Neural Computing and Applications (2023) 35:9349–9361
https://doi.org/10.1007/s00521-023-08260-2

S.I.: LATIN AMERICAN COMPUTATIONAL INTELLIGENCE

A hybrid method for fire detection based on spatial and temporal patterns

Pedro Vinícius A. B. de Venâncio[1] · Roger J. Campos[1,2] · Tamires M. Rezende[3] · Adriano C. Lisboa[3,4] · Adriano V. Barbosa[1,5]

Received: 1 April 2022 / Accepted: 6 January 2023 / Published online: 6 February 2023

Abstract
Fire detection is a vital task for social, economic and environmental reasons. Early identification of fire outbreaks is crucial in order to limit the damage that will be sustained. In open areas, this task is typically performed by humans, e.g., security guards, who are responsible for watching out for possible occurrences. However, people may get distracted, or may not have enough eyesight, which can result in considerable delays in identifying a fire, after much damage has occurred. Thus, the idea of having machines to automatically detect fires has long been considered an interesting possibility. Over the years, different approaches for fire detection have been developed using computer vision. Currently, the most promising ones are based on convolutional neural networks (CNNs). However, smoke and fire, the main visual indicators of wildfires, present additional difficulties for the vast majority of such learning systems. Both smoke and fire have a high intra-class variance, assuming different shapes, colors and textures, which makes the learning process more complicated than for well-defined objects. This work proposes an automatic fire detection method based on both spatial (visual) and temporal patterns. This hybrid method works in two stages: (i) detection of probable fire events by a CNN based on visual patterns (spatial processing) and (ii) analysis of the dynamics of these events over time (temporal processing). Experiments performed on our surveillance video database show that cascading these two stages can reduce the false positive rate with no significant impact either on the true positive rate or the processing time.

Keywords Deep convolutional neural network · Hybrid wildfire detection method · Temporal analysis

1 Introduction

Over the past years, much attention has been devoted to the prevention of uncontrolled fires, as they pose severe hazards. Such incidents can destroy property, harm the environment, endanger human and animal life and cause severe damage to financial infrastructures.

According to Brazil's National Institute for Space Research (INPE), approximately 600,338 fires were recorded in South America in 2021 and the first 10 months of 2022. Of these, about 338,580 cases were in Brazil [1]. A system that enables early fire detection followed by immediate intervention has the potential to significantly decrease such high numbers, reducing the damage caused to the environment.

Systems for early fire detection are typically based on physical sensors, such as thermal detectors and smoke detectors. These sensors are popular due to their low cost and simple operation [2]. However, they are usually associated with low accuracies, which results in high false alarm rates. Also, the need to transmit signals from the sensors limit their applications to indoor spaces [3].

Vision-based fire and smoke detection systems overcome many of the problems of systems based on point sensors. Cameras have the ability to scan large areas in short periods of time and can be integrated into local processing devices that run automatic fire and smoke detection algorithms. These algorithms can be classified into three categories, according to how the features used in their decision-making are obtained: (i) systems based on handcoded features (manual feature extraction) [4], (ii) systems based on deep learning (automatic feature

Pedro Vinícius A. B. de Venâncio, Roger J. Campos, Tamires M. Rezende, Adriano C. Lisboa, and Adriano V. Barbosa have contributed equally to this work.

Extended author information available on the last page of the article

extraction) [3, 5], and (iii) hybrid systems, based on both deep learning and handcoded features [6–9]. These algorithms can also be classified according to whether their decision rules are obtained automatically (e.g., by training a classifier) or manually (e.g., defined by a human expert). Deep learning based systems learn both their input features and their decision rules automatically. In turn, systems based on handcoded features may use classifiers or have their decision rules defined by a specialist.

Manually defining features for smoke and fire detection requires prior knowledge of the physics of such events. Also, if classifiers are not used, it is necessary to propose decision rules that involve an analysis of static and dynamic features, such as the use of color spaces [4, 10] and movement analysis [11, 12], among others [13, 14]. If classification is used, it is necessary to organize these features in a way that they can be used as input vectors to the classifier.

In general, conventional machine learning approaches do not take input data in raw format. Thus, some a priori knowledge is necessary to design a feature extractor that will transform the raw data into feature vectors that can be classified. On the other hand, more advanced deep learning architectures, such as convolutional neural networks (CNNs), avoid this difficulty and have substantially improved performance in automatic fire detection tasks [15–18]. These architectures, however, are associated with some important limitations, such as long training times, the need for rich and diverse datasets for training, a lack of control over internal processes, and high computational cost and memory consumption [19]. As a consequence, these architectures can produce false alarms (false positives) or miss real events (false negatives) [20], and they can also be financially impractical for large-scale environmental monitoring due to their high computational complexity.

To deal with the problem of high computational cost, several studies have proposed more efficient convolutional neural networks for fire and smoke detection. The main solutions consist of CNNs designed specifically for the task of detecting fire and smoke [5, 15, 16, 21], a process that requires significant human engineering, and optimizations to the CNN architectures [22]. As an attempt to improve performance in automatic fire detection, a hybrid method of deep learning and handcoded features, which brings together the main advantages of each approach, can be promising [6]. Combining spatial and temporal information to detect fire is a common approach [7–9]. In general, these methods are also composed of two stages. In the first stage, regions of interest (ROIs) are defined as a result of dynamic features extracted by analyzing color spaces, differences in motion, and flickering features between fire and other objects in the videos. In the second stage, CNNs are applied to the ROIs to extract spatial features of fire. This approach has greatly improved the accuracy of fire detection.

Considering the efficacy of hybrid methods, this work proposes a new hybrid fire detection method. This is an extension of a previous work [20], incorporating a new temporal detector and more efficient CNN architectures. Our approach changes the order of the first and second stages of the hybrid approaches mentioned above, using a spatial detector as the first stage followed by a temporal analysis technique as the second stage. The spatial detector is implemented by a 2D CNN responsible for detecting events in each frame of the input video stream. In turn, the temporal stage analyzes the detection history produced by the spatial detector with the goal of verifying whether the detections exhibit fire or smoke behavior over time. Therefore, our temporal stage does not analyze color or movement features, but rather features produced by the spatial detector.

Two approaches are proposed for the temporal analysis technique. The first one, the *area variation technique* (AVT), analyzes the temporal variation of the area of the bounding boxes produced by the spatial detector. The second approach, the *temporal persistence technique* (TPT), analyzes the temporal persistence of the bounding boxes in order to determine whether the fire or smoke detections are sustained over a certain period of time. By adding a second stage (the temporal stage) to the output of the spatial stage, our method increases the detection level by decreasing false detections without significantly decreasing the true positive rate. Also, the temporal stage adds very low computational cost compared to the first stage, which is implemented as an YOLO algorithm. In this work, hybrid systems using both approaches (AVT or TPT) are compared to FireNet [3] and MobileNet [5], two CNN-based models that have been applied to fire detection.

The rest of this paper is organized as follows. Section 2 describes the proposed fire detection system. Section 3 presents D-Fire, an image and video database we have developed and use to train and test our fire detection systems. Sect. 4 presents and discusses the results obtained from training the networks and from deploying the hybrid system to real situations. Finally, Sect. 5 concludes our research.

