IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 16 April 2024, accepted 19 May 2024, date of publication 27 May 2024, date of current version 18 June 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3406215

APPLIED RESEARCH

Wildfire Detection With Deep Learning–A Case Study for the CICLOPE Project

AFONSO M. GONÇALVES[1], TOMÁS BRANDÃO[2], AND JOÃO C. FERREIRA[2,3], (Senior Member, IEEE)

[1]Lovelytics, Arlington, VA 22201, USA
[2]Instituto Universitário de Lisboa (ISCTE-IUL), ISTAR, 1649-026 Lisbon, Portugal
[3]INOV INESC Inovação—Instituto de Novas Tecnologias, 1000-029 Lisbon, Portugal

Corresponding author: Tomás Brandão (tomas.brandao@iscte-iul.pt)

This work was supported in part by Fundação para a Ciência e a Tecnologia, I. P. (FCT) through ISTAR projects under Grant UIDB/04466/2020 and Grant UIDP/04466/2020, and by BLOCKCHAIN.PT (RE-C05-i01.01-Agendas/Alianças Mobilizadoras para a Reindustrialização, Plano de Recuperação e Resiliência de Portugal na sua componente 5-Capitalização e Inovação Empresarial e com o Regulamento do Sistema de Incentivos "Agendas para a Inovação Empresarial", approved by Ministerial Order No. 43-A/2022 of 19.01.2022).

ABSTRACT In recent years, Portugal has seen wide variability in wildfire damage associated to high unpredictability of climatic events such as severe heatwaves and drier summers. Therefore, timely and accurate detection of forest and rural wildfires is of great importance for successful fire containment and suppression efforts, as wildfires exponentially increase their spread rate from the moment of ignition. In the field of early smoke detection, the CICLOPE project currently trailblazes in the employment of a network of Remote Acquisition Towers for wildfire prevention and observation, along with a rule-based automatic smoke detection system, covering over 2, 700, 000 hectares of wildland and rural area in continental Portugal. However, the inherent challenges of automatic smoke detection raise issues of high false alarm rates that affect the system's prediction quality and overwhelm the Management and Control Centers with numerous false alarms. The research work presented in this paper evaluates the potential improvement in wildfire smoke detection accuracy and specificity using deep learning-based architectures. It proposes a solution based on a Dual-Channel CNN that can be deployed as a secondary prediction confirmation layer to further refine the CICLOPE automatic smoke detection system. The proposed solution takes advantage of the high true alarm coverage of the current detection system by taking only the predicted alarm images and respective bounding box coordinates as inputs. The Dual-Channel network combines the widely used DenseNet architecture with a novel detail selective network with spatial and channel attention modules trained separately with image data obtained from CICLOPE, fusing the extracted features from both networks in a concatenation layer. The results demonstrate that the proposed Dual-Channel CNN outperforms both single-channel networks, achieving an accuracy of 99.7% and a low false alarm rate of 0.20% when re-examining the alarms produced by the CICLOPE surveillance system.

INDEX TERMS Computer vision, convolutional neural networks, deep learning, smoke detection, wildfire detection.

I. INTRODUCTION

With the increasing variability of climate around the world, rural and forest fires pose a serious threat to public safety, with severe environmental and socio-economic effects. The Mediterranean region has observed some of the most disastrous wildfire occurrences in the last two decades, and while the total number of fires has shown a decreasing trend, the total burnt land area reflects the high unpredictability associated with extreme meteorological conditions, such as the severe heatwave experienced in Portugal that led to a catastrophic season of very large forest fires in 2017 [1], as can be observed in the plot depicted in Fig. 1.

From the moment of their ignition to their fully-developed stage, wildfires expand rapidly with an exponential increase in their spread rate [2]. Therefore, early and accurate

The associate editor coordinating the review of this manuscript and approving it for publication was Li Zhang.

© 2024 The Authors. This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License.
For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

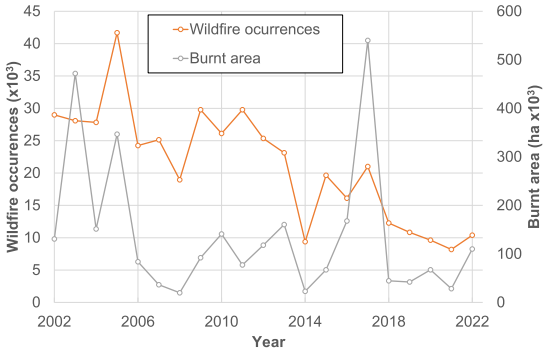


FIGURE 1. Number of wildfires and total burnt area in mainland portugal, over the last two decades.

wildfire detection, particularly during the initial smoldering stage when the first smoke columns appear, is essential for increasing the chance of success of fire containment efforts, as the time span between ignition and detection is proportional to potential damage [3].

Traditional human observation detection has inherent drawbacks, as it is human resource intensive, and becomes increasingly difficult in large-scale wildland coverage, even with the employment of watchtower surveillance imagery solutions. Automatic detection systems are therefore the optimal solution for timely smoke detection, capable of simultaneously covering extensive land areas, limited only by the optical reach and the spatial resolution of the cameras. However, the issue of accuracy and performance becomes more prominent in these systems, and concerns with wildfire coverage and false alarm rates define its applicability to real-world scenarios.

CICLOPE[1] is an integrated wildfire surveillance system with automatic detection capabilities operating in Portugal, covering over 2.700.000 hectares of wildland and rural area, as shown in Fig. 2. It is built upon a network of Remote Acquisition Towers mounted with visible and infrared wavelength cameras with continuous 360 degree pan range, 40 kilometers of effective zoom range, and a detection range of about 20 kilometers, along with autonomous power supply and weather data collection abilities. The video feeds from the camera network are processed and streamed to the Management and Control Centers for real-time observation and monitoring, while smoke alarms identified by the automatic detection system trigger visual and audio alerts for manual confirmation, constituting a valuable tool for timely first-response action. The automatic wildfire detection system operates with a rule-based algorithm that continuously analyses the video feeds in a frame-wise basis, identifying regions with a sudden increase or decrease in brightness levels. While the detection system reports very good coverage ability, correctly identifying most occurrences of true smoke alarms, the algorithm's over sensitivity tends to produce a higher rate of false alarm occurrences, resulting in a worse model specificity.

[1]https://www.inov.pt/en/project/ciclope/index.html

FIGURE 2. CICLOPE surveillance coverage in mainland portugal (green), as of december 2022.

With an average of about 34 daily wildfire occurrences in Portugal during the past five years, with many more during the warm season, and a high volume of image frames continuously collected, each subject to the detection algorithm, a high rate of false alarms results in a flood of noisy fire alarms that hide the true alarm occurrences and reduce the operators' confidence and trust in the automatic detection system. Therefore, there is a strong need to improve the specificity and overall accuracy of the system. Exploring innovative computer vision and deep learning-based solutions could prove of significant benefit to the integrated CICLOPE surveillance system.

The work developed and presented in this paper aims to answer the following research question: "Is it possible to improve the overall accuracy and reduce the false alarm rate of an automatic wildfire detection system by applying further Deep Learning-based classification methods?". Thus, it addresses a critical issue in current computer vision-based wildfire detection systems: the presence of high false alarm rates. It also emphasizes the benefit of combining rule-based and deep learning models. While designed for CICLOPE, the proposed solution's adaptability allows an easy application to other use cases.

The paper is organized as follows: after the literature review presented in Section II, Section III provides a valuable analysis on different data preparation strategies. This analysis is supported not only by the classification results but also by statistical significance tests directly comparing the possible data configurations. This may provide valuable information for future research. Section IV compares various deep learning models, highlighting their strengths and weaknesses, aiming to provide a better understanding on the models and to determine which are the most adequate for the wildfire smoke detection task. An enhanced fire detection model is then

proposed, combining a well-established model (DenseNet) with a detail-selective network based on attention models. This architecture is one of the main contributions of this paper, since it significantly lowers the false alarm rate when re-examining the alarms outputted by the rule-based CICLOPE's smoke detection algorithm. Finally, the main conclusions and directions for future work are provided in Section V.

II. LITERATURE REVIEW

A systematic review was performed following the Preferred Reporting Items for Systematic Reviews and Meta-Analysis (PRISMA) methodology [4], with a search over articles and conference papers in the Scopus database based on the following search query:

```
("smoke detect*" OR "wildfire detect*"
OR "forest fire detect*") AND
("deep learning" OR "computer vision"
OR "image classification" OR
"semantic segmentation")
AND PUBYEAR > 2009.
```

The search query returned 262 documents, with the large majority dating to the last five years, which highlights the relative novelty of deep learning studies for wildfire detection applications.

The identified references were subject to a screening process by analyzing document titles, abstracts and keywords, in order to determine their applicability to the topic of this research.

Afterwards, a full-text article analysis was conducted to assess the eligibility of the screened documents for quantitative synthesis, identifying references with relevant methods, approaches, and outcomes that were deemed useful for reviewing. In this stage, documents were excluded based on unsuitable data collection methods, proprietary software applications, or similar redundant approaches, resulting in about 20 papers used in this state-of-the-art review.

A. IMAGE CLASSIFICATION

In [5], the authors performed a review on deep learning-based methods for wildfire detection based on unmanned aerial vehicles (UAV) imagery, gathering 15 different articles. Of these, five used image classification techniques, seven used object detection approaches, and the remaining three were based on semantic segmentation. The authors concluded that smoke detecting models achieve better results than those based on flame detection, especially for early wildfire detection. However, smoke detection showed poorer performance for nighttime images and for images containing fog, clouds, or other smoke-like objects. Some researchers applied flame detection algorithms with thermal images to improve model performance, while others achieved good results with combinations of smoke and flame detecting algorithms with both optical and thermal images.

The work presented in [6] proposes a Convolutional Neural Network (CNN) model that achieves good performance in both clear and foggy environments, suggesting a multi-class approach instead of binary classification. Four classes are thus defined: smoke, non-smoke, non-smoke with fog, and fog. The authors used a VGG16 type architecture pre-trained on ImageNet, which uses smaller filter sizes and shorter strides. It outperformed other pre-trained models, namely GoogleNet and AlexNet, achieving an accuracy of 97.72%.

Another review, [7], performed a survey of recent techniques applied for computer vision-based fire and smoke detection. One of the methodologies analyzed was the use of dual-channel CNNs for image classification. This type of architecture utilizes two separate networks, with one channel focusing on extracting generalized features, and the second channel extracting detailed features.

