to view the images in real-time, capable of processing in synchronization with the camera's frame rate per second (FPS). Therefore, we will evaluate the proposed method and the classic deep learning methods in the literature. OpenCV version 4.5 library was used in this work alongside Python 3.6 language for the implementation.

The two main sources of information streamed through the internet, are the images from the IP camera and the meteorological data from the OpenWeather API. Therefore, we use the integration of IoT-HMI, which means an intelligent data visualization interface connected via the internet.

2.7. Datasets used to train the algorithm

Two datasets used in this work were compiled and augmented by the authors. All images in both datasets were captured from real forest fire scenarios. In this work, we have used 5209 images to train the ANN (Dataset 1) and 24 725 images to train each CNN separately in the same dataset (Dataset 2). A brief description of them is given below.

2.7.1. Dataset 1: Florestal images

The Florestal image dataset is a subset of the aerial images of wildfires dataset, in which ten classes were considered. Sample representation of this Florestal dataset can be seen in Fig. 2 and they are named from A to J. Their samples were crucial to train the ANN model and making it capable of recognizing a forest environment from the digital image is prohibited. In this way, ANN will allow the CNN detector to have a screening step that precedes it, in order to improve its performance in image-based forest fire detection. In all, there are 4516 images for training, 693 images for validation, and 1085 images for testing. The image format is *.jpeg* with a resolution of 512 × 512 pixels.

2.7.2. Dataset 2: Aerial images of wildfires

In the other dataset of aerial images of wildfires, to complete our conceptual model of wildfire and smoke detection from UAV images was built by Silva et al. (2022). In this dataset, public videos for fire detection from the UAVs were selected and processed to build the novel dataset. The videos recorded include fire, smoke, burned areas, and green areas, mainly in forested areas, but also in some urban areas. In total, 93 videos were available in the *.mp4* format, with 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 four classes:

1. *Burned-area (IAQ):* the IAQ class has 9348 captured images after the occurrence of fire. In general, the floor is observed to be black and the trees are without leaves.
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 was observed to be either 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, which includes images of the forest, dry vegetation, and pasture areas without any fire, smoke, or fog.

Fig. 1 shows examples of images from Dataset 1. Table 6 gives a description of the data. Combining these datasets, we have 49,452 samples and more than 93 different outdoor environments from 93 videos.

2.8. Training of the algorithms

The algorithms proposed in this work were trained in two stages. First, we trained the ANN with samples from dataset 1, which was partitioned into three parts: training, validation, and testing. We use 500 training epochs for the ANN with a fixed learning rate of 0.01. In the end, we obtained the weight vectors w1 and w2. Then, we proceeded with the training of the CNNs, separately, using dataset 2. This second dataset was partitioned into three parts similar to the previous dataset. At the end of the training, we obtained the models of each trained CNN with their respective weight vectors (.h5 and .json files). For all CNNs we did the training with 25 epochs of learning. The images were resized to 224 × 224 × 3 pixels and normalized between 0 and 1 before training. We use a batch size of 4 in all CNN training situations. We use SGD optimizer to train the models VGG16, Inception V3, Xception, DenseNet121, and VGG19. For the MobileNetV2 and EdgeFireSmoke models, we use the Adam optimizer. To train all these CNNs we used a learning rate of 0.001. We apply the transfer learning technique to train these CNNs using the original ImageNet weights, except for EdgeFireSmoke.

The source code of the proposed method and the databases are available in the repository.[1]

[1] https://sites.google.com/view/edgefiresmoke-plus/home.

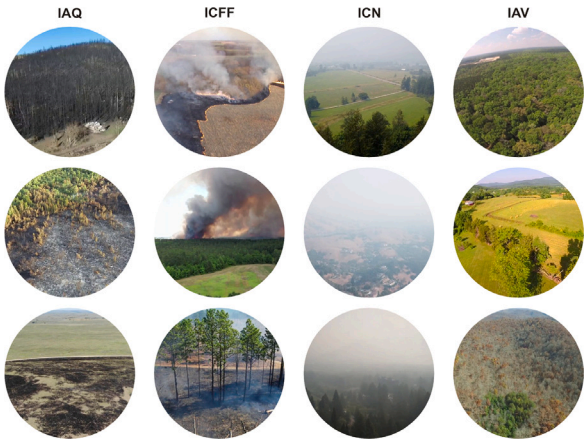

Fig. 1. Representative UAV images of Wildfires in Dataset 2. It contains real wildfire images and real hazy images. Labels: Burned-area (IAQ), Fire-smoke (ICFF), Fog-area (ICN), and Green-area (IAV).