2 Proposed approach

The fire detection approach we propose is composed of two sequential stages: (i) spatial detection, which consists of identifying and locating fire and smoke events on the scene based on spatial patterns of the input video stream, and (ii) temporal analysis of the events detected in the previous stage, in order to make a final decision on whether a fire is actually taking place. The system is shown in Fig. 1. The first stage (spatial detection) is implemented by an object detector based on a two-dimensional convolutional neural network (2D CNN) trained on a large and diverse dataset. The output generated by the first stage is then fed into the second stage, the temporal analysis technique, which is responsible for making a final decision based on certain assumptions about fire behavior, such as expansion and persistence over time. Thus, an alarm will only be set off if both visual and temporal patterns of fire or smoke are identified on the input video stream.

2.1 Fire detection using deep learning

Interest in deep learning techniques has grown steadily over the past decade. Computer vision is certainly one of the areas most impacted by advancements in this area, as evidenced by the *ImageNet Large Scale Visual Recognition Challenge (ILSVRC)* [23, 24]. The 2012 edition of the challenge was won by AlexNet [25], the first CNN to achieve outstanding results in the image classification problem, taking a significant lead over the second-best approach of the competition that year. Since the introduction of AlexNet, accuracy has improved significantly, to the point of surpassing that of an average person in many image classification tasks [26].

Although convolutional neural networks designed for image classification tasks usually provide encouraging results, they may not be able to support correct decisions in fire control and eradication. Object detection is arguably better suited for identifying fire and smoke, given that it also provides a location, which is especially of interest when dealing with large, open areas.

Object detection poses an additional difficulty compared to object classification: it must not only identify the existence of an object, but also its location on the image. Initial approaches to object detection used a sliding window, which went through the image until an object was found. Despite working well for some applications, sliding windows are computationally inefficient, as they run the CNN multiple times, once for each window position. This can lead to long response times which, in emergency situations such as fire detection, can cause a variety of human, environmental and financial damages. Later, region proposal methods were introduced. In these methods, regions of interest are first identified and then the presence, and the identification, of the object in these regions is determined. This type of method became known mainly for the R-CNN series [27–29] and is still widely used.

Assuming that a network has enough information to determine whether an object is present in an image, it seems plausible to believe that the network also has information to find the location of the object in the image. This is the idea behind the *you only look once* (YOLO) algorithms [30–33]. This joint approach helped YOLO become the state-of-the-art method for fast object identification and is the reason why we use it in the spatial detection stage of our method.

YOLO uses a single deep convolutional neural network to simultaneously predict class probabilities and bounding box positions directly from images in a single pass. Its original version was proposed by Joseph Redmon in 2016 [30] and later improved in two other versions (YOLOv2 and YOLOv3) in a self-authored and open source framework developed in the *C* language called Darknet [34]. After Redmon announced he would stop his computer vision research due to the massive use of his algorithms for military purposes, Alexey Bochkovskiy legally and officially continued the development of the YOLO series and its framework by publishing YOLOv4 [33] and later Scaled-YOLOv4 [35]. These new networks are usually released in two versions: tiny architectures designed for resource-constrained devices and large architectures designed for Graphics Processing Units (GPUs).

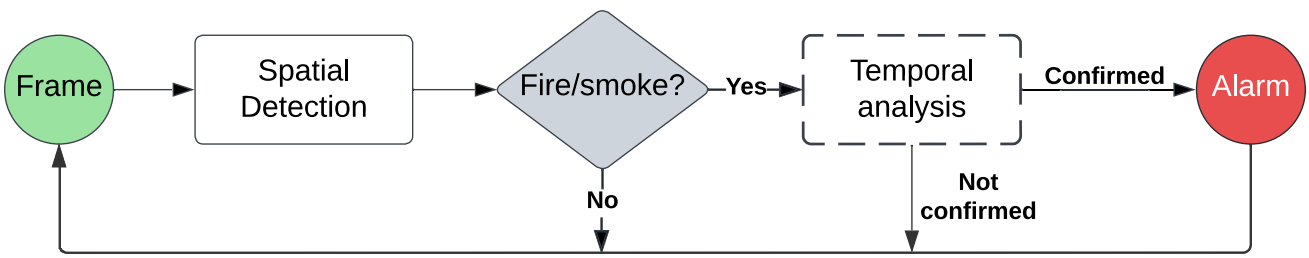


Fig. 1 Proposed fire detection method. The fire validation process undergoes multiple confirmation steps in order to reduce false alarms

YOLOv5 was released shortly after YOLOv4. Although the name suggests a continuation of the traditional YOLO series and its Darknet framework, YOLOv5 was proposed independently by Jocher et al. [36] in the PyTorch framework [37], an open-source Python library for tensor computation with strong GPU acceleration. Although the authors are not directly related, the YOLOv5 network was heavily inspired by the latest YOLO architectures at the time, being also made up of backbone, neck and head components [33, 36]. The main contribution of YOLOv5 is its backbone, where a focus structure replaces the first three layers of the YOLOv3 network, preserving CSPDarknet53 [33] as the feature extractor. In addition, YOLOv5 is lighter, faster to train and more scalable for real-world applications. Until the moment, it provides five models of the same overall structure at different sizes (i.e., different number of learnable parameters), which are YOLOv5n, YOLOv5s, YOLOv5m, YOLOv5l and YOLOv5x (referred to as *nano, small, medium, large* and *extra-large*, respectively). In this work, we analyze several of the most recent architectures of the YOLO series for fire detection, discussing the particularities of each one.

2.2 Temporal analysis techniques

In order to detect and classify an object in a scene, it is necessary to identify its visual patterns and, in the case of videos, to analyze its temporal behavior. When analyzing the behavior of fire and smoke, some assumptions can be made: (i) if there is fire, there is smoke (a flash of light unaccompanied by smoke is usually from a different light source); (ii) smoke usually starts white and becomes darker as the material burns; and (iii) the shape and intensity vary over time.

Based on these assumptions, our method implements a temporal analysis of the bounding boxes detected by the YOLO network in an attempt to reduce the false positive rate. Two approaches were used for the temporal analysis. In the first one, the temporal persistence of the spatial detections (i.e., the detections produced by the spatial detector) is analyzed in order to determine whether they persist over time. If they do, the fire or smoke event is confirmed and an alarm is triggered. In the second approach, an algorithm analyzes how the area of the bounding boxes provided by the spatial detector vary over time in order to determine whether this pattern matches what is expected for fire and smoke. These approaches are described in Sects. 2.2.1 and 2.2.2, respectively.

2.2.1 Temporal persistence

The proposed *temporal persistence technique* (TPT) consists of analyzing whether the fire or smoke detections last for a certain period of time. We propose this technique because we have observed that many false positives come from static objects such as car lights and streetlights. Typically, these false alarms last only a few frames, as the scores of the objects detected in them are usually very close to the confidence threshold, i.e., to the minimum score for a detection to be considered true by the convolutional neural network. In such scenarios, the scores of the detected objects will be just above the confidence threshold in some frames (which would trigger the alarm) and just below the threshold in other frames (which would not trigger the alarm), causing detection to vary considerably within a frame window. To avoid this erratic detection behavior, our system only triggers the alarm after detection has persisted over a certain amount of frames (temporal persistence).