In [8], this method was accomplished using an AlexNet network with transfer learning for the extraction of more general features, and a separately trained CNN to extract detailed features, fusing the output features of both networks in a concatenation layer. With this method, the authors combined the more comprehensive features generated from a pre-trained AlexNet architecture, with a task-specific fully trained network, and achieved an accuracy of 99.33%, outperforming AlexNet with transfer learning (99.08%).

Similarly, [9] also applies a Dual-Channel CNN, although taking a different approach, using a Selective-based Batch Normalization Network (SBNN) and a Skip Connection-based Neural Network (SCNN). The SBNN is a sequence of convolution layers with max pooling and batch normalization layers and aims at extracting detailed smoke features such as texture, while the SCNN introduces skip connection and a global average pooling layer to extract generic features, such as contour. When applying max pooling, the largest valued pixels are passed on, enhancing texture features, while average pooling has a smoothing effect, highlighting contour and shape features. The authors compared the performance of the proposed Dual-Channel CNN (DC-CNN) to various state-of-the-art architectures and each component network in terms of accuracy, detection rate, and false alarm rates over two different test sets. In both sets, the DC-CNN achieved the highest accuracy rate and lowest false alarm rates, with an accuracy of 99.7% and 99.4%, and a false alarm rate of 0.12% and 0.24%, over Sets 1 and 2, respectively. The best performing state-of-the-art networks on accuracy rates were DenseNet (98.6% and 98.4%), Xception (97.9% and 98.4%), and DNCNN (97.8% and 98.0%), while the lowest false alarm rates were achieved by DNCNN (0.48% and 0.48%), Xception (0.13% and 1.10%), and DenseNet (1.08% and 1.10%). The proposed DC-CNN also performed better than each subnetwork alone, as SBNN achieved accuracy of 98.3% and 98.7% and false alarm rates of 0.96% and 0.98%, whereas SCNN reached accuracy scores of 98.6% and 98.5% and false alarm rates of 0.84% and 0.48%. The significant improvement in performance from the proposed DC-CNN demonstrates that a larger diversification

of extracted features can produce a better generalizing model.

The aforementioned DNCNN was proposed in [10] and stands for Deep Normalization and Convolutional Neural Network. Using batch normalization, the authors replaced the traditional convolution layers in CNNs with normalization and convolutional layers. This process minimizes the effects of internal covariate shifts related to changes in the distribution of network activations during training, significantly accelerating the processing time and increasing model efficacy.

In [11], a dilation mechanism is employed in convolution layers in order to extract larger features, ignoring smaller ones, while reducing processing time and the number of parameters. Dilated convolutions apply a modified kernel by inserting gaps between the pixel elements based on a factor, where a factor of one is a regular convolution, and a factor of n expands the kernel by skipping $n - 1$ pixel elements. The author compared network performance with and without the dilation operator, having achieved an accuracy of 99.06% with the Dilated CNN and 97.53% without dilation. The authors also compared the proposed network with several state-of-the-art architectures, reporting the highest accuracy and F1 scores. However, model recall and precision scores were 97.46% and 98.27% respectively, while Inception V3 achieved a recall score of 99.80%, and VGG19 achieved a precision score of 99.49%. The authors also reported a larger error rate when classifying images in cloudy weather conditions. Processing time was also compared, with the dilated CNN reducing training time and prediction time considerably as opposed to other networks.

A Convolutional Block Attention Module (CBAM) was proposed in [12] by combining a Channel Attention Module (CAM) and a Spatial Attention Module (SAM). CAM attempts to focus on meaningful information between input channels by exploring the inter-channel relationships of the extracted features, whereas SAM focuses on the most informative spatial location of the feature maps. In [13], the authors applied a similar mechanism in the proposed SmokeNet model and applied it to smoke detection in satellite imagery, classifying between six different classes: Cloud, Dust, Haze, Land, Seaside, and Smoke. The proposed SmokeNet model outperformed several state-of-the-art architectures, reaching an accuracy score of 92.75%, with a precision score of 87.68% and a recall score of 94.68% on the smoke class.

B. OBJECT DETECTION

Object detection approaches have been widely applied in wildfire detection applications in order to identify and localize the object of interest within the picture frame. However, these algorithms are usually more computationally intensive than image classification models. Furthermore, object detection models can follow two-stage or single-stage architectures. In the case of two-stage detectors, the first stage selects regions of interest to be classified in the second stage. In contrast, single-stage architectures detect the image objects and classify them on a single pass.

In [14], the authors compared two two-stage detectors (Faster R-CNN and R-FCN) and one single-stage detector (SSD), implementing the feature extraction backbone with different CNN architectures. In the case of Faster R-CNN and R-FCN, they used Inception ResNet V2, Inception V2, ResNet V2 and MobileNet as feature extractors, while in the case of SSD only MobileNet and Inception V2 were used. The performance of the different detectors on a smoke detection dataset showed that SSD is faster to process test images but is less accurate, while Faster R-CNN is more computationally expensive but more accurate with each different feature extraction backbone. The results also show that Faster R-CNN with Inception ResNet V2 performed better, achieving a mean average precision (mAP) of 56.04%.

Another widely used single-stage detector is YOLO. In [15], YOLO-SMOKE is proposed, based on YOLOv3, which uses darknet-53 as the feature extraction backbone. The authors compared the performance of the original YOLOv3 model with the modified YOLO-SMOKE model, by introducing an efficient channel attention module (ECA), changing the loss function to focal loss in order to handle the problem of class imbalance, and introducing dropblock layers as a regularization method. The experiments on the test set showed that the proposed model improved YOLOv3 mAP from 81.95% to 86.86% without increasing image processing time.

Similarly, [16] proposes an improved framework based on YOLOv4 with CSPdarknet53 as backbone, using depthwise separable convolutions and spatial pyramid pooling. Depthwise separable convolutions significantly reduce the number of parameters by performing the convolution on each channel layer separately and afterward performing pointwise convolution with a $1 \times 1 \times n$ kernel, where n corresponds to the number of channels. Since the fully connected layer requires a fixed-size input, spatial pyramid pooling enables multi-scale input images by making the pooling operation proportional to the image size. The proposed model achieved an accuracy rate of 97.8% and a false alarm rate of 1.7%, while YOLOv4 performed at an accuracy rate of 96.7% and a false alarm rate of 3.0%.

In [17], a dynamic background modeling mechanism was applied for improving the performance of an SSD detector using a MobileNet backbone. Considering the motion characteristic of smoke objects in video sequences, the ViBe algorithm separates the dynamic foreground objects from the stationary background in the image. The proposed framework intersects the SSD detection output with the extracted moving target to improve detection accuracy. The proposed model achieved a mAP of 51.87% with $R = 3$ and $IoU = 0.03$, improving on the single application of SSD-MobileNet with a mAP of 23.81%.

Ensemble methods work by combining the outputs of various models to improve the prediction output. In [18],

an ensemble strategy is employed, merging object detection and image classification. Two detectors, YOLOv5 and EfficientDet, are trained separately to generate candidate boxes, applying a non-maximum suppression algorithm to remove redundant bounding boxes. In parallel, a classification network based on EfficientNet is applied to classify the entire image, retaining the bounding box based on the image classification output. The proposed framework was compared to a two-learner framework without the image classification branch and other object detection architectures. The two-learner model achieved the highest AP with an IoU = 0.5 of 79.7% followed by the proposed three-learner model with an AP of 79.0%, however, the false alarm rate for the two-learner model was 51.6%, whereas the proposed framework achieved 0.3%, suggesting that the ensemble approach of combining an image classification model with object detection appreciably reduces false positives while not decreasing AP significantly.

C. SEMANTIC SEGMENTATION

Semantic segmentation approaches are more computationally intensive due to the classification of each pixel within the image set. In smoke detection, it becomes particularly hard given that the smoke target is not well defined, as diffusion introduces ambiguity in the precise location of smoke. In [19] a new method is proposed to solve this problem, utilizing concentration weight labeling by incorporating a mask over the ground truth label based on the relationship to pixel values. The authors applied an encoder-decoder architecture with MobileNet as the downsampling layer, and PSPnet as the upsampling layer, with a weighted loss function and 4 smoke categories – Thick smoke, Thin smoke, Thick smoke and clouds, and Thin smoke and clouds. The results show that the weight-based network achieved a mIoU of 75.38%, as opposed to 73.86% without concentration weighting.

D. TRANSFER LEARNING

Transfer learning can be a very useful technique when implementing state-of-the-art architectures that have already been intensively trained on very large datasets. In [20], the authors compared several architectures on performance levels and training time with and without transfer learning, over a smoke recognition task. The studied networks were AlexNet, VGG16, Inception V3, ResNet50, and MobileNet, and the authors concluded that the application of transfer learning sorely improved model accuracy and training time, with the best model trained without transfer learning being AlexNet, reaching an accuracy of 98.91% after 200 epochs, while VGG16 with transfer learning reached an accuracy of 99.73% after 15 epochs.

Another work, [21], applies a pre-trained MobileNetV2 network over a smoke detection dataset and compares it to two pre-trained models, AlexNet and FireNet, as well as a fully trained standard CNN, and achieved an accuracy of 99.3% with MobileNetV2 with transfer learning, while AlexNet, FireNet, and the standard CNN, performed at accuracy scores of 95%, 97.5%, and 85.6%, respectively.

E. DATA AUGMENTATION

As also stated, data augmentation can too be beneficial, especially in the event of small and imbalanced datasets. In [22], an image manipulation technique was used through synthetically implanting smoke column objects in non-smoke images, in order to increment the number of positive samples. The authors applied a Faster R-CNN detection network, and tested a network trained on only real data samples against a network trained on synthetically augmented data, over four video sets. The detection rates improved from 98.90% to 100.00% on video 1, from 51.84% to 73.62% on video 2, from 73.62% to 98.77% on video 3, and maintained at 100.00% on video 4, suggesting that the applied data augmentation technique can improve detection ability on the same architecture.

The work in [23] presents a deep learning data augmentation approach, and trained a VGG16, ResNet50 and DenseNet networks on a smoke detection dataset, and compared performance with real training data against augmented training data. The authors applied a CycleGAN network to produce new artificial samples based on the original data and concluded that accuracy decreased for VGG16 from 93.76% to 93.28%, while for ResNet50 it increased from 96.73% to 96.93%, and DenseNet improved more expressively from 96.73% to 98.27%.