Table 6
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	14 833	30	2804	4673	2928	4428
Test	24 727	50	4674	7790	4881	7382
Total	**49 452**	**100**	**9348**	**15 579**	**9762**	**14 763**

2.9. Evaluation metrics

In this work, four statistical metrics were evaluated from the scikit learn library: Accuracy, Recall, Precision, and F-score. Accuracy refers to the number of hits made among all predicted values, given by Eq. (16)

$$Acc\ (\%) = \frac{TP + TN}{TP + FN + FP + TN} \times 100 \tag{16}$$

The recall is the ratio of the correctly identified values among the set of given values represented by Eq. (17)

$$Recall = \frac{TP}{TP + FN} \tag{17}$$

Precision defines the ratio of correct values among the available set of values defined as shown in Eq. (18)

$$Precision = \frac{TP}{TP + FP} \tag{18}$$

F1-score is a measure of the balance between Precision and Recall among the unbalanced classes, which are represented as shown in Eq. (19)

$$F1\text{-}score\ (\%) = 2 \times \frac{Precision \times Recall}{Precision + Recall} \times 100 \tag{19}$$

3. Experimental results and discussion

In this section, the results obtained using the proposed method and the other six approaches are presented. The methods were evaluated on a computing terminal with an AMD Ryzen 5 3550H processor and 16 GB of RAM using TensorFlow 2.1, OpenCV 4.5, and Python 3.6 language.

Table 7 shows the result obtained during ANN training. The ANN training was performed with 500 learning epochs using the chosen configuration of 10 neurons in the input layer, 128 hidden neurons, and 10 neurons in the output layer. After training, an error of 0.28 was observed.

In Table 8 summarizes the result obtained after training the CNN and presents the closely relevant information about the evaluated