The temporal analysis is performed on a buffer corresponding to the last w_p frames (*buffer size*) of the input video stream.[1] For each new arriving video frame, the corresponding position in the buffer is marked as TRUE if there is at least one fire or smoke object detected in the frame; otherwise the buffer position is marked as FALSE (see Fig. 2). A persistence coefficient is then computed as the ratio between the number of TRUE values in the buffer and the total buffer size. If the persistence coefficient is greater than a pre-defined persistence threshold t_p, then fire or smoke detection is confirmed and an alarm is triggered. TPT helps minimize the number of false alarms because the events causing false detections do not show a sustained trend over time.

2.2.2 Area variation

Similar to TPT, we propose another temporal analysis technique based on temporal behaviors that we observe from the fire. Specifically, we formulated this approach after analyzing the fire expansion behavior in forest environments and the false alarms returned by the spatial detector. The objective is to investigate the bounding boxes returned by the YOLO algorithm and their dynamics, which can be quite peculiar: it starts gradually but can take significant proportions over time (see Fig. 3).

The *area variation technique* (AVT) is a multi-step process that consists of tracking fires detected by the spatial detection stage in order to analyze their behavior over time. When the detector identifies fires in a video frame, the bounding boxes associated with them are given as input to the tracker. Subsequently, the tracker determines the centroid of each bounding box from its coordinates.

Therefore, centroids are determined for every frame where at least one of the classes is detected. In case the identified fires given to the tracker appear for the first time,

[1] This is a circular buffer initially filled with False values.

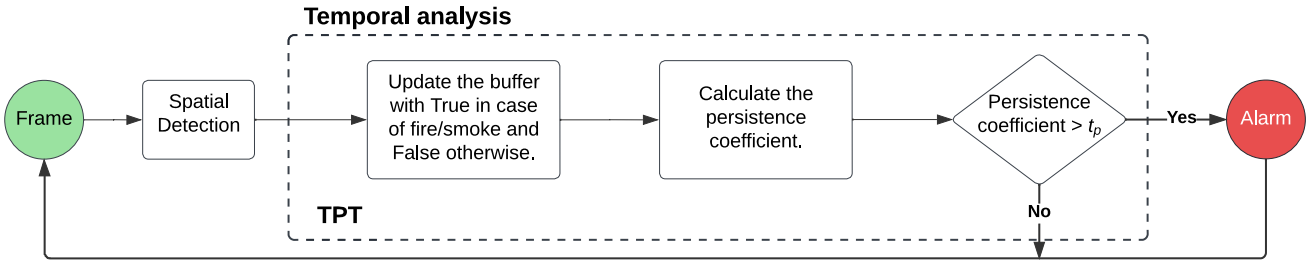


Fig. 2 Temporal persistence technique (TPT) is a process based on the persistence of detected fire and smoke objects in window size

Fig. 3 Forest fires in early stages typically exhibit an expansion behavior over time

since the initialization of the system, a unique identifier is determined for each object. Otherwise, the bounding boxes in the current frame Q_t are associated with those that have been already tracked in the previous frame Q_{t-1}.

Afterward, Euclidean distances between the centroids of the objects in frames Q_t and Q_{t-1} are calculated. Finally, each object in frame Q_t is associated with the object in frame Q_{t-1} whose centroid is closest. This is equivalent to choosing the subtle movement for the fire and smoke objects across consecutive frames. If there are more detections than events being tracked, that is, in case there are more detected objects in frame Q_t than in frame Q_{t-1}, a unique identifier is attributed to the object that was not associated with any object in the previous frame. However, if an object is not associated with another object from the previous frame for a total of m consecutive frames, its bounding box is removed of the historical record, and its numeric identifier is available for future assignments. This process is shown in Fig. 4.

In order to increase the system efficiency, the tracker determines whether the temporal behavior of the detected objects matches to fire or smoke in their initial stages. The system monitors the size (area) of the detected bounding boxes over time. If the areas grow above a certain threshold t_a over a time span of w_a frames, the fire or smoke detection is confirmed and an alarm is triggered.

The coefficient of variation is used to quantify the temporal growth rate of the detected objects. It is calculated over all frames in the current time window, as the ratio between the standard deviation and the mean of the detected objects' areas. The area variation technique (AVT) minimizes the number of false alarms because the main events that cause the false detections are static or do not show a growth trend over time, such as car lights, raindrops and solar reflections.

3 The D-Fire dataset

Since the proposed fire detection system consists of a spatial detection stage and a temporal analysis stage, training and evaluating it requires two databases, one composed of images and other composed of videos. The former is used to train and test different versions of the YOLO network as the first stage, whereas the latter is used to test the efficiency of both proposed temporal analysis techniques as the second stage, as well as adjustment of their hyperparameters (i.e., window size and threshold).

Regarding the image database, we used D-Fire [38], our free, community-available dataset designed specifically for fire classification and localization tasks. Currently, D-Fire contains 21,527 images with their respective labels, i.e., class identifiers and normalized bounding box coordinates. Among them, 1164 contain only fire, 5867 contain only smoke, 4658 contain both fire and smoke, and 9838 contain negative examples (i.e., neither fire nor smoke). However, most images have more than one occurrence of a given class.

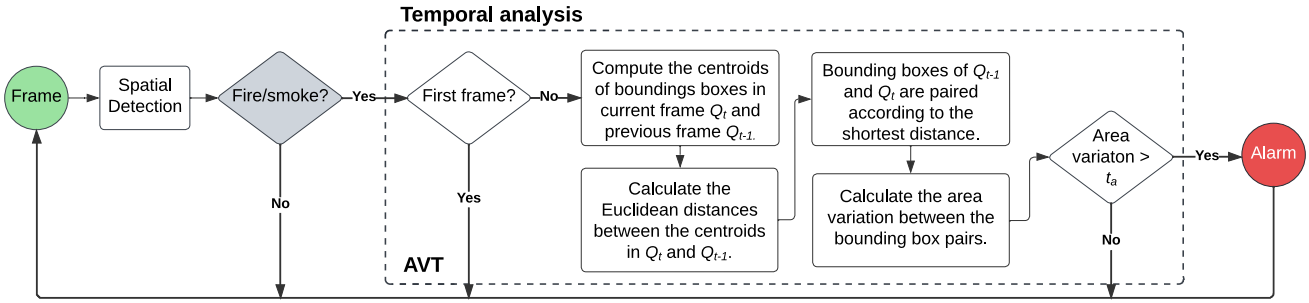


Fig. 4 Area variation technique (AVT) is a multi-step process based on the centroids of detected fire and smoke objects

D-Fire images were collected mainly from the Internet and from surveillance cameras of the Put out the Fire! ("Apaga o Fogo!") website [39], where our proposed system is running in production. All detections are notified by the website, which sends along a short video of the event so that a human can confirm if the event has been correctly (true positive) or incorrectly (false positive) detected by the system. In addition, we generated some synthetic images of long-distance fires and performed some legally conducted fire simulations in green areas of the Technological Park of Belo Horizonte (BH-TEC), Brazil. The dataset is quite diverse in terms of the smoke type (color, shape, texture, density, size), the fire origin (forest, industry, buildings), ignition source (impact, electrical sparks, smoking materials, fuel-fired objects), the camera's view point (views of drones and firefighting planes, environmental monitoring towers, ordinary citizens), confusing elements and noisy environments (clouds, fogs, shiny objects). Figure 5 shows some challenging scenarios available in the dataset. We invest in the diversity with the purpose of inducing learning algorithms to maximize their knowledge and, therefore, generalize, when well trained, in any fire situations.