F. RULE-BASED METHODS

Many current wildfire detection applications still use rule-based image processing techniques for automatic smoke identification, reason why incorporating these along with deep learning models could configure worthwhile solutions.

Reference [24] applies CNN models over suspected regions extracted through image processing techniques. The authors applied dynamic background subtraction, based on the notion that smoke objects will tend to expand and move through different frames, and subsequently extracted the dark-channel image using the dark-channel prior method, and inputted the suspected target into a CNN. The registered performance over two test sets showed an improvement with the application of the proposed image processing techniques, increasing accuracy on test set 1 from 93.96% to 99.77%, and on test set 2 from 93.37% to 99.06%.

A similar strategy was employed in [25], with the application of Kalman filtering to extract foreground moving objects, followed by a color segmentation to extract gray shaded pixels, feeding into a fully-trained standard CNN model. Model performance was compared to an entirely rule-based algorithm, AdViSED, over the same test set, in which the results were comparable, with the proposed model reaching an accuracy score of 84.38%, as opposed to 85.00% on the AdViSED algorithm, while F1-Score was 88.37% and 87.50%, respectively.

FIGURE 3. Example of a CICLOPE surveillance camera mounted on a watchtower.

The dark-channel prior method was also applied in [26] along with the Lucas-Kanade Optical Flow method for vertical flow detection between image frames. Inception V3 was used as the CNN architecture for smoke detection on the pre-processed images. It outperformed SSD and Faster R-CNN detectors, FireNet, rule-based optical flow and dark-channel pre-processing algorithm, and Gaussian Mixture Modeling (GMM) with Inception V3. The proposed framework achieved an accuracy of 97.0% with an F1-Score of 97.0%.

In [27], a different technique was adopted, applying multichannel binary thresholding and HSV colorspace thresholding over the original images. Binary thresholding comprises defining a fixed threshold value and minimizing or maximizing each pixel value based on whether it is below or above the threshold. Multichannel binary thresholding performs this function on each color channel. HSV colorspace thresholding will apply a cutoff value over the resulting image and turn each pixel below this value equal to zero. The authors compared the performance of a depth-wise separable convolution network without image processing against rule-based image processing, based on if the resulting processed image contains pixels not equal to zero, as well as the combination of both methods. The proposed combined model outperformed both alternatives, achieving an accuracy of 93.60% on the test set, while the rule-based technique achieved 90.99%, and the single network tallied 91.76%.

III. DATASETS

Image data used in the scope of this research comprise four image sets captured by the CICLOPE cameras mounted on watchtowers, such as the one depicted on Fig. 3. The collected images represent wildfire alarms signaled by the rule-based smoke detection algorithm currently in operation, classified into true and false fire alarms. The collected image sets were as follows:

- `CasteloBranco_TP` contains 538 annotated images of true wildfire smoke alarms with associated bounding boxes encompassing the image region responsible for triggering the alarm. These bounding boxes are outputted by the current detection system.

TABLE 1. Training, validation and test split for the used image sets.

Image set	Train	Val	Test	Total
`CasteloBranco_TP`	376	107	55	538
`Leiria_FP_fields&forest`	502	143	73	718
`Leiria_FP_clouds&fog`	2262	646	323	3231
`Fires_2020_Gnd`	3152	900	452	4504
Total	6292	1796	1796	8991

- `Leiria_FP_fields&forest` contains a collection of 718 annotated images of false fire alarms and associated bounding boxes. These false fire alarms were further classified into the subtype "Fields and Forest", as the false detections were due to shadowing and lighting effects occurring over image regions showing fields and forest.
- `Leiria_FP_clouds&fog` contains 3231 annotated images of false alarms with bounding box identification, belonging to the subtype "Clouds and Fog", where smoke was incorrectly detected due to the presence of clouds or fog.
- `Fires_2020_Gnd` comprises a collection of 4504 annotated images of true alarms, without bounding box annotations regarding the smoke region. For the latter, a manual bounding box identification was performed using the application CiclopeAFDTools which enables manual annotation and produces a CSV file containing each image name and the bounding box coordinates.

Table 1 presents the split details across the four image sets used in this work. For each image set, a 70/20/10% split was applied to build the training, validation, and test sets, ensuring a proportional representation of samples coming from each original image set.

A. DATA ANALYSIS

The collected images were captured from 2018 to 2021, with the earliest image taken on 2018-10-03, and the latest on 2021-10-27. In the span of total available dates, 195 days have associated images, which corresponds to 17.4% of all possible days in the considered time period.

Fig. 4 illustrates the distribution of images across the time of day (with timestamps rounded up to the nearest hour). The figure shows a clear distinction between the distribution of true and false fire alarms (true and false positives). The latter predominantly occurs early in the day, between 08:00 AM and 10:00 AM, while the former exhibits a stronger prevalence between 11:00 AM and 05:00 PM.

B. CLASSES

As previously detailed in [6], a multi-class approach was implemented to deal with the challenge of smoke detection in foggy environments. Taking into consideration the characteristics of the available image sets, two separate datasets were created using distinct labeling strategies:

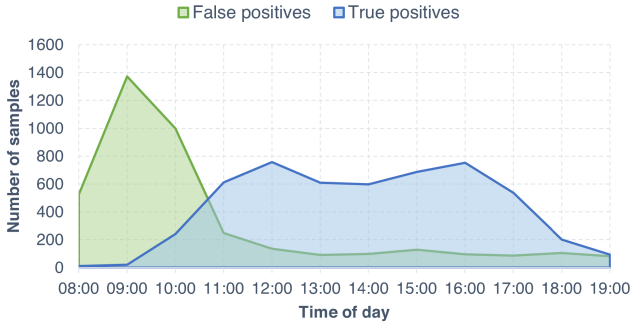


FIGURE 4. Distribution of true and false fire alarms along the daytime, for all samples in the available data.

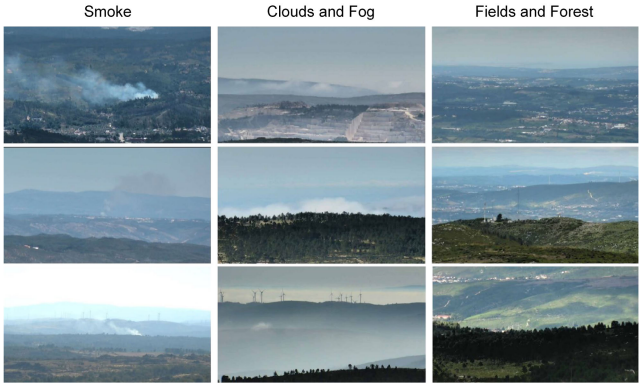


FIGURE 5. Image examples for each considered class: Smoke (left column), Clouds and fog (central column), and fields and forest (right column).

- Dataset `bin` will employ a binary classification strategy, where True Alarms represent verified wildfire smoke occurrences and will be assigned a label of 1. In contrast, false alarms congregate all other images without the verified presence of wildfire smoke, with an assigned label of 0.
- Dataset `multi` will follow a multi-class approach and classify between "Smoke", "Clouds and Fog", and "Fields and Forest".

By training and comparing the performance of the same models across both labeling strategies, an assertion can be made regarding the benefit of binary or multi-class classification as it concerns to this particular use case.

As Fig. 5 illustrates, "Smoke" class samples are characterized by a funnel-like shape, with a denser smoke base, and a diffusing smoke column that typically propagates diagonally in accordance with wind direction. Smoke columns will display different characteristics depending on the landscape of the background, with lighter coloration on dark terrain backgrounds, whereas on light above-horizon background smoke can appear darker in color.

In the case of "Clouds and Fog", these occurrences represent the majority of false alarms collected, as the passing of these objects more frequently triggers the current detection system. It is often hard to distinguish from true smoke objects as they share similar characteristics in texture and color. However, these objects can exhibit larger variation in shape and size, where smoke displays a columnar form more consistently.

"Fields and Forest" represent a small portion of false alarms collected, and gather samples which neither contain smoke, or clouds and fog, where detected objects do not include the haze and shape characteristics of one or the other.

C. BOUNDING BOXES

Many approaches to the problem of wildfire detection have adopted object detection strategies, where two separate operations take place – the first establishes a suspected region as a bounding box of the original image, and the second classifies the extracted suspect region. Other approaches, such as the ones reflected in [24], [25], [26], and [27], implement an initial rule-based image processing strategy to extract foreground or otherwise define a suspected smoke region. Such implementations can benefit from reducing noise in the original image by eliminating features and background objects that are irrelevant to the target label.

A comparison can be drawn from the aforementioned methods to the use case of this research work, where the images collected are gathered from a rule-based image processing algorithm that produces bounding box (BB) coordinates, enabling the extraction of the suspected region.

However, as previously stated, images collected from the `Fires_2020_Gnd` dataset do not contain associated bounding box coordinates identification and were thus manually classified, resulting in rather distinct BB sizes as the current rule-based system produces very small-sized BBs. This can be verified as the average BB area extracted from the detection system contains about 1930 pixels, while the manually annotated BBs contained an average of about 43095 pixels per BB.

In order to obtain similar sized BBs for both cases, the original bounding boxes were enlarged by p pixels, where $p = 5$ for the manually annotated BBs case, and $p = 150$ for the detection system's outputted BBs. Assuming that (x_1,y_1) and (x_2,y_2) are the original BB upper-left and lower-right corners, respectively, the enlarged BB coordinates (x'_1,y'_1) and (x'_2,y'_2) can be computed using the following rules:

$$\begin{aligned} x'_1 &= \max(x_1 - p, 0); \\ x'_2 &= \min(x_2 + p, W); \\ y'_1 &= \max(y_1 - p, 0); \\ y'_2 &= \min(y_y + p, H), \end{aligned} \tag{1}$$

where W and H represent the images' width and height, respectively.

Two additional datasets `bin-bbox` and `multi-bbox` were created, where the former compiles the extracted bounding box images labeled in the binary strategy that corresponds to the full-image dataset two-classes, whereas the latter gathers the extracted bounding box images labeled in the multi-class approach used in three-classes.

Comparing model performance across datasets allows for an evaluation on the best pre-processing strategy, by assessing the noise reduction advantages of bounding box images in contrast to a potential gain in contextual information that the full images may provide.