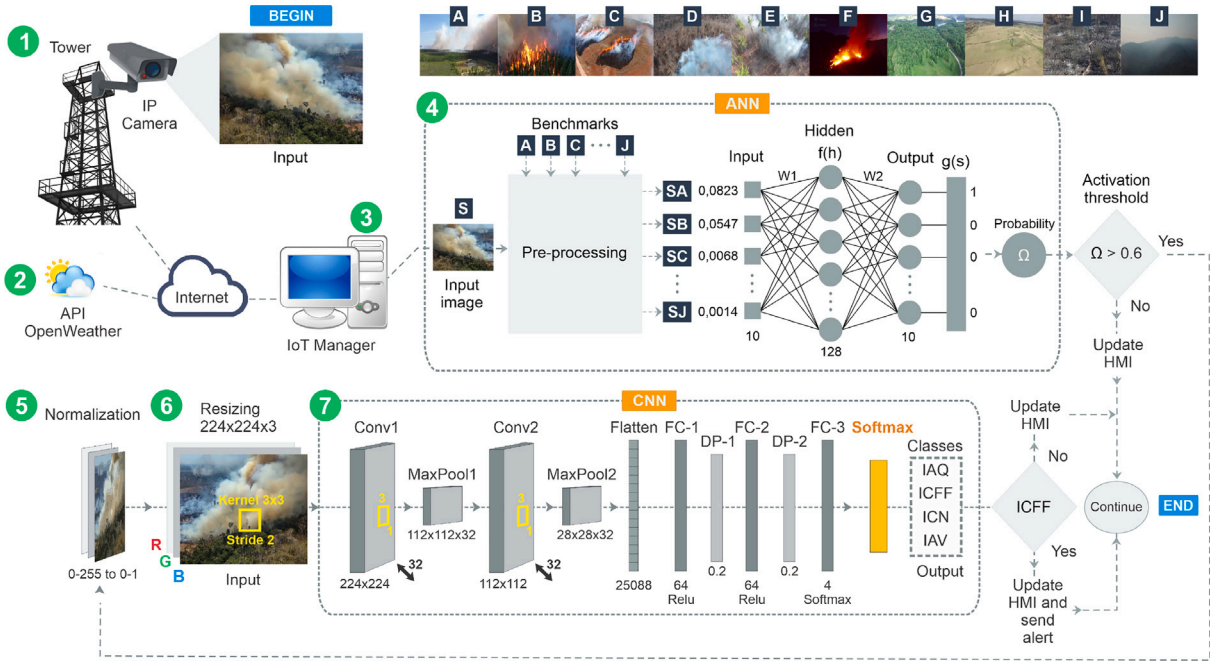

Fig. 2. Proposed conceptual model illustrated in seven steps. (1) It represents the way of capturing the images; (2) Meteorological data obtained from the internet; (3) Computer with the embedded algorithm; (4) ANN for sorting input images; (5) Normalization of images; (6) Image resizing for processing with CNN; (7) CNN proposed by Silva et al. (2022).

Table 7
Results of training ANN algorithm.

Source	ANN model	Layers	Epochs	Learn rate	Training loss
Dataset 1	Proposed ANN	Preprocessing + neural network: 10/128/10	500	0.01	0.28

models. It is observed that the EdgeFireSmoke (Silva et al., 2022) model has the most compact architecture, with only 7 MB of storage and a depth of only 10 layers. It is also the one with the fewest parameters, with a total of 1.62 million. During training, the EdgeFireSmoke model recorded an accuracy of 98.25% in the training and an accuracy of 95.45% in the validation set, where the obtained results were observed to be much better than the other evaluated models.

Subsequently, in the training MobileNet V2 model, it provides an accuracy of 95.25%. However, in the validation phase, it registers a performance below expectations. From the perspective of the architecture, the MobileNet V2 is the second most compact in terms of resource consumption with 14MB of storage and 3.54 million trainable parameters. The validation set serves as a preview of the test set, so the performance obtained in the validation will reflect on the result obtained in the test.

Other CNN models also showed relatively better results in training and validation, with the Inception V3 model providing comparatively inferior performance, with an accuracy of 84.03%. In all CNN models, the observations were made with 25 training epochs. As expected, more compact models have learned with a few dozen epochs, while sophisticated models demand a greater number of epochs. It is important to highlight that our main objective was to evaluate which CNN model will allow real-time processing, along with the other steps involved. Therefore, regardless of the number of training epochs, the processing time is expected to be lower.

Table 9 summarizes the result obtained with the test images in data set 1. The test data were partitioned into 10 subsets with an equal amount. The EdgeFireSmoke model performed best with the test images. It recorded an accuracy of 95.41% and maintained a balance between Accuracy, Recall, and F1-Score. In the experimentation, the second best CNN model with great performance in dataset 2 was VGG16, with 92.15% accuracy and also balanced in the other metrics. Despite being compact, an interesting feature for real-time processing, the MobileNet V2 model recorded inferior performance in the testing phase, with 53.63% of Accuracy, and 70.59% of Precision, and it is observed to be significantly above 50% in Recall and F1-Score.

A relevant experiment in this work was the real-time processing test. In this experiment, we used the two cameras presented in Table 2 and evaluated the performance of the proposed ANN together with the CNNs. The results achieved are shown in Table 10.

Analyzing Table 10, it can be seen again that the best performance was obtained with the proposed ANN, combined with the CNN originally proposed by Silva et al. (2022). This combination, which we named EdgeFireSmoke++, as it contains an extra pre-processing step, achieved a rate of 33 frames per second (FPS) with the HD resolution IP camera, 1280 × 720 pixels. Processing the images in VGA resolution from the webcam, EdgeFireSmoke++ achieved a rate of 40 FPS. The model proposed in this work needs only 540MB of RAM and is capable of working with IP cameras up to 30 FPS. It is possible to observe that with the other evaluated models, real-time processing becomes unfeasible, as they cannot follow the camera's FPS. The proposed method consumes 29.86 ms to process an image acquired from an IP camera, with a resolution of 1280 × 720 pixels.