Several studies and research groups use a small number of videos to validate their approaches [40–43]. Generally, collecting videos from the Internet is not an easy task, since most of them have copyright and privacy rights, as well as they can be too heavy to process or of poor quality when compressed. In order to build the video database to carry out the experiments of the second stage of our system, we collected 100 videos. Half of them correspond to true positives, whereas the other half correspond to false positives (these contain elements that pose several difficulties to the detection networks, such as building lights, fogs, clouds, dusts and raindrops). This sample of videos are also available in our public repository [38].

4 Experimental results

The first step of our method is to train the convolutional network that will be responsible for the detection of fire and smoke. For this purpose, we considered four YOLO networks, two deeper (YOLOv4 and YOLOv5l) and two shallower ones (Tiny YOLOv4 and YOLOv5s). We selected the best network in terms of predictive performance and the best network in terms of detection speed to investigate which one behaves as the best first stage in our proposed hybrid fire detection method. Then, we evaluated the integration of these networks with each of the temporal analysis techniques described in Sect. 2.2 and selected the best combination of stages, i.e., the one that produced the best values in a set of predictive performance metrics. We also evaluated the impact on frame rate when adding the temporal analysis stage to the decision process. All experiments were conducted using a single NVIDIA Quadro P5000 GPU on a machine with 256GB of RAM.

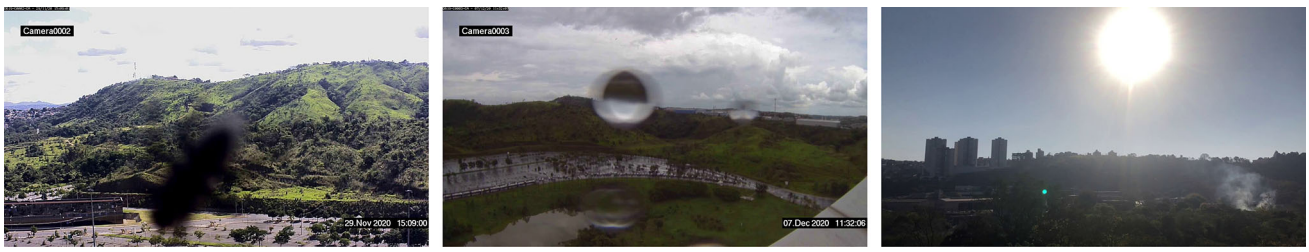


Fig. 5 Some examples of challenging scenarios that compose the D-Fire dataset [38]. On the left, there is an insect on the camera lens, obstructing the view to the mountain. In the center, there are raindrops scattered across the camera lens, which can be mistaken for smoke. On the right, there is the reflection of the sun, which can be confused with saturated fire

Table 1 Best hyperparameters found for training the deep detection networks

Network	Image size	Epochs	Learning rate	Batch size	Momentum	Weight decay	mAP (%)
Tiny YOLOv4	640	112	0.0020	64	0.900	0.0005	62.12 ± 0.32
YOLOv4	416	112	0.0013	64	0.949	0.0005	73.98 ± 0.40
YOLOv5s	640	100	0.0100	16	0.937	0.0005	77.04 ± 0.15
YOLOv5l	640	100	0.0100	16	0.937	0.0005	77.62 ± 0.37

The mAP values, denoted by $\mu \pm \sigma$, represent the mean μ and the standard deviation σ of the evaluations on each validation set from the five folds

The source code and models are available at https://github.com/pedbrgs/Fire-Detection.

4.1 Detection stage

To train the candidate networks to be used in the first stage of our hybrid system, the D-Fire dataset was split into training (17,221 images; 80% of the data) and testing (4306 images; 20% of the data) partitions. Before training the networks using the mini-batch gradient descent method, we performed an optimization of the hyperparameters of the networks, namely, the image size in the RGB color space, the number of epochs, the learning rate, the batch size, the momentum and the weight decay. In the training partition, we employed a grid search with five folds cross validation and selected the best hyperparameter configuration for each YOLO network based on the *mean average precision* (mAP) criterion, which is given by the mean value of the area under the precision-recall curve over all classes. Thus, the best hyperparameter configuration for each network was defined as the one resulting in the highest average mAP obtained over the five validations sets associated with the five folds. In this work, we use the mAP computed with an *intersection over union* (IoU) threshold of 0.50, referred to as mAP@0.50. The best hyperparameters selected for training each of the networks are shown in Table 1.

After tuning the hyperparameters, we trained each network six times on the entire training set and evaluated them on the test set, using the best hyperparameters for each network. The results are shown in Table 2, which also shows, besides the mAP, the average precision (AP) for each class, the harmonic mean of precision and recall, known as F_1-score, and the computational cost measured in billions of floating point operations (BFLOPs).[2]

From Table 2 we can see that the YOLOv5 networks learned the visual patterns of fire and smoke better than the YOLOv4 networks, as their predictive performance metrics are higher. Despite a slight improvement in fire detection compared to YOLOv4, the YOLOv5 networks presented greater difficulty in detecting fire than smoke, a trend we already observed with YOLOv4 networks in our previous work [20]. This happens because, in the early stages, the fire objects in the D-Fire images are typically smaller than the smoke objects.

A comparison between the lighter versions of the YOLOv4 and YOLOv5 networks (Tiny YOLOv4 vs YOLOv5s) shows an average difference of 15.14 p.p. in the mAP. The same comparison between the deeper networks (YOLOv4 vs YOLOv5l) shows an average difference of 2.89 p.p. in the mAP. In terms of computational cost, the YOLOv5 networks also outperform the YOLOv4 networks for a same image size. For an image with 640×640 pixels, the difference between the lighter networks is 0.27 BFLOPs (16.07 vs 15.80), whereas the difference between the deeper networks is 33.20 BFLOPs (141.00 vs 107.80).[3]

Based on the results above, we chose the YOLOv5 networks (the median model of each network obtained in the six training runs) for the first stage of our hybrid method rather than their YOLOv4 counterparts. The YOLOv5s network can be a good choice for applications running on low-power, resource-constrained devices, as it has a computational cost 6.82 times lower than the YOLOv5l network. On the other hand, the YOLOv5l network can be a good alternative for applications without hardware constraints, as it has a better predictive performance according to the mAP, the AP_{smoke}, and the F_1-score metrics. The performance difference relative to the YOLOv5s network is considerably small (on average, 0.95 p.p), but it can contribute to more robust detections in environmental surveillance scenarios, where the camera is installed far away from the monitoring area and therefore smoke becomes the main visual indicator of a wildfire.