D. AUGMENTED DATA

In situations where the different classes within a dataset are represented disproportionately, we may encounter difficulties associated with class imbalance, such as poor performance on the minority class. Due to the dominance of a majority class in the dataset, if a model predicts the dominant class there is a greater chance that prediction might be correct, therefore the model may conform to a bias towards the majority class, leading to a higher probability of misclassification of the minority class.

In the case of the binary labeled datasets `bin` and `bin-bbox`, the imbalance is not significant as True Alarms represent 56.1% of the training set, and false alarms make up the remaining 43.9%. However, in the case of the multi-class labeled datasets `multi` and `multi-bbox`, the "Smoke" class represents 56.1%, "Clouds and Fog" represents 35.9%, and "Fields and Forest" only 8.0%, configuring a more severe case of class imbalance.

Typically, the two most widely used techniques to handle class imbalance are undersampling and oversampling. In undersampling, the size of the majority class is reduced by extracting a randomized sample of the total original set, whereas oversampling can include randomly duplicating records in the minority class to increase its relative size.

Similarly, data augmentation can be leveraged to artificially increase the number of samples within the minority class. Considering the nature of our dataset, image manipulations were performed with horizontal flipping operations on each "Fields and Forest" image, creating a duplicate mirrored version of each. Other operations, such as vertical flipping, or rotations, were not applied as they may disturb the natural orientation of the original set, where the top and bottom of each picture show the sky and ground, respectively.

The horizontal flip operation can be computed as:

$$F_I(x, y) = I(W - x - 1, y), \tag{2}$$

where I is the input image, F_I is the horizontally flipped image, W is the image's width, and (x, y) are the pixel coordinates.

The flipping operation was applied to the training set only, to maintain the original distribution throughout the validation and test sets and each of the previously generated datasets were replicated to evaluate the impact of data augmentation applied on the minority class Fields and Forest as it pertains to model performance across all classes.

The augmented datasets leveled class imbalance where in the case of binary datasets the distribution changed to 51.9% for true alarms and 48.1% for false alarms, while in the case of multi-class datasets, "Smoke" class represents 51.9%, "Clouds and Fog" represents 33.2%, and "Fields and Forest" increased to 14.9%.

The distribution of the eight dataset configurations created for model application are presented in Table 2.

TABLE 2. Class distribution for the different dataset configurations.

Dataset Configuration	Class	Train	Val	Test	**Total**
`bin`	True Alarm	3528	1007	507	**5042**
	False Alarm	2764	789	396	**3949**
`bin-aug`	True Alarm	3528	1007	507	**5042**
	False Alarm	3266	789	396	**4451**
`bin-bbox`	True Alarm	3528	1007	507	**5042**
	False Alarm	2764	789	396	**3949**
`bin-bbox-aug`	True Alarm	3528	1007	507	**5042**
	False Alarm	3266	789	396	**4451**
`multi`	Smoke	3528	1007	507	**5042**
	Clouds and Fog	2262	646	323	**3231**
	Fields and Forest	502	143	73	**718**
`multi-aug`	Smoke	3528	1007	507	**5042**
	Clouds and Fog	2262	646	323	**3231**
	Fields and Forest	1004	143	73	**1220**
`multi-bbox`	Smoke	3528	1007	507	**5042**
	Clouds and Fog	2262	646	323	**3231**
	Fields and Forest	502	143	73	**718**
`multi-bbox-aug`	Smoke	3528	1007	507	**5042**
	Clouds and Fog	2262	646	323	**3231**
	Fields and Forest	1004	143	73	**4504**

IV. FIRE DETECTION FRAMEWORK

A. INITIAL TRANSFER LEARNING APPROACH

At the early development stages of machine learning applications, starting off with simple approaches that can quickly return results can be beneficial, as it enables a fast output that can be examined in order to guide the workflow, rather than investing too much time in a detailed approach that may lead to less conclusive results [28].

In this section, a set of state-of-the-art models are used as an initial approach to the problem of wildfire detection observed in this research. The goal of this initial framework is to analyze the results obtained from the various models implemented regarding the defined datasets, to understand the differences of each implementation and their impact on the problem, and ultimately to select the most promising dataset and best performing model. The models were compiled using the Keras library with TensorFlow as the backend, using Python 3.7.13 on Google Colab Pro running on High-RAM Google Compute Engine with TPU backend.

Algorithm 1 transcribes the pipeline followed to pre-process each dataset, compile and train each model, return predictions, and output evaluation metrics, where D represents each dataset directory, M identifies each selected model, and C defines the classification type as either binary or multi-class. In the sequential processes, α, β, and γ stand for the training, validation, and test sets, respectively, μ and ν represent the compiled model and the trained model, while π represents the predicted classes returned.

The final classification report method will output a list of evaluation metrics that will be referenced for interpretation: Accuracy rate (Acc), Precision (Prec), Recall (Rec),

F1-Score (F1) and False Alarm Rate (FAR). These evaluation metrics can be computed using (3) to (7), where True Positives (TP) are predicted fire alarms that correspond to real fires, False Positives (FP) are predicted fire alarms that are actually false alarms, True Negatives (TN) are correctly predicted false fire alarms, and False Negatives (FN) are true fire alarms incorrectly classified as false alarms.

$$\text{Acc} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}; \tag{3}$$

$$\text{Prec} = \frac{\text{TP}}{\text{TP} + \text{FP}}; \tag{4}$$

$$\text{Rec} = \frac{\text{TP}}{\text{TP} + \text{FN}}; \tag{5}$$

$$\text{F1} = \frac{2 \cdot \text{Prec} \cdot \text{Rec}}{\text{Prec} + \text{Rec}}; \tag{6}$$

$$\text{FAR} = \frac{\text{FP}}{\text{TP} + \text{FP}}. \tag{7}$$

Algorithm 1 Pre-Processing, Training and Testing Models

Input: D, M, C
$[\alpha, \beta, \gamma]$ = Pre-process Images (D + [train, val, test], target size = (224, 224), rescale ratio = 1/255, batch size = 128, C)
μ = Compile Model (M, initial weights = 'imagenet', optimizer = Adam, C)
if *C is binary* **then**
    classification nodes = 1
    activation = sigmoid
    loss function = binary cross entropy
else
    classification nodes = 3
    activation = softmax
    loss function = categorical cross entropy
ν = Fit Model(μ, train data = α, val. data = β, epochs = 10)
π = Predict Classes (ν, test data = γ)
evalMetrics = Classification Report (π, γ)

1) DATA SELECTION

Table 3 presents a comparison of each model-dataset pairing in terms of accuracy and false alarm rates. The best performing model for each dataset configuration is highlighted in bold, while the highest accuracy scores for each model are underlined, in order to emphasize the outcomes of the different data preparation strategies.

The binary bounding box strategies, with and without data augmentation, produced substantial better results, where Xception and DenseNet returned their highest accuracy rates on `bin-bbox`, and VGG16 having its best result with `bin-bbox-aug`, whereas MobileNetV2 obtained the same accuracy rate for both, but a lower False Alarm Rate on the augmented dataset. In terms of model evaluation, DenseNet achieved the highest accuracy scores with all dataset strategies, while VGG16 consistently produced the worst scores.

TABLE 3. Performance comparison: Accuracy (Acc) and False Alarm Rate (FAR) for each dataset configuration and tested CNN architecture (values in percentage).

Dataset	Metric	VGG16	Xception	MobileNetV2	DenseNet
`bin`	Acc	94.5	97.1	97.4	**98.1**
	FAR	7.45	2.75	2.17	**1.20**
`bin-aug`	Acc	94.8	97.2	97.4	**97.9**
	FAR	5.25	2.18	1.98	**1.00**
`bin-bbox`	Acc	97.0	99.2	99.2	**99.3**
	FAR	1.81	0.98	**0.40**	0.79
`bin-bbox-aug`	Acc	97.9	98.8	**99.2**	**99.2**
	FAR	2.34	0.80	**0.40**	**0.40**
`multi`	Acc	93.1	95.2	95.6	**96.2**
	FAR	6.21	3.50	3.31	**1.20**
`multi-aug`	Acc	93.9	95.0	95.7	**96.2**
	FAR	4.41	2.75	2.18	**1.39**
`multi-bbox`	Acc	93.0	96.2	97.6	**97.8**
	FAR	2.53	0.79	**0.20**	0.78
`multi-bbox-aug`	Acc	90.9	96.2	97.6	**97.8**
	FAR	7.52	0.99	**0.79**	1.36

Table 4 further analyses dataset configuration strategies by evaluating the statistical significance of the results achieved for each option. The Pearson correlation (R-Score) resulting from an hypothetical improvement in model accuracy using the multi-class strategy instead of the binary approach shows a strong negative correlation with a value of -0.736. This negative value, combined with a very low P-Value of less than 10^{-5}, means that the multi-class strategy leads to statistically significant worse results for the fire detection task when compared with the binary classes strategy. These results can be justified on the fact that multi-class models are forced to learn additional features that are not indicative of the presence of smoke, reducing their ability to discern the false alarms when compared with binary classification models.

On the other hand, using the extracted bounding boxes for classification has shown to be a better method when compared with using the entire image. Furthermore, removing contextual noise from multiple objects that can be present in the wide landscape images, which in some cases include the presence of fog and clouds in true alarm images, significantly improves model accuracy, with a R-Score of 0.589 and a P-Value of 0.000391 (< 0.05).

As for data augmentation, its use is less conclusive as some models improved on these augmented datasets, while others had worse or comparable performances. The statistical significance tests led to a low absolute value of the Pearson correlation coefficient (R-Score $= -0.064$), and a P-Value of 0.732, which means that the test is not statistically significant ($P > 0.05$).

Considering this analysis for the experimented dataset configuration strategies, further experiments have been performed using the `bin-bbox` dataset configuration, as it showed to have the most potential for achieving better model performance.

2) MODEL SELECTION

Examining the performance metrics of the various models applied over dataset `bin-bbox`, displayed in Table 5, it is

TABLE 4. Statistical significance of accuracy improvement between the dataset configuration strategies used (negative *R*-Scores mean that the configuration strategy on the right side led to better results).

Configuration strategy	*R*-Score	*P*-Value	Significant ($P < 0.05$?)
Multiclass *vs.* Binary	-0.736	$< 10^{-5}$	Yes
Bounding Box region *vs.* Full Image	0.589	0.000391	Yes
Augmented Data *vs.* No augmentation	-0.064	0.732	No

TABLE 5. Classification assessment metrics for the tested models.

