INSTITUTO
UNIVERSITÁRIO
DE LISBOA

Wildfire Detection with Deep Learning - A Case Study for the CICLOPE Project

Afonso Meireles Gonçalves

Master's in Integrated Business Intelligence Systems

Supervisor:
PhD. Professor João Carlos Amaro Ferreira, Assistant Professor,
ISCTE-IUL

Co-Supervisor:
PhD. Professor Tomás Gomes da Silva Serpa Brandão, Assistant Professor,
ISCTE-IUL

October, 2022

TECNOLOGIAS
E ARQUITETURA

Department of Information Science and Technology

Wildfire Detection with Deep Learning - A Case Study for the CICLOPE Project

Afonso Meireles Gonçalves

Master's in Integrated Business Intelligence Systems

Supervisor:
PhD. Professor João Carlos Amaro Ferreira, Assistant Professor,
ISCTE-IUL

Co-Supervisor:
PhD. Professor Tomás Gomes da Silva Serpa Brandão, Assistant Professor,
ISCTE-IUL

October, 2022

To my dear wife, Rhea.

“If I have seen further, it is by standing on the shoulders of giants.”

- *Sir Isaac Newton*

Acknowledgements

This work and the master's degree it concludes would not have been possible without the unwavering support from my family and loved ones, friends and colleagues, and faculty members, all of whom I would like to express my sincere gratitude and due credits for the part they have taken in this important moment for me.

To my wife, for being my inspiration and always pushing me to be ambitious and confident of my capabilities.

To my family, for believing in me and always encouraging me to pursue my goals.

To my mother, for inspiring me with her strength and resilience.

To my supervisors, Professor João Carlos Ferreira and Professor Tomás Brandão, for their guidance and availability, and keeping me motivated and on the right track.

To INOV, for the support and necessary image resources provided which enabled this work.

To my colleagues, in particular to Inês Carvalho, Kenny Matos, Bruno Gil, and Joana Figueira, for all the knowledge share and the good-humoured hours we have spent working together in so many projects throughout our degree.

To all these people, for their contribution to this work and my own personal journey, I am deeply thankful, and will cherish the memories that are now tied with this dissertation. You are the giants whose shoulders I have stood on.

Resumo

Nos últimos anos Portugal tem observado uma elevada variabilidade em danos resultantes de fogos florestais, associada à alta imprevisibilidade de fenómenos climáticos como as fortes ondas de calor e Verões mais secos. A deteção correta e atempada de fogos rurais e florestais é portanto de grande importância para o sucesso no combate e contenção de incêndios, sendo que os fogos aumentam exponencialmente a velocidade de expansão desde o momento da ignição. No campo da deteção precoce de fumo e incêndios, o projeto CICLOPE é atualmente pioneiro no emprego de uma vasta rede de Torres de Aquisição Remota para a prevenção e monitorização de incêndios florestais, aliado a um sistema automático de deteção de colunas de fumo baseado em regras, cobrindo mais de 2,700,000 hectares de território rural e florestal em Portugal continental. No entanto, os desafios inerentes à deteção automática de colunas de fumo levantam problemas com elevadas taxas de Falsos Alarmes que afetam a qualidade de classificação e sobrecarregam os Centros de Gestão e Controlo com inúmeros falsos alarmes. Esta dissertação tem como objetivo avaliar o potencial de melhoria da taxa de deteção e especificidade com a implementação de arquiteturas de Deep Learning e propõe uma solução como Prova de Conceito baseada numa Dual-Channel CNN que pode ser implementada como uma segunda camada de confirmação da classificação de forma a refinar o sistema de deteção automática do CICLOPE. A solução proposta toma partido da elevada cobertura de verdadeiros alarmes de incêndio do sistema atual ao utilizar apenas as imagens associadas às deteções de alarmes e respetiva região delimitada do objeto de suspeição de incêndio. A rede Dual-Channel CNN combina uma arquitetura DenseNet do estado da arte e uma nova rede seletiva com módulos de atenção espacial e de canal, treinadas separadamente com dados obtidos do sistema CICLOPE, fundindo os atributos extraídos por cada rede numa camada de concatenação. Os resultados experimentais da Prova de Conceito indicam que a Dual-Channel CNN proposta atinge melhores resultados do que as redes de cada canal individual, e efetivamente alcança uma elevada taxa de deteção de 99.7% e uma baixa taxa de Falsos Alarmes de apenas 0.20%.