[2] All discussions about computational cost in this work are based on the YOLO networks designed for fire and smoke detection ($C = 2$ classes). However, we note that the computational cost increases as C increases.

[3] The forward propagation of a 640x640 RGB image through a YOLOv4 network, not shown in Table 2, requires 141.00 BFLOPs.

Table 2 Performance metrics, on the test set, of the networks trained with their best hyperparameters

Network	Image size	mAP (%)	AP_{smoke} (%)	AP_{fire} (%)	F_1-score	BFLOPs
Tiny YOLOv4	640	63.01 ± 0.31	61.76 ± 0.44	64.26 ± 0.22	0.61 ± 0.00	16.07
YOLOv4	416	76.21 ± 0.35	82.82 ± 0.66	69.59 ± 0.35	0.74 ± 0.01	59.57
YOLOv5s	640	78.15 ± 0.15	83.85 ± 0.44	**72.45 ± 0.33**	0.76 ± 0.00	**15.80**
YOLOv5l	640	**79.10 ± 0.36**	**85.88 ± 0.35**	72.32 ± 0.52	**0.78 ± 0.00**	107.80

The metric values, denoted by $\mu \pm \sigma$, represent the mean μ and the standard deviation σ of the evaluations over each of the six training runs. The best average result obtained in each metric is shown in bold

4.2 Temporal analysis stage

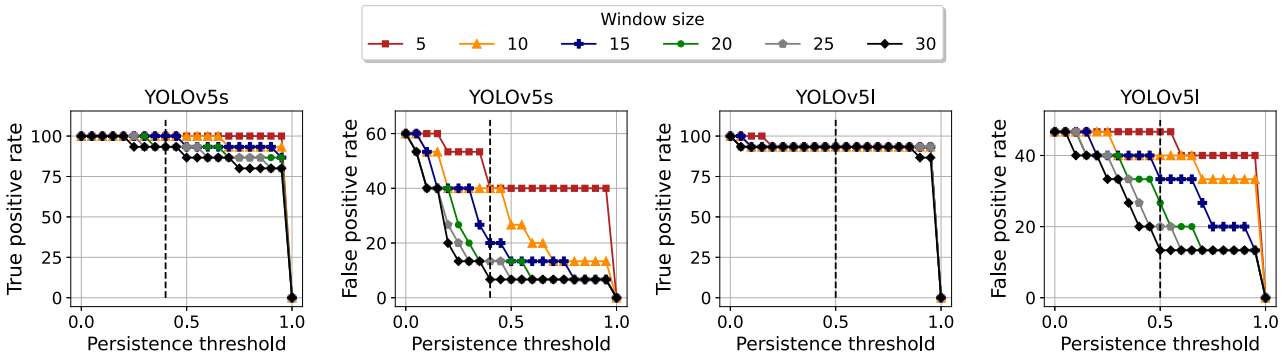
After defining the candidate detection networks, the next step is to investigate the integration of each network with each of the temporal analysis techniques, one by one, to compose the possible hybrid systems for fire and smoke detection. However, the performance of these techniques heavily depend on their hyperparameters, namely the window size and area threshold for the AVT, and the window size and persistence threshold for the TPT. For example, in both techniques, large values for the threshold can suppress many true fire detections, whereas small values may not be sufficient to suppress false positives. On the other hand, a small window size may not be adequate to capture the temporal patterns of the object identified as fire or smoke, whereas a large window size can be computationally expensive to process.

In order to choose the values of the hyperparameters, we performed a grid search over a randomly selected subset consisting of 30% of the videos, with 15% of them containing real fires and the other 15% containing objects that could be confused by the network with fire or smoke, such as fogs, raindrops, clouds, and building lights, to name a few. For the temporal persistence technique, we evaluated the persistence threshold t_p between 0 and 1 in steps of 0.05 and the window size w_p between 5 and 30 in steps of 5 frames. For the area variation technique, we evaluated the area suppression threshold t_a between 0 and 0.2 in steps of 0.01 and the window size w_a between 5 and 30 in steps of 5 frames. Figure 6 shows the influence of the hyperparameters on the performance of the hybrid systems according to the true positive rate (TPR) and the false positive rate (FPR). In calculating these evaluation metrics, we consider (i) a true positive (TP) if there is a fire in the video and the system detects it in at least one video frame, (ii) a false positive (FP) if there is no fire in the video and the system detects one in at least one video frame, (iii) a true negative (TN) if there is no fire in the video and the system does not detect a fire in any video frame, and (iv) a false negative (FN) if there is a fire in the video and the system does not detect it in any video frame. We chose this evaluation method, where a video is evaluated according to the predictions in all its frames, because we think it is more appropriate for real-time applications. If the model predicts a false positive in a single frame of the video, we assume that it misses the entire video, as this error would possibly trigger a false alarm in such a situation, causing unnecessary firefighter efforts.

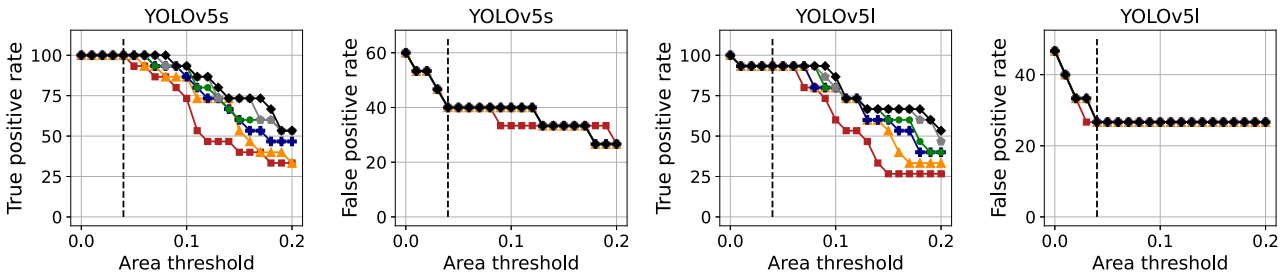
Overall, we can observe in Fig. 6 that, as the threshold of each temporal analysis technique increases, both the true positive rate and the false positive rate of the system decrease. In other words, the system becomes more "rigid", suppressing more detections, whether true positives or false positives. The best threshold value for both techniques will be the one that reduces the FPR as much as possible but without considerably degrading the TPR.

From Fig. 6a we can see that, for both networks, the TPR curve falls quite more smoothly than the FPR curve, which is a good behavior. However, the predictive performance of the YOLOv5l network is less sensitive to variations in the persistence threshold, such that decreases in the TPR and the FPR are more evident only for higher threshold values. Regarding the window size, larger values advance the downward trend of the curves, while smaller values delay this trend. Overall, we note that large window sizes associated with small threshold values allow larger decreases in the false positive rate with little impact on the true positive rate, which is desirable. Thus, we chose $t_p = 0.4$ and $t_p = 0.5$ for YOLOv5s and YOLOv5l (dashed vertical lines in Fig. 6a), respectively, and the window size $w_p = 30$ for both networks.