Model	Acc (%)	Prec (%)	Rec (%)	F1 (%)	FAR (%)	AUC
VGG16	97.0	98.2	96.5	97.3	1.81	.9962
Xception	99.2	99.0	**99.6**	99.3	0.98	.9991
MobileNetV2	99.2	**99.6**	99.0	99.3	**0.40**	.9993
DenseNet	**99.3**	99.2	**99.6**	**99.4**	0.79	**.9999**

apparent that VGG16 presents a significant performance gap in relation to other models. In terms of accuracy rates, Xception, MobileNetV2, and DenseNet returned comparable results, with the highest score being attributed to DenseNet at 99.3%, which slightly edges over the former two. MobileNetV2 achieved the lowest False Alarm Rate (0.40%), which is directly related to a higher Precision score of 99.6%. On the contrary, both Xception and DenseNet achieve a lower Precision score, but the highest Recall of 99.6%. This duality presents the balance between prediction quality, which pertains to Precision, and sensitivity to the target label, which is highlighted by Recall. F1-Score present a single-value metric to combine both perceptions, and DenseNet achieves the highest score of 99.4%, signifying a better decision balance.

When applying each model to the test set, the produced outputs reflect the prediction probability from the sigmoid activation function, where a value close to 1 signifies a higher probability of True Alarm, and a value close to 0 indicates a low probability of True Alarm. To achieve a categorical classification on the prediction probabilities, a cut-off value is employed within a decision function. The mentioned metrics were calculated over classifications generated using a cut-off value of 0.5, meaning prediction probabilities above or equal to 0.5 were classified as True Alarms, and those below 0.5 were classified as false alarms. The more deterministic the models are, the more spread out the prediction probabilities will be, where values will be very close to 1 or very close to 0, indicating a high level of discrimination between classes, whereas if values are closer to the cut-off value, the models have decreased discrimination capacity.

We can obtain a good indication of this property with the Receiver Operating Characteristic (ROC) curve. The ROC curve computes the ratio of True Positives Rate (TPR) over the False Positives Rate (FPR), across different decision threshold values. This relation can be better summarized in a single-value metric using the Area Under the Curve (AUC), given by

$$\text{AUC} = \int_0^1 \text{TPR}\left(\text{FPR}^{-1}(x)\right) dx. \tag{8}$$

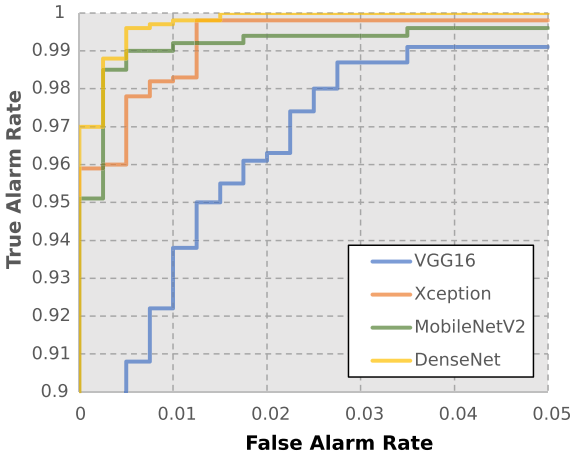


FIGURE 6. ROC curves corner detail for the tested models.

An AUC score of 1 would indicate the model can perfectly distinguish between classes, where all True Alarms have a prediction probability of 1.0, and false alarms have a prediction probability of 0.0, meaning that whatever the cut-off value, TPR is always 100%, and FPR is always 0%. An AUC close to 1 reveals a very good ability to distinguish between classes, being a very important metric to evaluate.

The detail for the ROC curves displayed in Fig. 6 show that Xception, MobileNetV2, and DenseNet all have a very high degree of discriminative ability between classes, as these can achieve high TPR values without compromising the FPR. In addition, the previously identified higher Recall scores for Xception and DenseNet in particular, match the observed curves, as both models can surpass 0.995 TPR while keeping the FPR below 0.05.

Analysis of evaluation metrics and subsequent selection of the best model is subjective to the use case and the specific needs for the problem. In the case of wildfire detection, it can be argued that the importance of the target label requires a higher degree of conservatism in selecting a model that can achieve high recall levels, in order to prioritize the identification of true alarms, and compromising on slightly lower Precision and higher number of false alarms. in this sense, DenseNet can be considered the most qualified model as it consistently showed better balance between these stances, having outputted the highest accuracy, recall and F1 scores, as well as displaying very good class discrimination as evidenced by the AUC score.

B. SCAM-SCNN

While the advantages of transfer learning have been previously explored, a key aspect of this method is the implementation of tendentiously generic pre-trained filter kernels that identify a broad range of common visual features, resulting in models that generalize well when applied to diverse image sets in production. On the contrary, training models from scratch implies the training of all network parameters without a pre-trained default base, producing

filters that identify more specific features of the target label used during the training process. A potential benefit of this strategy is the use of simpler lightweight networks which can often be more suitable than complex architectures, as the problem scope is reduced.

In this section, a selective CNN architecture is presented, implementing spatial and channel attention modules, trained exclusively over `bin-bbox`. The goal of this network is to capture more selective features tailored to the target label, so to identify informative feature maps of wildfire smoke objects and improve upon the previously trained DenseNet model by enriching the generic feature extractors with additional selective feature maps.

1) NETWORK ARCHITECTURE

The Spatial and Channel Attention Modularized Selective CNN (SCAM-SCNN) was inspired by the architecture of SBNN [9], and the Convolutional Block Attention Module (CBAM) proposed in [12]. The network architecture consists of 4 blocks of 2 convolutional layers followed by a Channel Attention Module (CAM), a Spatial Attention Module (SAM), and a max-pooling layer, where in the final block the pooling layer is replaced with a batch normalization layer, followed by the final output layer, as depicted in Table 6.

The convolution operation is a widely used image transformation process, where a filter kernel is passed through an input image, also denoted as the input tensor, and consists of the matrix multiplication of the kernel with sub-regions of the input matrix of the same size, generating a new output feature map. This process can be generally defined as:

$$\mathbf{B}_{m,n} = (\mathbf{A} * \mathbf{k}) = \sum_{i=0}^{W_k-1} \sum_{j=0}^{H_k-1} A_{m-i,n-j} \times k_{i,j} \tag{9}$$

where $\mathbf{A}$ and $\mathbf{B}$ represent the input and output matrices, respectively, and $\mathbf{k}$ is the filter kernel. In order to obtain output matrices with same size as the input ones, SCAM-SCNN uses padding on each convolutional layer.

Another feature of SCAM-SCNN is the use of strides in the first convolutional layer after a max-pooling layer. With a stride of 2×2 the filter kernel shifts 2 pixels as it passes through the input tensor, resulting in the same dimensionality reduction as pooling layers. While pooling is a fixed operation, introducing longer strides in the convolutional layer can be seen as learning the pooling operation [29]. As the outputs of the final layer before the classification layer are intended to be concatenated with the last feature maps of DenseNet, with dimensions of 7×7, these need to match in size. Strided convolutions revealed better results in achieving this downsampling goal while keeping the network architecture compact. With this change, the dimensions of output tensor can be defined as:

$$n_B = \frac{n_A - n_k + 2p}{s - 1}, \tag{10}$$

where s represents the size of the stride. To improve the training process and accelerate convergence, a batch normalization layer is introduced before the classification layer, providing a regularization effect, and reducing internal covariate shift [30]. This normalization step is defined as:

$$\hat{x} = S_f \cdot \frac{x - E(x)}{\sqrt{\text{var}(x) + \epsilon}} + O_f, \tag{11}$$

where $\hat{x}$ represents the new value of a single component, $E(x)$ is its mean within a batch, $\text{var}(x)$ is its variance within a batch, S_f is a learned scaling factor, ϵ is a small constant, and O_f is a learned offset factor.

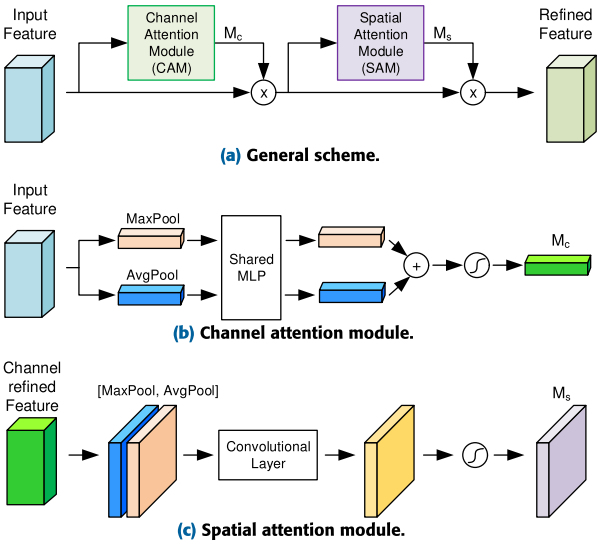


FIGURE 7. CBAM architecture.

2) SPATIAL AND CHANNEL ATTENTION MODULES

Attention mechanisms have been increasingly studied to improve the performance of CNNs, attempting to approximate the role of attention in human perception. For example, in the case of identifying wildfire smoke objects, this can be noted where humans might pay attention to certain colors and locations in the picture frame that are most informative, focusing on the features within channels and regions to make the decision.

The Spatial and Channel Attention Modules utilized in SCAM-SCNN apply the design from the CBAM proposed in [12], composed of CAM and SAM sequentially, as shown in Fig. 7.

CAM aims at extracting the most informative feature maps from an input F, denoted as channels, and works by compressing the spatial dimension into two vectors F^c_{max} and F^c_{avg} of dimensionality $f \times 1 \times 1$ using max-pooling and global average-pooling. These vectors are then passed to a shared multi-layer perceptron (MLP) with 3 layers, where the number of neurons in the input and output layers is defined by the number of channels f, while in the hidden layer these are set by a parameter ratio as $\lfloor \frac{f}{ratio} \rfloor$, where in the case of SCAM-SCNN $ratio = 8$. The resulting outputs are summed and fed through a sigmoid function that will generate a final $f \times 1 \times 1$ channel attention mapping vector with values between

TABLE 6. Structure and layer's parameters of SCAM-SCNN.