Fig. 3 shows a plot that presents the outcomes of ANN learning during the 500 training epochs. Fig. 4 shows the analysis and learning curves of CNNs during the 25 training epochs. Finally, the plot in Fig. 5 shows in an orderly way the performance during the real-time testing obtained with the proposed ANN and its CNN variations. From the observations, it is clearly evident that the best performance in processing time was achieved by the EdgeFireSmoke++ method proposed in this work. The worst time performance was recorded by the ANN+VGG19 combination.

It is extremely important to emphasize that the requirements for the proposed method are: having a source of natural or artificial lighting in the environment; how cameras on the site need to be connected to the internet; the cameras must have a connection via the RTSP protocol; It is recommended that the computer on which the algorithm being

Table 8
Results of training deep learning models with 25 epochs.

Source	CNN model	Size (MB)	Depth	Parameters			Accuracy (%)	
				Trainable	Non-trainable	Total	Training	Validation
Dataset 2	EdgeFireSmoke++	7	10	1.62M	0.00M	1.62M	98.25	95.42
	VGG16	528	16	18.89M	14.71M	33.61M	90.29	91.88
	MobileNet V2	14	105	1.28M	2.25M	3.54M	95.25	53.84
	Inception V3	92	189	25.19M	21.80M	46.99M	84.03	82.36
	Xception	88	81	25.19M	20.86M	46.05M	87.13	82.53
	DenseNet121	33	242	20.99M	7.04M	28.03M	94.30	76.96
	VGG19	549	19	18.90M	20.02M	38.92M	86.87	88.73

Table 9
Results of test performance with 10 iterations.

Source	CNN model	Test			
		Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Dataset 2	EdgeFireSmoke++	95.41 ± 0.24	95.49 ± 0.28	95.38 ± 0.23	95.41 ± 0.24
	VGG16	92.15 ± 0.82	92.37 ± 0.96	92.51 ± 0.76	92.13 ± 0.82
	MobileNet V2	53.63 ± 1.65	70.59 ± 1.67	55.79 ± 1.76	54.20 ± 1.83
	Inception V3	82.65 ± 1.13	81.34 ± 1.14	82.19 ± 1.18	82.41 ± 1.14
	Xception	82.60 ± 1.04	85.76 ± 1.16	79.13 ± 1.27	81.49 ± 1.13
	DenseNet121	76.62 ± 0.82	81.53 ± 1.16	73.03 ± 0.95	72.51 ± 1.26
	VGG19	82,70 ± 1.46	87.13 ± 1.13	80.52 ± 1.75	82.70 ± 1.52

Table 10
Results of the real-time test, in milliseconds.

Source	Preprocessing	CNN model	FPS	Memory (GB)	Time (ms)			
					Mean	Median	Maximum	Minimum
Cam 1	Proposed ANN	EdgeFireSmoke++	33	0.54 ± 0.01	29.86 ± 2.59	29.86	54.13	22.11
		VGG16	4	0.89 ± 0.03	241.08 ± 9.81	241.70	289.25	221.70
		MobileNet V2	20	0.53 ± 0.01	49.46 ± 3.30	49.51	73.84	40.18
		Inception V3	12	0.77 ± 0.01	77.26 ± 3.54	76.77	89.81	68.64
		Xception	8	0.83 ± 0.01	120.48 ± 7.25	120.16	224.90	111.59
		DenseNet121	9	0.76 ± 0.02	100.39 ± 5.14	100.14	159.88	90.16
		VGG19	3	0.94 ± 0.02	301.14 ± 16.93	297.09	342.63	268.82
Cam 2	Proposed ANN	EdgeFireSmoke++	40	0.52 ± 0.02	24.77 ± 2.52	24.60	34.06	17.85
		VGG16	4	0.83 ± 0.04	241.22 ± 10.02	239.29	276.14	218.01
		MobileNet V2	25	0.48 ± 0.01	38.99 ± 2.94	38.91	48.36	30.29
		Inception V3	13	0.76 ± 0.01	75.77 ± 2.61	75.55	88.75	69.15
		Xception	8	0.79 ± 0.01	113.67 ± 3.08	113.50	130.89	105.79
		DenseNet121	10	0.68 ± 0.01	95.78 ± 4.74	95.00	111.34	84.31
		VGG19	3	0.88 ± 0.04	292.07 ± 6.33	292.07	319.90	274.89