From Fig. 6b we notice that, for both networks, the true positive rate falls sharply for $t_a \geq 0.06$, especially for smaller window sizes. On the other hand, the false positive rate falls substantially until the threshold $t_a = 0.04$ and then flattens out, with further decreases only in the case of the YOLOv5s network. Based on that, we conservatively chose the threshold $t_a = 0.04$ for both networks. For the YOLOv5s network, an area suppression threshold of 0.04 is the lowest value that does not reduce the TPR and still significantly decreases the FPR. In the case of the YOLOv5l network, the lowest non-zero area suppression threshold that promotes the maximum decrease in the FPR without significantly reducing the TPR is also 0.04. Finally, we note that increasing the window size implies a delay in



(a) Hybrid systems composed by the integration of fire detection networks with the temporal persistence technique as a second stage.



(b) Hybrid systems composed by the integration of fire detection networks with the area variation technique as a second stage.

Fig. 6 Impact of the hyperparameter values on true positive and false positive rates of the detection networks in the test video set

the downward trend of the TPR curve, so we chose $w_a = 30$ for both networks.

The remaining 70% of the videos were used for two purposes: (i) to evaluate the impact of post-processing performed by the temporal analysis techniques on network detections and (ii) to test the hybrid systems on real surveillance scenes. For comparison purposes, we also considered two previously proposed networks for fire classification. The first is FireNet, a manually designed network for fire and smoke classification [3]. Its architecture is composed of 14 layers, with 3 of them being convolutional layers. The second network is a MobileNet trained for fire detection [5]. Its architecture is composed of 28 layers, if width-wise and point-wise convolution is counted as separate layers. We also considered another video database for testing which contains 46 fire videos and 16 non-fire videos; this is the database used to test FireNet in its original work [3]. Performing a comparison on a different video database allows us to remove a potential bias that our methods might have for being trained in a context similar to the tested surveillance videos, while evaluating the generalization of our methods on completely different scenarios.

The predictive performance of the methods is summarized in Table 3 which shows, besides the TPR and the FPR, the accuracy, the F_1-score, the false negative rate (FNR), and true negative rate (TNR). The rate of processed frames per second (FPS), averaged over the video database, is also presented as an estimation of the computational cost added by the proposed temporal analysis techniques in the detections of the YOLOv5 networks. For comparison purposes, we also show the performance of the YOLOv5s and YOLOv5l networks without the temporal analysis post-processing stage, that is, the possible fire detection networks for the first stage of our final hybrid system.

For the test set of our video database [20], it is possible to see that the AVT as a second stage improves all the predictive performance metrics of both detection networks, with the exception of the TPR, which remains constant at high levels. The FPR of the YOLOv5s and YOLOv5l networks falls by 14.28% and 14.29%, respectively. In the case of the TPT, the FPR falls more significantly, i.e., 19.05% for YOLOv5s and 32.14% for YOLOv5l, but at the cost of decreases in TPR of 5.87% for YOLOv5s and 2.86% for YOLOv5l. It is expected that the AVT be better than the TPT as a second stage in this test set. In general, this video database contains environmental surveillance videos where wildfires are typically composed of smoke located far from the camera. In such cases, it is possible to observe the fires from their initial stages and with a

Table 3 Performance of our hybrid fire detection methods and also of FireNet and MobileNet in two different video databases

Video database	Method	Accuracy (%)	F_1-score	TPR (%)	FPR (%)	FNR (%)	TNR (%)	FPS
[20]	YOLOv5s	68.57	0.76	97.14	60.00	2.86	40.00	162.27
	YOLOv5s+TPT	71.43	0.76	91.43	48.57	8.57	51.43	**162.81**
	YOLOv5s+AVT	**72.86**	**0.78**	97.14	51.43	2.86	48.57	159.65
	YOLOv5l	60.00	0.71	**100.00**	80.00	**0.00**	20.00	59.34
	YOLOv5l+TPT	71.43	0.77	97.14	54.29	2.86	45.71	59.36
	YOLOv5l+AVT	65.71	0.75	**100.00**	68.57	**0.00**	31.43	59.44
	FireNet [3]	54.00	0.50	46.00	38.00	54.00	62.00	32.06
	MobileNet [5]	35.00	0.00	0.00	**30.00**	100.00	**70.00**	23.66
[3]	YOLOv5s	91.94	0.95	**100.00**	31.25	**0.00**	68.75	**138.90**
	YOLOv5s+TPT	**95.16**	**0.97**	97.83	**12.50**	2.17	**87.50**	137.74
	YOLOv5s+AVT	90.32	0.94	95.65	25.00	4.35	75.00	138.22
	YOLOv5l	93.55	0.96	**100.00**	25.00	**0.00**	75.00	55.85
	YOLOv5l+TPT	**95.16**	**0.97**	97.83	**12.50**	2.17	**87.50**	55.70
	YOLOv5l+AVT	93.55	0.96	97.83	18.75	2.17	81.25	55.29
	FireNet$^{(*)}$ [3]	87.10	0.92	95.65	37.50	4.35	62.50	32.91
	MobileNet [5]	93.55	0.96	95.65	**12.50**	4.35	**87.50**	27.90

The best metric values in each video database are in bold

$^{(*)}$We note that the results obtained in this case differ from those presented in the original work because the evaluation metrics are calculated differently in the present work. Details can be seen at the beginning of Sect. 4.2

notable growth trend, making the area variation technique more advantageous in these dynamic scenes.

In contrast, for the video database proposed in [3], the AVT suppresses many detections, whether true positives or false positives, which makes its predictive performance worse for both networks, especially for the YOLOv5s. However, the TPT improves the accuracy and F_1-score, and considerably decreases the FPR at the cost of a smaller reduction in the TPR of both detection networks. The fires present in these videos are static, controlled and indoors, which justifies the temporal persistence technique being better in these well-behaved scenarios. It is also interesting to highlight that none of the temporal analysis techniques considerably delay the detections of YOLOv5 networks. In the worst case shown in Table 3, there is a delay of only 3 FPS between YOLOv5s and YOLOv5s+AVT, that is, a reduction of only 1.61%.

Finally, we have all the evidence needed to define the stages chosen to compose our final hybrid system. Since YOLOv5s is at least about 2.48 times faster than YOLOv5l, as well as being the fire detection network in the hybrid systems with better predictive performance in both video databases according to accuracy and F_1-score, we chose it as the first stage of the hybrid system. As mentioned before, to select the best second stage, we must consider the environment that will be monitored. For indoor applications, we recommend the TPT as the second stage. However, we are usually interested in large-scale real monitoring with high degrees of uncertainty, which is a more challenging scenario, with higher environmental and social impacts. In such environments, with large forest areas, we recommend the AVT as the second stage.