Layer	Type	Parameters
L1	Convolution	Filter size: 3x3 Filter number: 32 Stride: 1x1 Padding: Same Activation function: ReLU
L2	Convolution	Filter size: 3x3 Filter number: 64 Stride: 1x1 Padding: Same Activation function: ReLU
L3	Channel Attention	Neurons number: 84 - 8 - 64 Activation function: Sigmoid
L4	Spatial Attention	Filter number: 1 Filter size: 7x7 Activation function: Sigmoid
L5	Pooling	Pooling region size: 3x3 Stride: 2x2 Padding: Same Pooling method: Max-pooling
L6	Convolution	Filter size: 3x3 Filter number: 128 Stride: 2x2 Padding: Same Activation function: ReLU
L7	Convolution	Filter size. 3x3 Filter number: 128 Stride: 1x1 Padding: Same Activation function: ReLU
L4	Channel Attention	Neurons number: 128 - 16 – 128 Activation function: Sigmoid
L5	Spatial Attention	Filter number: 1 Filter size: 7x7 Activation function: Sigmoid
L6	Pooling	Pooling region size: 2x2 Stride: 2x2 Padding: Same Pooling method: Max-pooling
L7	Convolution	Filter size: 3x3 Filter number: 256 Stride: 2x2 Padding: Same Activation function: ReLU
L8	Convolution	Filter size: 3x3 Filter number: 256 Stride: 1x1 Padding: Same Activation function: ReLU
L9	Channel Attention	Neurons number: 256 - 32 - 256 Activation function: Sigmoid
L10	Spatial Attention	Filter number: 1 Filter size: 7x7 Activation function: Sigmoid
L11	Pooling	Pooling region size: 2x2 Stride: 2x2 Padding: Same Pooling method: Max-pooling
L12	Convolution	Filter size: 3x3 Filter number: 384 Stride: 1x1 Padding: Same Activation function: ReLU
L13	Convolution	Filter size: 3x3 Filter number: 384 Stride: 1x1 Padding: Same Activation function: ReLU
L14	Channel Attention	Neurons number: 384 - 48 - 384 Activation function: Sigmoid
L15	Spatial Attention	Filter number: 1 Filter size: 7x7 Activation function: Sigmoid
L16	Normalization	Normalization type: Batch-normalization
L17	Output	Neurons number: 1 Activation function: Sigmoid

0 and 1, that is subsequently multiplied over F, generating a refined feature block where the most informative channels are highlighted. CAM is characterized in Fig. 7b, and can be described as:

$$M_C = \mathcal{S}(\text{MLP}(\text{MaxPool}(F)) + \text{MLP}(\text{AvgPool}(F))), \quad (12)$$

where $\mathcal{S}(\cdot)$ represents the sigmoid function.

For the case of SAM, a similar but opposite operation is performed, where the channel dimension of the input F is compressed into two feature maps F^s_{max} and F^s_{avg} of dimensionality $1 \times h_F \times w_F$ using max-pooling and average-pooling, respectively, where h_F and w_F represent the height and width of F. The resulting feature maps are concatenated and forwarded through a convolutional layer with a 7×7 filter, using sigmoid as the activation function, generating a final $1 \times h_F \times w_F$ feature map with values between 0 and 1, which is then multiplied over input F, similarly highlighting the most informative regions of the feature block. This module is characterized in Fig. 7c, and can be defined as:

$$M_S = \mathcal{S}\left(\text{Conv}_{7\times7}\left([\text{MaxPool}(F); \text{AvgPool}(F)]\right)\right), \quad (13)$$

where $\text{Conv}_{7\times7}(\cdot)$ represents the outcome of the convolutional layer.

3) PERFORMANCE ASSESSMENT

In this section the performance of SCAM-SCNN is analyzed with the results obtained from training and testing on dataset `bin-bbox`, evaluating the impact of the application of spatial and channel attention modules.

The training and testing procedures were similar to those presented during the initial transfer learning approach described in sec. IV-A2, applying the pipeline presented in algorithm 1 to pre-process, train and test each model.

Table 7 represents the evaluation metrics outputted from the classification report of SCNN and SCAM-SCNN, displaying a noticeable improvement in model performance

TABLE 7. SCNN *vs.* SCAM-SCNN performance.

Model	Acc (%)	Prec (%)	Rec (%)	F1 (%)	FAR (%)	AUC
SCNN	98.4	98.4	**98.8**	98.6	1.57	.9951
SCAM-SCNN	**98.9**	**99.6**	98.4	**99.0**	**0.40**	**.9993**

to SCNN when employing spatial and channel attention modules. While Recall is slightly decreased from 98.8% to 98.4%, accuracy rate improved from 98.4% to 98.9%, Precision improved expressively from 98.4% to 99.6%, F1-Score increased from 98.6% to 99.0%, while False Alarm Rate decreased from 1.57% to 0.40%. The improvement of the AUC score from .9951 to .9993 also shows an increased discriminative ability when using SCAM layers, indicating that the effects of spatial and channel activations add to the model's ability to make decisions using the most informative spatial and channel features, resulting in a better performing model.

In order to visualize network activations, GradCAM (Gradient-weighted Class Activation Mapping) [31] is employed as a visualization tool, which provides a visual explanation to model decision, highlighting the importance of spatial locations as it pertains to target label detection. Fig. 8 shows the outputs of the different activation mappings obtained using GradCAM where the mappings obtained from SCAM-SCNN visibly display better target coverage when compared to the base SCNN model, where the latter reveals a higher importance over the edge regions of smoke columns, while for SCAM-SCNN the attention modules improve the highlighting of informative features, displaying a higher spatial importance across the entire smoke column object.

Overall, the application of spatial and channel attention modules positively affects model performance and discriminative ability, as experiments revealed improved detection ability whilst reducing the number of false alarms.

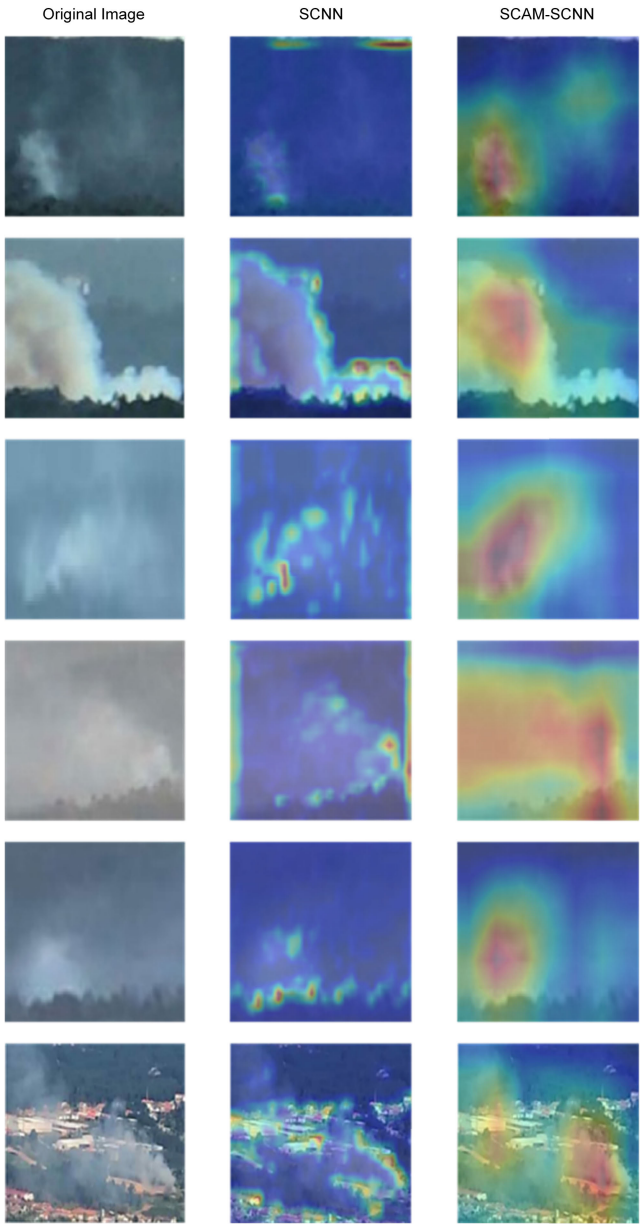


FIGURE 8. Comparison of SCNN *vs.* SCAM-SCNN network activations: original images (left column), SCNN activations (central column), and SCAM-SCNN activations (right column).

Additionally, visual explainers show a more robust feature detection ability, which demonstrates that the proposed implementation of SCAM layers is an effective mechanism in the scope of this use case of wildfire smoke detection.

Taking into consideration the observed results, the following section will present the implementation of dual-channel networks by combining SCAM-SCNN with the previously trained DenseNet model. As SCAM-SCNN is a novel architecture trained from scratch, the expectation is that the features extracted in the convolution layers of the network will reveal selective characteristics optimized for the task of wildfire smoke detection portrayed by the

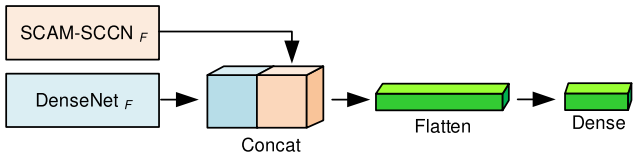


FIGURE 9. Simplified structure of the dual-channel CNN.

image set utilized during training. Through concatenating the resulting feature maps of each model, an attempt is made at enhancing DenseNet by introducing feature diversification, combining selective and generic features, increasing the information passed to the classification layer, aiming to improve performance.

C. PROPOSED DUAL-CHANNEL CNN

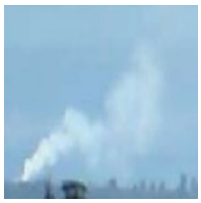
The proposed Dual-Channel CNN combines the previously described DenseNet and SCAM-SCNN models as branches of a common network, fusing the outputs of the last layer of each network before the classification layer, where DenseNet$_F$ and SCAM-SCNN$_F$ represent the feature extraction parts for each network (i.e., the original CNN models without the last fully-connected layers). The concatenated features are then subject to a new classification layer as represented in Fig. 9.

As each branch network was previously trained independently, the generated feature maps contain all the information used by each model alone for the identification of the target label where both models revealed satisfactory performance. Training both models simultaneously within the dual-channel architecture would lead to complimentary feature extractions and diminish the benefit of the diversification introduced with the combination of features extracted from individually trained models.