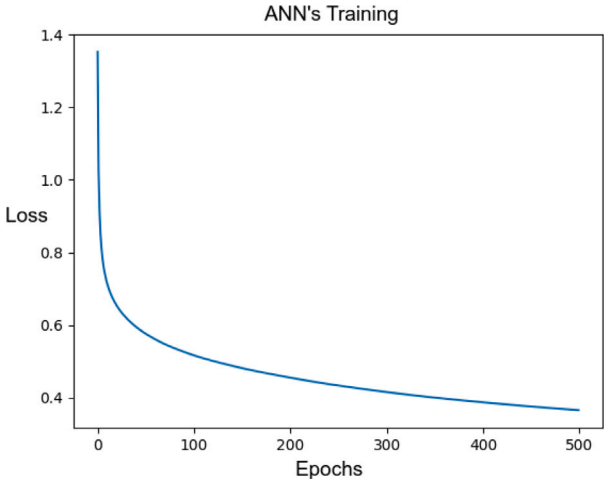

Fig. 3. ANN learning curve over 500 epochs with samples of dataset 1.

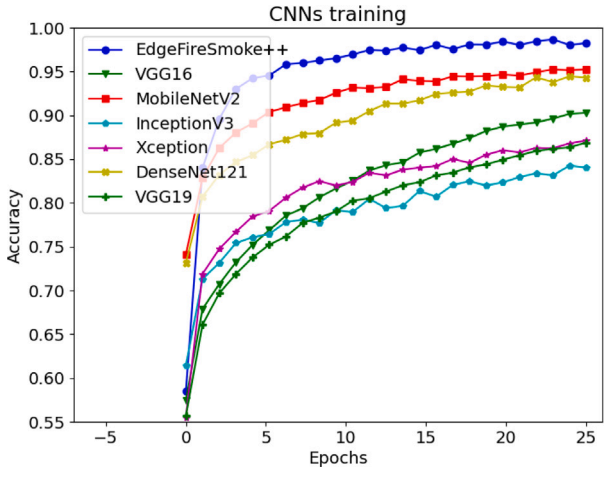

Fig. 4. Learning curves of CNNs over 25 epochs with samples of dataset 2.

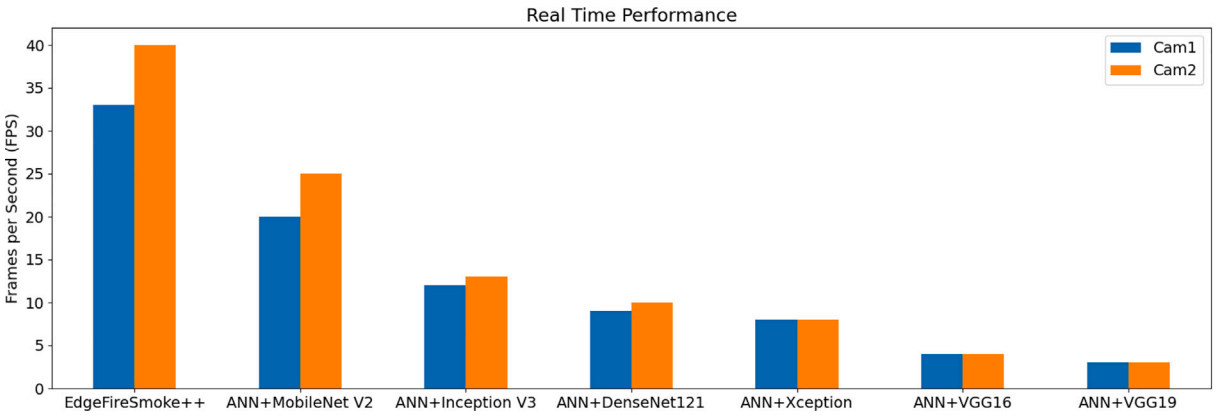

Fig. 5. Performance obtained with the proposed ANN and the CNN variations evaluated.