Comparing our results with previous works, we have that YOLOv5s+AVT outperforms both fire classification networks (FireNet and MobileNet) in [20] and that YOLOv5s+TPT does the same in [3], which are exactly the types of environments that we recommended each hybrid system. However, as our first stage is a much deeper network than previously proposed in the literature, with 224 layers and about 7.2 million trainable parameters, they are expected to have lower detection speeds. Interestingly, networks of the YOLOv5 series have more optimized architectures, which gives them a higher FPS even with more layers. Furthermore, these detection networks were not hand-designed specifically for fire detection like FireNet, which avoids a time-consuming and costly engineering process of building a convolutional neural network architecture [22].

Another interesting aspect that we can highlight from Table 3 is that all methods had worse predictive performances in our test videos than in the videos collected in [3]. The best F_1-score obtained in our video test set, for example, is 19.59% lower than the best value obtained in the other test set, which shows how challenging our video

database is. Moreover, MobileNet proposed in [5] had extremely poor predictive performance in this context. Since this network was only trained to classify fire events and our videos have wildfires composed essentially by smoke, no real fires were identified by it. This further corroborates the importance of detecting smoke as a visual indicator of wildfires, especially at long distances and in bright environments.

5 Conclusions

This work proposed a robust fire detection tool based on a hybrid method composed of two sequential stages: (i) a deep CNN and (ii) a temporal analysis technique. In general, the learning-based approaches found in the literature tend to have high false positive rates, and the incorporation of a second, temporal analysis stage, reduces this rate considerably. This study presented two different techniques for the temporal analysis stage: temporal persistence (TPT) and area variation (AVT). Both techniques are based on imposing constraints on the movement of the wildfire detections.

In our previous work, of which the current work is an extension, we noticed an important trade-off between the true positive rate and the false positive rate when adding the temporal analysis stage. However, the proposed hybrid systems achieved better predictive performance and detection speed. Additionally, it reduced the false positive rate without significantly impacting either the true positive rate or the processing frame rate.Kindly check and confirm the edit made in Author contributions statement.Everything is right.

Acknowledgements Financial support for this work was provided by CEMIG-ANEEL (R &D project D0619), by the National Council for Scientific and Technological Development (CNPq, Brazil) to Adriano Chaves Lisboa (Grant 304506/2020-6), by the Foundation for Research of the State of Minas Gerais (FAPEMIG, Brazil) to Adriano Vilela Barbosa (Grant APQ-03701-16), and by the Coordination for the Improvement of Higher Education Personnel (CAPES, Brazil).

Author Contributions PVABdV: methodology; formal analysis; software; writing—original draft, visualization. RJC: writing—original draft, visualization. TMR: writing—original draft, visualization. ACL: formal analysis; writing—review and editing; validation; supervision. AVB: writing—review and editing; supervision.

Data availability The datasets that support the findings of this study are available in the GitHub repository: https://github.com/gaiasd/DFireDataset.

Declarations

Conflict of 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.

References

1. National Institute for Space Research (INPE) (1998) Wildfire Monitoring Program. https://queimadas.dgi.inpe.br/queimadas/portal-static/situacao-atual/. Accessed 29 Oct 2022
2. Yin Z, Wan B, Yuan F, Xia X, Shi J (2017) A deep normalization and convolutional neural network for image smoke detection. IEEE Access 5:18429–18438. https://doi.org/10.1109/ACCESS.2017.2747399
3. Jadon A, Omama M, Varshney A, Ansari MS, Sharma R (2019) FireNet: a specialized lightweight fire and & smoke detection model for real-time IoT applications. Preprint at https://arxiv.org/abs/1905.11922. https://doi.org/10.48550/arxiv.1905.11922
4. Toulouse T, Rossi L, Celik T, Akhloufi M (2016) Automatic fire pixel detection using image processing: a comparative analysis of rule-based and machine learning-based methods. SIViP 10(4):647–654. https://doi.org/10.1007/s11760-015-0789-x
5. Mukhopadhyay D, Iyer R, Kadam S, Koli R (2019) FPGA deployable fire detection model for real-time video surveillance systems using convolutional neural networks. In: 2019 global conference for advancement in technology (GCAT). IEEE, Bangalore, India, pp 1–7. https://doi.org/10.1109/GCAT47503.2019.8978439
6. Gaur A, Singh A, Kumar A, Kumar A, Kapoor K (2020) Video flame and smoke based fire detection algorithms: a literature review. Fire Technol 56(5):1943–1980. https://doi.org/10.1007/s10694-020-00986-y
7. Xie Y, Zhu J, Cao Y, Zhang Y, Feng D, Zhang Y, Chen M (2020) Efficient video fire detection exploiting motion-flicker-based dynamic features and deep static features. IEEE Access 8:81904–81917. https://doi.org/10.1109/ACCESS.2020.2991338
8. Nguyen MD, Vu HN, Pham DC, Choi B, Ro S (2021) Multistage real-time fire detection using convolutional neural networks and long short-term memory networks. IEEE Access 9:146667–146679. https://doi.org/10.1109/ACCESS.2021.3122346
9. Shahid M, Virtusio JJ, Wu Y-H, Chen Y-Y, Tanveer M, Muhammad K, Hua K-L (2022) Spatio-temporal self-attention network for fire detection and segmentation in video surveillance. IEEE Access 10:1259–1275. https://doi.org/10.1109/ACCESS.2021.3132787
10. Hashemzadeh M, Zademehdi A (2019) Fire detection for video surveillance applications using ICA K-medoids-based color model and efficient spatio-temporal visual features. Expert Syst Appl 130:60–78. https://doi.org/10.1016/j.eswa.2019.04.019
11. Qian Z, Xiao-jun L, Lei H (2020) Video image fire recognition based on color space and moving object detection. In: 2020 international conference on artificial intelligence and computer engineering (ICAICE). IEEE, Beijing, China, pp 367–371. https://doi.org/10.1109/ICAICE51518.2020.00077
12. Kong SG, Jin D, Li S, Kim H (2016) Fast fire flame detection in surveillance video using logistic regression and temporal smoothing. Fire Saf J 79:37–43. https://doi.org/10.1016/j.firesaf.2015.11.015
13. Çetin AE, Merci B, Gûnay O, Uğur Töreyin B, Verstockt S (2016) Methods and techniques for fire detection: signal, image and video processing perspectives. Academic Press, London, pp 1–87. https://doi.org/10.1016/C2014-0-01269-5
14. Abid F (2020) A survey of machine learning algorithms based forest fires prediction and detection systems. Fire Technol 57:559–590. https://doi.org/10.1007/s10694-020-01056-z
15. Muhammad K, Ahmad J, Baik SW (2018) Early fire detection using convolutional neural networks during surveillance for effective disaster management. Neurocomputing 288:30–42. https://doi.org/10.1016/j.neucom.2017.04.083