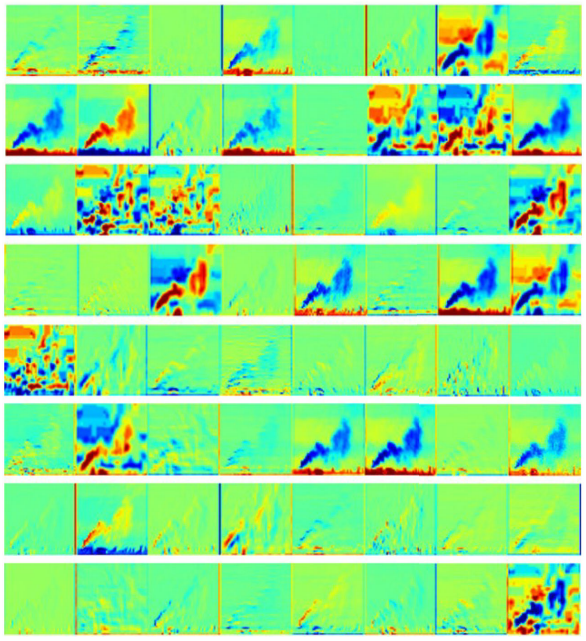
As previously detailed, DenseNet with transfer learning extracts more comprehensive generic features, while SCAM-SCNN focuses on selective detailed features of wildfire smoke. This diversity of features can be visually interpreted by observing the outputs of the first convolution layer of each model and comparing the feature map activations. An example is depicted in Fig. 10.

As the outputs of the first convolutional layer still maintain a noticeable resemblance to the original input image shown in Fig. 10a, a comparison between each convoluted image is easily traced back to its original features. The more wide-ranging and varied features of DenseNet are observable as the resulting feature maps highlight different spatial elements, identifying distinct features of the same input image, while SCAM-SCNN more consistently displays features that explicitly target the smoke column object, thus being perceptible how each model is behaving differently and employing opposing feature extracting strategies.

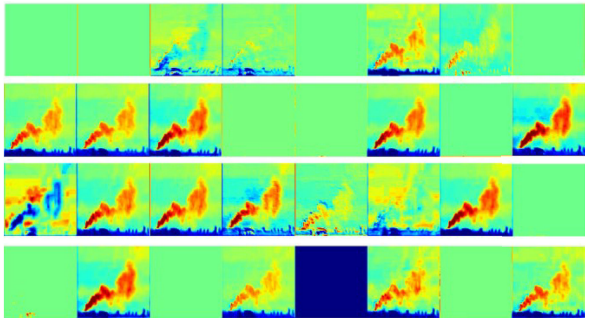
This visualization illustrates the back works of each model, and clearly portrays the different features obtained from each model, and how combining them can enrich the



(a) Example image.



(b) DenseNet.



(c) SCAM-SCNN

FIGURE 10. Feature map visualization of an example image (a) for the first convolutional layer of DenseNet (b) and SCAM-SCNN (c).

information base used in the classification layer to produce better predictions.

1) PERFORMANCE ASSESSMENT

The performance of the Dual-Channel CNN was analyzed and compared to that of its branch networks, using the `bin-bbox` dataset configuration for training the classification layer portion of the network, evaluating the results obtained and effectiveness of the dual-channel strategy.

Table 9 displays the performance metrics attained from the classification report and demonstrate the superiority of the proposed Dual-Channel CNN model, having achieved significant improvements from the branch models. By combining DenseNet with SCAM-SCNN, the dual-channel model improved the highest accuracy rate of DenseNet from 99.3% to 99.7%, while having a substantial decrease in FAR to only 0.20%. Precision, recall, and F1-scores also improved from both branch models to 99.8%, 99.6%, and 99.7% respectively, while AUC matches the highest score of 0.9999 obtained with DenseNet.

TABLE 8. Performance comparison between the proposed dual-channel CNN and each of its CNN components.

Model	Acc (%)	Prec (%)	Rec (%)	F1 (%)	FAR (%)	AUC
DenseNet	99.3	99.2	**99.6**	99.4	1.01	**.9999**
SCAM-SCNN	98.9	99.6	98.4	99.0	0.40	.9993
Proposed DC-CNN	**99.7**	**99.8**	**99.6**	**99.7**	**0.20**	**.9999**

While DenseNet showed particularly good coverage of true alarm samples, with a Recall score of 99.6%, it also showed a higher False Alarm Rate of 1.01%. SCAM-SCNN on the contrary revealed lower sensitivity, with a Recall score of 98.4%, but greater specificity and prediction quality, with a Precision score of 99.6%. The proposed Dual-Channel CNN not only achieves a good compromise between sensitivity and specificity, but also retains or improves the score for each individual metric, indicating that the combination of the two models increases overall robustness and reliability for true alarm recall and false alarm rates.

The achieved results clearly show a strong benefit in employing a dual-channel strategy, particularly when combining a robust transfer learning-based model, such as DenseNet, with an effective detail selective model such as SCAM-SCNN, as the combination of features improves upon each singular branch model by harnessing the advantages of both strategies, and further bolstering generalization and detection abilities.

2) TIME-OF-DAY-BASED DECISION ADJUSTMENT

As previously evidenced in Fig. 4, false alarms occur within the early morning hours of the day with higher frequency, predominantly between 08:00 and 11:00. In contrast, true alarms are much more common between 10:00 and 17:00. As stated in sec. III-A, this can be explained with the understanding of the two types of false alarms reported having a strong association to morning hour climatic events, when fogs and lower clouds are common due to the temperature change. Shadowing effects from the presence of hills and vegetation are caused by a lower altitude of the sun.

In order to take advantage of this knowledge, a very simple strategy was experimented: to increase the classifier's decision boundary during the early day hours, i.e., the decision threshold for True Alarms is increased for the early morning since a higher frequency of false alarms is expected to occur during such time of the day.

$$D(t) = \begin{cases} 0.6 & \text{if} \quad t < 10{:}00\,\text{AM} \\ 0.5 & \text{if} \quad t \geq 10{:}00\,\text{AM} \end{cases} \tag{14}$$

Applying a time-based condition to the decision function D can improve model robustness as an exogenous variable that

TABLE 9. Performance comparison using and not using a time-of-day-based decision (TD) threshold.

Model	Acc (%)	Prec (%)	Rec (%)	F1 (%)	FAR (%)	AUC
DenseNet	99.3	99.2	**99.6**	99.4	1.01	**.9999**
SCAM-SCNN	98.9	99.6	98.4	99.0	0.40	.9993
DC-CNN (no TD)	99.7	99.8	**99.6**	99.7	0.20	**.9999**
DC-CNN (TD)	**99.8**	**100.0**	**99.6**	**99.8**	**0.00**	**.9999**

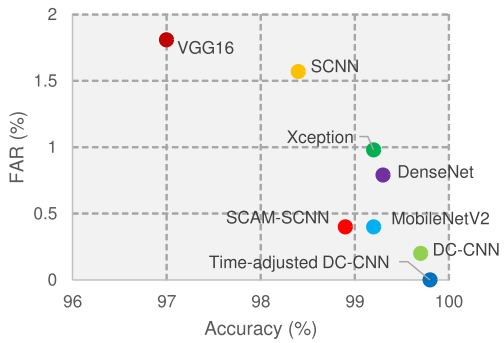


FIGURE 11. Accuracy vs FAR on smoke detection, for each tested model.

is not contemplated by the network is introduced, with the possibility of further refinement with more in-depth studies of time and climatic factors across a larger sample size, with no overhead to the model itself.

Table 9 also compares the performance of the proposed Dual-Channel CNN with and without the decision boundary adjusted as function of time. Despite representing the correct re-classification of only one previously misclassified true alarm, the potential impact of improving prediction quality with time-based conditions is apparent, as the accuracy rate was further improved to 99.8%, while false alarms did not occur on the test set.

Fig. 11 depicts a plot representing the accuracy and False Alarm Rate for all tested models, using the `bin-bbox` dataset. A remarkable increase in model performance is noted between the dual-channel based architectures (DC-CNN) and the remaining models, especially for the Time-adjusted DC-CNN architecture using the decision function in (14).

V. CONCLUSION

The work developed in this paper answers the initially posed research question – "Is it possible to improve the overall accuracy and to reduce the false alarm rate of an automatic wildfire detection system by applying Deep Learning methods?" – and demonstrates that there are significant benefits to the application of Deep Learning methods to achieve improved performance, as evidenced by the results obtained and detailed in the previous section: without time-based decision function adjustments, the proposed solution achieved an accuracy of 99.7% while keeping a low false alarm rate of 0.20%.

Experimentation with several data preparation strategies extracted valuable insights on the effects of binary and multi-class labeling, full image and bounding box image inputs, as well as data augmentation techniques, highlighting important considerations on its impact in model performance. Finally, on the perspective of modeling strategies, the comparisons between several Deep Learning-based classification models demonstrated the advantages and drawbacks of each implementation, primarily on its impact in feature extractions and performance implications.

One key aspect of this solution is the integration of a rule-based detection algorithm with a posterior Deep Learning model. The literature review identified several implementations of similar rule-based processes to extract suspect smoke regions to improve model performance, showcasing that this combination resulted in a higher reliability and overall performance for wildfire detection than fully Deep Learning-based systems. Considering the main issue of low specificity identified in the CICLOPE detection system, it was shown that the application of Deep Learning-based models over the universe of fire alarms can act as a secondary filtering stage to significantly reduce the number of false alarms without compromising the true alarms recall rate inherent to the primary rule-based decision system.

As for future directions, it may be worth to further optimize the existing architecture by tuning up its hyperparameters, experimenting with different network combinations and to investigate the use of attention mechanisms specifically designed for the detection of smoke. These adjustments may potentially improve accuracy and reduce false alarms further.

An important step to perform in the near future is to conduct large-scale field tests and evaluate the proposed solution within the actual CICLOPE system. Additional work may be required to ensure real-time operation of the proposed model, namely possible adjustments oriented to the existing computational resources and the implementation of an integration strategy between the existing rule-based system and the deep learning model that ensures smooth data flow and efficient operation.

It can also be worth to evaluate the generalization of the proposed approach to other domains. While the initial deployment was applied to the specific context of the CICLOPE project, the inherent flexibility of the proposed solution can be easily applied to potentially enhance other rule-based anomaly detection tasks in different contexts. Its adaptability not only accelerates the integration process but also maximizes the solution's potential to contribute value across different domains.

ACKNOWLEDGMENT

The authors would like to thank INOV-INESC for providing them access to the images acquired in the scope of the CICLOPE project.