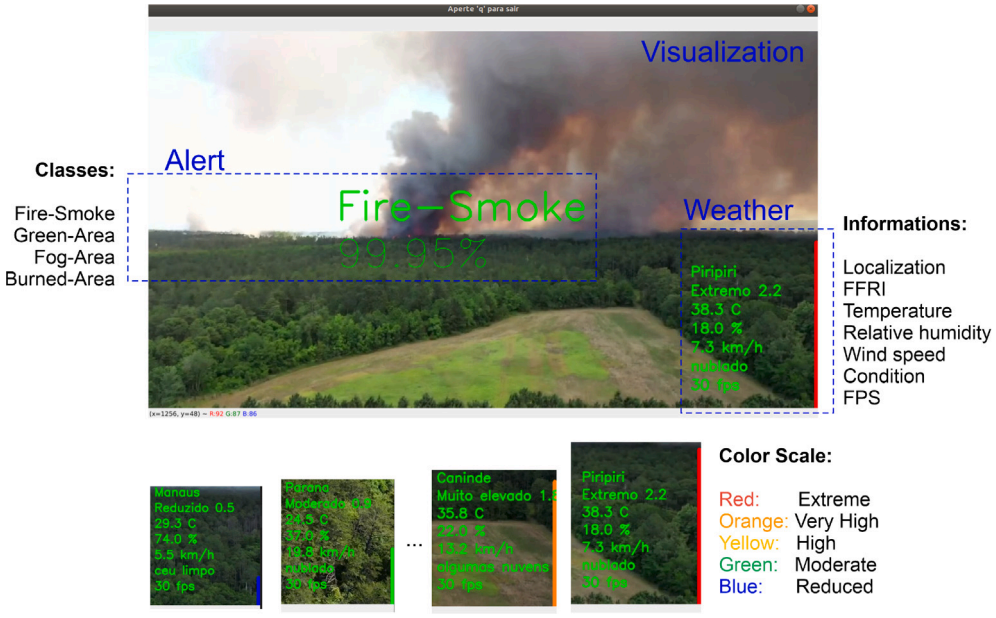

Fig. 6. The proposed IoT-HMI shows the algorithm predictions in real-time, the weather information from weather API and the risk of wildfires using a color scale.

processed is equipped with at least 16 GB of RAM, an 8-core processor, with uninterrupted internet connection.

Finally, we present an interactive data visualization interface connected to the internet, allowing the system to obtain real-time information from the monitored location, obtained from the forest fire detection algorithm. The experimental outcomes of the proposed algorithm can be visualized in Fig. 6.

4. Conclusion and future works

In this work, we presented a novel algorithm capable of detecting and classifying the occurrence of forest fires from the images captured through the CCTV systems. The algorithm presented improvements in relation to the work proposed by Silva et al. (2022). Among them, we present the inclusion of a pre-processing step by an ANN-based algorithm for similarity comparison and creating a data visualization interface connected to the internet. The EdgeFireSmoke++ proposed in this work proved superior to other CNN methods in the literature. In the evaluation performed on Dataset 2, considering the test set, we obtained 95.41% accuracy, 95.49% of precision, 95.38% Recall, and 95.41% of F1-score. The proposed algorithm recorded the best FPS rates from the HD IP camera recorded as 33 FPS and with the USB VGA camera at 40 FPS. For its operation, the proposed algorithm proved to be very light, being able to work on a 4-core CPU, 2.1 GHz, with an average consumption of 540MB of RAM memory. We hereby highlight two significant limitations of the algorithm: (1) the need to use the algorithm during the day, because the model training was done without special cameras, and (2) the need to use cameras connected to the internet. Therefore, the proposed algorithm, EdgeFireSmoke++, showed promise for forest fire detection by real-time video monitoring (CCTV). Its information visualization interface (IoT-HMI) constitutes an important advantage of use. In future work, we suggest investigating the performance of the model with special cameras to increase the accessibility of forest fire images during day and night conditions.