16. Muhammad K, Khan S, Elhoseny M, Ahmed SH, Baik SW (2019) Efficient fire detection for uncertain surveillance environment. IEEE Trans Ind Inf 15(5):3113–3122. https://doi.org/10.1109/TII.2019.2897594
17. Li P, Zhao W (2020) Image fire detection algorithms based on convolutional neural networks. Case Stud Thermal Eng 19:100625. https://doi.org/10.1016/j.csite.2020.100625
18. Majid S, Alenezi F, Masood S, Ahmad M, Gündüz ES, Polat K (2022) Attention based CNN model for fire detection and localization in real-world images. Expert Syst Appl 189:116114. https://doi.org/10.1016/j.eswa.2021.116114
19. Schmidhuber J (2015) Deep learning in neural networks: an overview. Neural Netw 61:85–117. https://doi.org/10.1016/j.neunet.2014.09.003
20. Venâncio PVAB, Rezende TM, Lisboa AC, Barbosa AV (2021) Fire detection based on a two-dimensional convolutional neural network and temporal analysis. In: 2021 IEEE Latin American conference on computational intelligence (LA-CCI). IEEE, Temuco, Chile, pp 1–6. https://doi.org/10.1109/LA-CCI48322.2021.9769824
21. Saponara S, Elhanashi A, Gagliardi A (2021) Real-time video fire/smoke detection based on CNN in antifire surveillance systems. J Real-Time Image Proc 18(3):889–900. https://doi.org/10.1007/s11554-020-01044-0
22. Venâncio PVAB, Lisboa AC, Barbosa AV (2022) An automatic fire detection system based on deep convolutional neural networks for low-power, resource-constrained devices. Neural Comput Appl. https://doi.org/10.1007/s00521-022-07467-z
23. Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, Huang Z, Karpathy A, Khosla A, Bernstein M et al (2015) ImageNet large scale visual recognition challenge. Int J Comput Vis 115(3):211–252. https://doi.org/10.1007/s11263-015-0816-y
24. Deng J, Dong W, Socher R, Li L-J, Li K, Fei-Fei L (2009) ImageNet: a large-scale hierarchical image database. In: Conference on computer vision and pattern recognition. IEEE, Miami, pp 248–255. https://doi.org/10.1109/CVPR.2009.5206848
25. Krizhevsky A, Sutskever I, Hinton GE (2017) ImageNet classification with deep convolutional neural networks. Commun ACM 60(6):84–90. https://doi.org/10.1145/3065386
26. Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A (2015) Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition, Boston, pp 1–9. https://doi.org/10.1109/CVPR.2015.7298594
27. Girshick R, Donahue J, Darrell T, Malik J (2014) Rich feature hierarchies for accurate object detection and semantic segmentation. In: 2014 IEEE conference on computer vision and pattern recognition, pp 580–587. https://doi.org/10.1109/CVPR.2014.81
28. Girshick R (2015) Fast R-CNN. In: Proceedings of the IEEE international conference on computer vision. IEEE, Santiago, Chile, pp 1440–1448. https://doi.org/10.1109/ICCV.2015.169
29. Ren S, He K, Girshick R, Sun J (2015) Faster R-CNN: towards real-time object detection with region proposal networks. In: Proceedings of the 28th international conference on neural information processing systems—volume 1. NIPS'15. MIT Press, Cambridge, MA, USA, pp 91–99
30. Redmon J, Divvala S, Girshick R, Farhadi A (2016) You only look once: unified, real-time object detection. In: IEEE conference on computer vision and pattern recognition. IEEE, Las Vegas, pp 779–788. https://doi.org/10.1109/CVPR.2016.91
31. Redmon J, Farhadi A (2017) YOLO9000: better, faster, stronger. In: 2017 IEEE conference on computer vision and pattern recognition (CVPR). IEEE, Las Vegas, USA, pp 6517–6525. https://doi.org/10.1109/CVPR.2017.690
32. Redmon J, Farhadi A (2018) YOLOv3: an incremental improvement. arXiv preprint arXiv:1804.02767. https://doi.org/10.48550/ARXIV.1804.02767
33. Bochkovskiy A, Wang C-Y, Liao H-YM (2020) YOLOv4: optimal speed and accuracy of object detection. Preprint at https://arxiv.org/abs/2004.10934. https://doi.org/10.48550/arXiv.2004.10934
34. Bochkovskiy A (2013) Darknet: open source neural networks in C. https://git.io/JTICL. Accessed 29 Dec 2021
35. Wang C-Y, Bochkovskiy A, Liao H-YM (2021) Scaled-YOLOv4: scaling cross stage partial network. In: 2021 IEEE/CVF conference on computer vision and pattern recognition (CVPR). IEEE, Nashville, USA, pp 13024–13033. https://doi.org/10.1109/CVPR46437.2021.01283
36. Jocher G, Stoken A, Chaurasia A, Borovec J, Kwon Y, Michael K et al (2021) Ultralytics/yolov5: v6.0—YOLOv5n 'Nano' models, Roboflow integration, TensorFlow export, OpenCV DNN support. Zenodo. https://doi.org/10.5281/zenodo.5563715
37. Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, Killeen T, Lin Z, Gimelshein N, Antiga L, et al (2019) Pytorch: an imperative style, high-performance deep learning library. Preprint at https://arxiv.org/abs/1912.01703
38. Gaia, Solutions on Demand (2018) D-Fire: an image data set for fire detection. https://git.io/JONna. Accessed 13 May 2022
39. CEMIG, UFMG, Gaia, RaroLabs and UFVJM (2020) Apaga o Fogo! https://apagaofogo.eco.br/. Accessed 15 May 2022
40. Celik T, Demirel H (2009) Fire detection in video sequences using a generic color model. Fire Saf J 44(2):147–158. https://doi.org/10.1016/j.firesaf.2008.05.005
41. State Key Lab of Fire Science (SKLFS) (2012) Video smoke detection. http://staff.ustc.edu.cn/~yfn/vsd.html. Accessed 11 Feb 2022
42. National Fire Research Laboratory (NFRL) (2019) National Institute of Standards and Technology (NIST). https://www.nist.gov/fire. Accessed 11 Feb 2022
43. Chakrabortya DB, Detania V, Jigneshkumar SP (2021) Fire threat detection from videos with Q-rough sets. Preprint at https://doi.org/10.48550/ARXIV.2101.08459

Authors and Affiliations

Pedro Vinícius A. B. de Venâncio[1] · Roger J. Campos[1,2] · Tamires M. Rezende[3] · Adriano C. Lisboa[3,4] · Adriano V. Barbosa[1,5]

✉ Pedro Vinícius A. B. de Venâncio
pedrovinicius@ufmg.br

Roger J. Campos
rogercampos@unifei.edu.br

Tamires M. Rezende
tamires.rezende@gaiasd.com

Adriano C. Lisboa
adriano.lisboa@gaiasd.com

Adriano V. Barbosa
adrianovilela@ufmg.br

[1] Graduate Program in Electrical Engineering, Universidade Federal de Minas Gerais, Belo Horizonte, Brazil

[2] Institute of Technological Sciences, Federal University of Itajubá, Itajubá, Brazil

[3] Gaia Solutions on Demand, Technological Park of Belo Horizonte, Belo Horizonte, Brazil

[4] Graduate Program in Computational and Mathematical Modelling, Federal Center for Technological Education of Minas Gerais, Belo Horizonte, Brazil

[5] Department of Electronics Engineering, Universidade Federal de Minas Gerais, Belo Horizonte, Brazil