REFERENCES

[1] J. San-Miguel-Ayanz, D. Oom, T. Artes, D. Viegas, P. Fernandes, N. Faivre, S. Freire, P. Moore, F. Rego, and M. Castellnou, "Forest fires in Portugal in 2017," in *Science for Disaster Risk Management 2020: Acting Today, Protecting Tomorrow*, A. C. Valles, M. Marin Ferrer, K. Poljanšek, and I. Clark, Eds. Luxembourg, U.K.: EN Publications Office of the European Union, 2020, doi: 10.2760/571085.

[2] S. L. Manzello, *Encyclopedia of Wildfires and Wildland-Urban Interface (WUI) Fires*. Cham, Switzerland: Springer, 2020, doi: 10.1007/978-3-319-51727-8.
[3] P. Kourtz, "The need for improved forest fire detection," *Forestry Chronicle*, vol. 63, no. 4, pp. 272–277, Aug. 1987, doi: 10.5558/tfc63272-4.
[4] M. J. Page et al., "The PRISMA 2020 statement: An updated guideline for reporting systematic reviews," *BMJ*, Mar. 2021, doi: 10.1136/bmj.n71.
[5] A. Bouguettaya, H. Zarzour, A. M. Taberkit, and A. Kechida, "A review on early wildfire detection from unmanned aerial vehicles using deep learning-based computer vision algorithms," *Signal Process.*, vol. 190, Jan. 2022, Art. no. 108309, doi: 10.1016/j.sigpro.2021.108309.
[6] S. Khan, K. Muhammad, S. Mumtaz, S. W. Baik, and V. H. C. de Albuquerque, "Energy-efficient deep CNN for smoke detection in foggy IoT environment," *IEEE Internet Things J.*, vol. 6, no. 6, pp. 9237–9245, Dec. 2019, doi: 10.1109/JIOT.2019.2896120.
[7] S. Geetha, C. S. Abhishek, and C. S. Akshayanat, "Machine vision based fire detection techniques: A survey," *Fire Technol.*, vol. 57, no. 2, pp. 591–623, Mar. 2021, doi: 10.1007/s10694-020-01064-z.
[8] F. Zhang, W. Qin, Y. Liu, Z. Xiao, J. Liu, Q. Wang, and K. Liu, "A dual-channel convolution neural network for image smoke detection," *Multimedia Tools Appl.*, vol. 79, nos. 45–46, pp. 34587–34603, Dec. 2020, doi: 10.1007/s11042-019-08551-8.
[9] K. Gu, Z. Xia, J. Qiao, and W. Lin, "Deep dual-channel neural network for image-based smoke detection," *IEEE Trans. Multimedia*, vol. 22, no. 2, pp. 311–323, Feb. 2020, doi: 10.1109/TMM.2019.2929009.
[10] Z. Yin, B. Wan, F. Yuan, X. Xia, and J. Shi, "A deep normalization and convolutional neural network for image smoke detection," *IEEE Access*, vol. 5, pp. 18429–18438, 2017, doi: 10.1109/ACCESS.2017.2747399.
[11] Y. Valikhujaev, A. Abdusalomov, and Y. I. Cho, "Automatic fire and smoke detection method for surveillance systems based on dilated CNNs," *Atmosphere*, vol. 11, no. 11, p. 1241, Nov. 2020, doi: 10.3390/atmos11111241.
[12] S. Woo, J. Park, J.-Y. Lee, and I. So Kweon, "CBAM: Convolutional block attention module," 2018, *arXiv:1807.06521*.
[13] R. Ba, C. Chen, J. Yuan, W. Song, and S. Lo, "SmokeNet: Satellite smoke scene detection using convolutional neural network with spatial and channel-wise attention," *Remote Sens.*, vol. 11, no. 14, p. 1702, Jul. 2019, doi: 10.3390/rs11141702.
[14] J. Zeng, Z. Lin, C. Qi, X. Zhao, and F. Wang, "An improved object detection method based on deep convolution neural network for smoke detection," in *Proc. Int. Conf. Mach. Learn. Cybern. (ICMLC)*, vol. 1, Jul. 2018, pp. 184–189, doi: 10.1109/ICMLC.2018.8527037.
[15] W. Cai, C. Wang, H. Huang, and T. Wang, "A real-time smoke detection model based on YOLO-SMOKE algorithm," in *Proc. Cross Strait Radio Sci. Wireless Technol. Conf. (CSRSWTC)*, Fuzhou, China, Dec. 2020, pp. 1–3, doi: 10.1109/CSRSWTC50769.2020.9372453.
[16] Y. Huo, Q. Zhang, Y. Jia, D. Liu, J. Guan, G. Lin, and Y. Zhang, "A deep separable convolutional neural network for multiscale image-based smoke detection," *Fire Technol.*, vol. 58, no. 3, pp. 1445–1468, Jan. 2022, doi: 10.1007/s10694-021-01199-7.
[17] G. Wang, J. Li, Y. Zheng, Q. Long, and W. Gu, "Forest smoke detection based on deep learning and background modeling," in *Proc. IEEE Int. Conf. Power, Intell. Comput. Syst. (ICPICS)*, Jul. 2020, pp. 112–116, doi: 10.1109/ICPICS50287.2020.9202287.
[18] R. Xu, H. Lin, K. Lu, L. Cao, and Y. Liu, "A forest fire detection system based on ensemble learning," *Forests*, vol. 12, no. 2, p. 217, Feb. 2021, doi: 10.3390/f12020217.
[19] Z. Wang, C. Zheng, J. Yin, Y. Tian, and W. Cui, "A semantic segmentation method for early forest fire smoke based on concentration weighting," *Electronics*, vol. 10, no. 21, p. 2675, Oct. 2021, doi: 10.3390/electronics10212675.
[20] Y. Li, A. Wu, N. Dong, J. Han, and Z. Lu, "Smoke recognition based on deep transfer learning and lightweight network," in *Proc. Chin. Control Conf. (CCC)*, Jul. 2019, pp. 8617–8621, doi: 10.23919/ChiCC.2019.8865302.
[21] H. Wu, H. Li, A. Shamsoshoara, A. Razi, and F. Afghah, "Transfer learning for wildfire identification in UAV imagery," in *Proc. 54th Annu. Conf. Inf. Sci. Syst. (CISS)*, Mar. 2020, pp. 1–6, doi: 10.1109/CISS48834.2020.1570617429.
[22] Q.-X. Zhang, G.-H. Lin, Y.-M. Zhang, G. Xu, and J.-J. Wang, "Wildland forest fire smoke detection based on faster R-CNN using synthetic smoke images," in *Proc. 8th Int. Conf. Fire Sci. Fire Protection Eng.*, vol. 211, 2017, pp. 441–446, doi: 10.1016/j.proeng.2017.12.034.
[23] M. Park, D. Q. Tran, D. Jung, and S. Park, "Wildfire-detection method using DenseNet and CycleGAN data augmentation-based remote camera imagery," *Remote Sens.*, vol. 12, no. 22, p. 3715, Nov. 2020, doi: 10.3390/rs12223715.
[24] Y. Luo, L. Zhao, P. Liu, and D. Huang, "Fire smoke detection algorithm based on motion characteristic and convolutional neural networks," *Multimedia Tools Appl.*, vol. 77, no. 12, pp. 15075–15092, Jun. 2018, doi: 10.1007/s11042-017-5090-2.
[25] A. Gagliardi, F. de Gioia, and S. Saponara, "A real-time video smoke detection algorithm based on Kalman filter and CNN," *J. Real-Time Image Process.*, vol. 18, no. 6, pp. 2085–2095, Dec. 2021, doi: 10.1007/s11554-021-01094-y.
[26] D.-K. Kwak and J.-K. Ryu, "A study on the dynamic image-based dark channel prior and smoke detection using deep learning," *J. Electr. Eng. Technol.*, vol. 17, no. 1, pp. 581–589, Jan. 2022, doi: 10.1007/s42835-021-00880-9.
[27] S. Dutta and S. Ghosh, "Forest fire detection using combined architecture of separable convolution and image processing," in *Proc. 1st Int. Conf. Artif. Intell. Data Anal. (CAIDA)*, Apr. 2021, pp. 36–41, doi: 10.1109/CAIDA51941.2021.9425170.
[28] A. Ng, *Machine Learning Yearning*, 2018. [Online]. Available: https://info.deeplearning.ai/machine-learning-yearning-book
[29] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, "Striving for simplicity: The all convolutional net," 2014, *arXiv:1412.6806*.
[30] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," 2015, *arXiv:1502.03167*.
[31] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," *Int. J. Comput. Vis.*, vol. 128, no. 2, pp. 336–359, Feb. 2020, doi: 10.1007/s11263-019-01228-7.

AFONSO M. GONÇALVES received the B.S. degree in naval military science from Escola Naval, in 2020, and the M.S. degree in integrated business intelligence systems from ISCTE-IUL, in 2022. He is currently a Data Engineer with Lovelytics, a data, AI, and analytics consultancy. His research interests include artificial intelligence, deep learning, and computer vision.

TOMÁS BRANDÃO was born in Lisbon, Portugal, in 1975. He received the Licenciatura, M.S., and Ph.D. degrees in electrical and computer engineering from the Technical University of Lisbon, Portugal, in 1999, 2002, and 2011, respectively. He is currently an Assistant Professor with the Department of Information Science and Technology, ISCTE-IUL. He is also an Integrated Researcher with ISTAR-IUL. His main research interests include computer vision and deep learning.

JOÃO C. FERREIRA (Senior Member, IEEE) is currently an Assistant Professor with habilitation at ISCTE-IUL. He has authored more than 350 papers in computer science, led more than 40 projects (including six as a principal investigator), and reviewed more than 250 scientific papers. He has also served on more than 25 scientific project evaluation panels. His research interests include data science, text mining, the IoT, AI, and their applications in health, energy, transportation, electric vehicles, and intelligent transportation systems (ITS). He was the IEEE CIS Chair, from 2016 to 2018. He is the Vice Chair of IEEE Blockchain PT, CIS PT Chapter, and Brussels AI and Robotics. He has organized major international conferences, such as OAIR 2013 and INTSYS (2018–2023). He is also a Guest Editor and a Topic Editor for MDPI journals in *Energies*, *Electronics*, and *Sensors*. In addition, he served as the President of the IEEE CIS in Portugal (2017–2018) and holds a patent for an edge computing monitoring system for fishing vessels.