CRediT authorship contribution statement

Jefferson S. Almeida: Conceptualization, Methodology, Witing – original draft, Resources. **Senthil Kumar Jagatheesaperumal:** Investigation, Writing – review and editing. **Fabrício G. Nogueira:** Conceptualization, Investigation, Visualization, Resources. **Victor Hugo C. de Albuquerque:** Conceptualization, Methodology, Writing – review and editing, Visualization, Funding acquisition, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The data that has been used is confidential.

Acknowledgments

This study was partly financed by the Brazilian Federal Agency for Support and Evaluation of Graduate Education (CAPES). Jefferson S. Almeida acknowledges the sponsorship from CAPES, Brazil and the Federal University of Ceará (UFC), Brazil. This work is supported by the CNPq, Brazil via grant n° #3055172022-8.

References

Agarwal, P., & Jha, G. (2021). Forest fire detection using classifiers and transfer learning. In *2021 IEEE International conference on robotics, automation and artificial intelligence (RAAI)* (pp. 29–33). IEEE.

Chatragadda, A., Chalasani, S. H. V., Challa, N., Gupta, N. V. R., Oleti, P. P., & Amarendra, K. (2022). Convolutional neural networks based enhanced forest monitoring system for early fire detection. In *2022 7th International conference on communication and electronics systems (ICCES)* (pp. 425–432). IEEE.

Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1251–1258).

Fan, R., & Pei, M. (2021). Lightweight forest fire detection based on deep learning. In *2021 IEEE 31st International workshop on machine learning for signal processing (MLSP)* (pp. 1–6). IEEE.

González-Moles, M. Á., Ramos-García, P., & Warnakulasuriya, S. (2021). An appraisal of highest quality studies reporting malignant transformation of oral lichen planus based on a systematic review. *Oral Diseases*, *27*(8), 1908–1918.

Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4700–4708).

Hussain, T., Muhammad, K., Del Ser, J., Baik, S. W., & de Albuquerque, V. H. C. (2019). Intelligent embedded vision for summarization of multiview videos in IIoT. *IEEE Transactions on Industrial Informatics*, *16*(4), 2592–2602.

IBAMA (2017). Burn monitoring in satellite images. URL http://www.ibama.gov.br/consultas/incendios-florestais/consultas-monitoramento-de-queimadas/monitoramento-de-focos-de-queimadas-em-imagens-de-satelites.

INPE (2021). Monitoring of active spots by countries. URL https://queimadas.dgi.inpe.br/queimadas/portal-static/estatisticas_paises/.

Jagatheesaperumal, S. K., Muhammad, K., Saudagar, A. K. J., & Rodrigues, J. J. (2023). Automated fire extinguishing system using a deep learning based framework. *Mathematics*, *11*(3), 608.

Karjalainen, E., Sarjala, T., & Raitio, H. (2010). Promoting human health through forests: overview and major challenges. *Environmental Health and Preventive Medicine*, *15*(1), 1–8.

Khan, S., Muhammad, K., Hussain, T., Del Ser, J., Cuzzolin, F., Bhattacharyya, S., et al. (2021). Deepsmoke: Deep learning model for smoke detection and segmentation in outdoor environments. *Expert Systems with Applications*, *182*, Article 115125.

Khan, S., Muhammad, K., Mumtaz, S., Baik, S. W., & de Albuquerque, V. H. C. (2019). Energy-efficient deep CNN for smoke detection in foggy IoT environment. *IEEE Internet of Things Journal*, *6*(6), 9237–9245.

Li, Y., Shen, Z., Li, J., & Xu, Z. (2022). A deep learning method based on SRN-YOLO for Forest Fire Detection. In *2022 5th International symposium on autonomous systems (ISAS)* (pp. 1–6). IEEE.

Li, W., Xiaobo, S., Jun, C., & Ying, L. (2021). Research on forest fire image recognition algorithm based on color feature statistics. In *2021 6th International conference on intelligent computing and signal processing (ICSP)* (pp. 346–349). IEEE.

Life, W. W. (2022). Responsible forestry. URL https://www.worldwildlife.org/industries/responsible-forestry.

Lourenço, L. (1991). Uma fórmula expedita para determinar o índice meteorológico de risco de eclosão de fogos florestais em portugal continental.