Palavras-chave: Deteção de Incêndios; Deteção de Fumo; Redes Neuronais Convolucionais; Aprendizagem Aprofundada; Visão por Computador.

Abstract

In recent years Portugal has seen a wide variability in wildfire damage that is associated with the high unpredictability of climatic events such as severe heatwaves and drier summers. Timely and accurate detection of wildland and rural fires is therefore of great importance for successful fire containment and suppression efforts, as wildfires exponentially increase 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 wildfire detection system, covering over 2,700,000 acres 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 Centres with numerous false alarms. This dissertation aims at evaluating the potential improvement in detection accuracy and specificity with the implementation of Deep Learning architectures and proposes a Proof of Concept 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 alarms images and respective bounding box coordinates as inputs. The Dual-Channel network combines a state-of-the-art 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 experimental Proof of Concept results show that the proposed Dual-Channel CNN outperforms both single-channel networks and effectively returns a high detection rate with an Accuracy of 99.7% and much lower False Alarm Rate of 0.20%.

Keywords: Wildfire Detection; Smoke Detection; Convolutional Neural Networks; Deep Learning; Computer Vision.

Table of Contents

List of Tables

List of Figures

Acronyms

AI – Artificial Intelligence

AP – Average Precision

AUROC – Area Under the Receiver Operating Characteristic

CAM – Channel Attention Module

CBAM – Convolutional Block Attention Module

CNN – Convolutional Neural Network

CRISP-DM – Cross-Industry Standard Process for Data Mining

CV – Computer Vision

DF – Decision Function

DL – Deep Learning

ECA – Efficient Channel Attention Module

FPR – False Positives Rate

GAN – Generative Adversarial Network

GMM – Gaussian Mixture Modeling

GradCAM – Gradient-weighted Class Activation Mapping

HSV – Hue Saturation Value

IoU – Intersection over Union

mAP – Mean Average Precision

ML – Machine Learning

MLP – Multi-Layer Perceptron

PRISMA – Preferred Reporting Items for Systematic Reviews and Meta-Analysis

ReLU – Rectified Linear Activation Unit

RGB – Red Green Blue

ROC – Receiver Operating Characteristic

SAM – Spatial Attention Module

SCAM-SCNN – Spatial and Channel Attention Modularized Selective Convolutional Neural Network

SCNN – Selective Convolutional Neural Network

SSD – Single-Stage Detector

TPR – True Positives Rate

UAV – Unmanned Aerial Vehicle

CHAPTER 1

Introduction

1.1. Motivation

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 seen in Figure 1.