Lourenço, L., Gonçalves, A. B., & Loureiro, J. (1997). Sistema de informação de risco de incêndio florestal. In *ENB, Revista técnica e formativa da escola nacional de bombeiros* (pp. 16–25).

Muhammad, K., Hussain, T., Del Ser, J., Palade, V., & De Albuquerque, V. H. C. (2019). DeepReS: A deep learning-based video summarization strategy for resource-constrained industrial surveillance scenarios. *IEEE Transactions on Industrial Informatics*, *16*(9), 5938–5947.

Muhammad, K., Hussain, T., Tanveer, M., Sannino, G., & de Albuquerque, V. H. C. (2019). Cost-effective video summarization using deep CNN with hierarchical weighted fusion for IoT surveillance networks. *IEEE Internet of Things Journal*, *7*(5), 4455–4463.

Muhammad, K., Khan, S., Palade, V., Mehmood, I., & De Albuquerque, V. H. C. (2019). Edge intelligence-assisted smoke detection in foggy surveillance environments. *IEEE Transactions on Industrial Informatics*, *16*(2), 1067–1075.

Muhammad, K., Rodrigues, J. J., Kozlov, S., Piccialli, F., & de Albuquerque, V. H. C. (2020). Energy-efficient monitoring of fire scenes for intelligent networks. *IEEE Network*, *34*(3), 108–115.

Murtagh, F. (1991). Multilayer perceptrons for classification and regression. *Neurocomputing*, *2*(5–6), 183–197.

Oliver, A. S., Ashwanthika, U., & Aswitha, R. (2020). Detection of forest fire using convolutional neural networks. In *2020 7th International conference on smart structures and systems (ICSSS)* (pp. 1–6). IEEE.

Omar, N., Al-zebari, A., & Sengur, A. (2021). Deep learning approach to predict forest fires using meteorological measurements. In *2021 2nd International informatics and software engineering conference (IISEC)* (pp. 1–4). IEEE.

Organization, W. H. (2022). Wildfires. URL https://www.who.int/health-topics/wildfires#tab=tab_1.

Park, M., Jeon, Y., Bak, J., Park, S., et al. (2022). Forest-fire response system using deep-learning-based approaches with CCTV images and weather data. *IEEE Access*, *10*, 66061–66071.

Rachana, P., Rajalakshmi, B., Bhat, T., Kaur, S., & Bimali, S. (2022). Comparative study of different methods for fire detection using convolutional neural network (CNN). In *2022 4th International conference on smart systems and inventive technology (ICSSIT)* (pp. 1759–1765). IEEE.

Reddy, P. R., & Kalyanasundaram, P. (2022). Novel detection of forest fire using temperature and carbon dioxide sensors with improved accuracy in comparison between two different zones. In *2022 3rd International conference on intelligent engineering and management (ICIEM)* (pp. 524–527). IEEE.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, *323*(6088), 533–536.

Sadi, M., Zhang, Y., Xie, W.-F., & Hossain, F. A. (2021). Forest fire detection and localization using thermal and visual cameras. In *2021 International conference on unmanned aircraft systems (ICUAS)* (pp. 744–749). IEEE.

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L.-C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4510–4520).

Silva, J., Huang, C., Nogueira, F., Bhatia, S., & Albuquerque, V. (2022). EdgeFireSmoke: a novel lightweight CNN model for real-time video fire-smoke detection. *IEEE Transactions on Industrial Informatics*.

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & 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).

Tahir, H. U. A., Waqar, A., Khalid, S., & Usman, S. M. (2022). Wildfire detection in aerial images using deep learning. In *2022 2nd International conference on digital futures and transformative technologies (ICoDT2)* (pp. 1–7). IEEE.

Tlig, M., Bouchouicha, M., Sayadi, M., & Moreau, E. (2022). Infrared-visible images' fusion techniques for forest fire monitoring. In *2022 6th International conference on advanced technologies for signal and image processing (ATSIP)* (pp. 1–6). IEEE.

Wang, S., Chen, T., Lv, X., Zhao, J., Zou, X., Zhao, X., et al. (2021). Forest fire detection based on lightweight yolo. In *2021 33rd Chinese control and decision conference (CCDC)* (pp. 1560–1565). IEEE.