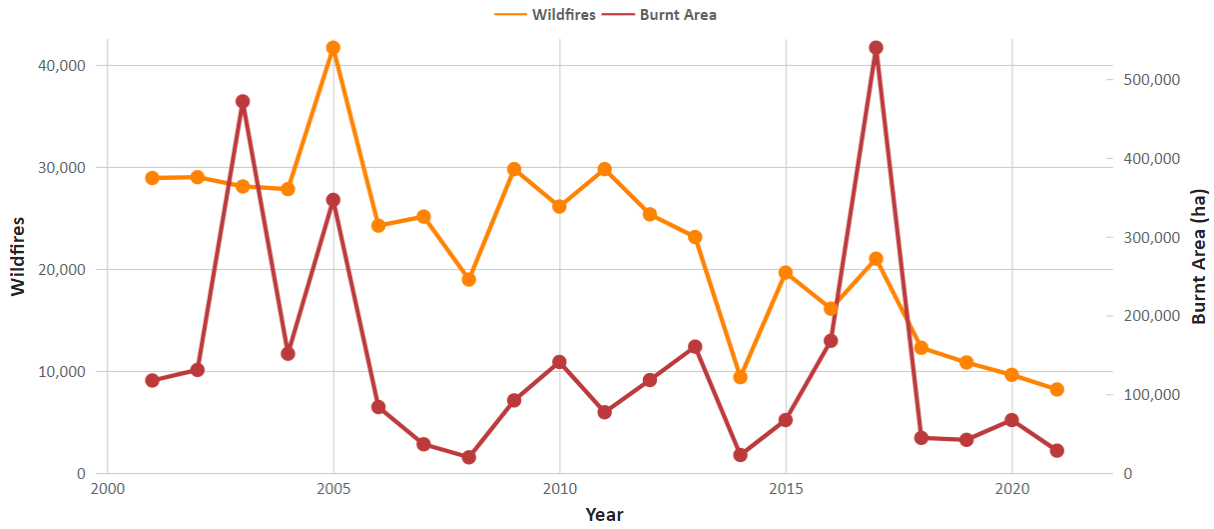

Figure 1 - Number of Wildfires and Total Burnt Area in mainland Portugal. Source: PORDATA

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 detection of fires, 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 exponentially 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 is an integrated wildfire surveillance system with automatic detection capabilities currently operating in Portugal, covering over 2.700.000 hectares of wildland and rural area, as shown in Figure 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 kilometres of effective zoom range, and a detection range of about 20 kilometres, 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 Centres for real-time observation and monitorization, 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.

Figure 2 - CICLOPE surveillance coverage in green, as of October 2022

CICLOPE's automatic wildfire detection system operates with a rule-based algorithm that continuously analyses the video feeds in a framewise basis, identifying regions with a sudden increase or decrease in brightness levels. While the detection system reports very good coverage ability, being able to correctly identify 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.

With an average of 33.9 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 applied through the detection algorithm, a high rate of false alarms results in a flood of noisy alarms that hide true alarm occurrences and reduce the operators' confidence and trust in the automatic detection system. Therefore, there is opportunity to improve the specificity and overall accuracy of the system, and exploring innovative Computer Vision and Deep Learning based solutions could prove of significant benefit to the integrated CICLOPE surveillance system.

The rapid expansion of Computer Vision technology and Deep Learning methods has been notorious in recent years, with several applications across industries replacing traditional rule-based mechanisms, with improved performances. The case of CICLOPE shows great potential for the integration of the current system with Deep Learning methods, as the high sensitivity and low specificity algorithm results in a reduction of the initial input image universe, maintaining most occurrences of true alarms. Hence, an integration applied as a posterior detection refinement model can prove more feasible and performant, as opposed to an overhaul of the total system.

1.2. Objectives

The work developed and presented in this dissertation aims to answer the following research question: "Is it possible to improve the overall accuracy and to reduce the false alarm rate of the CICLOPE automatic wildfire detection system by applying Deep Learning methods?". To answer this question, the following objectives are established for this dissertation:

- To advance knowledge on Deep Learning solutions for image-based wildfire detection;
- To analyse the feasibility and applicability of a Deep Learning solution that can integrate with the CICLOPE wildfire surveillance system;
- To develop a Proof of Concept (POC) based on Deep Learning architectures for wildfire detection in the scope of the CICLOPE system, that can be deployed by INOV;
- To evaluate the performance of the proposed Proof of Concept and its potential to improve overall wildfire detection rates and reduce False Alarm rates.

1.3. Outline of the Dissertation

After the introduction presented in the current chapter, this dissertation is organized according to the following structure:

Chapter 2 – Presents the state-of-the-art review for computer vision techniques applied to wildfire detection, analysing collected references using the PRISMA systematic review process.

Chapter 3 – Describes the image data utilized for Deep Learning applications with metadata analysis, along with all the data preparation techniques used to transform and pre-process the datasets for application to models.

Chapter 4 – Presents the Proof of Concept solution and the iterative experimentation process of applying models to the prepared datasets, interpreting and evaluating the results obtained for each stage of the modelling phase, as well as the discussion as a final assessment of the detection framework.

Chapter 5 – Rebates the research question and objectives defined in the Introduction by analysing the work developed and achieved results, highlighting future work to follow and improve upon the proposed framework.

1.4. Methodology

The development of the detection framework in this dissertation followed the Cross-Industry Standard Process for Data Mining (CRISP-DM) methodology. CRISP-DM defines an iterative process model of sequential stages that are commonly applied as a standard methodology for data mining and data science projects across industries. It is comprised of 6 stages – Business Understanding; Data Understanding; Data Preparation; Modelling; Evaluation; and Deployment –, as displayed in the diagram in Figure 3.

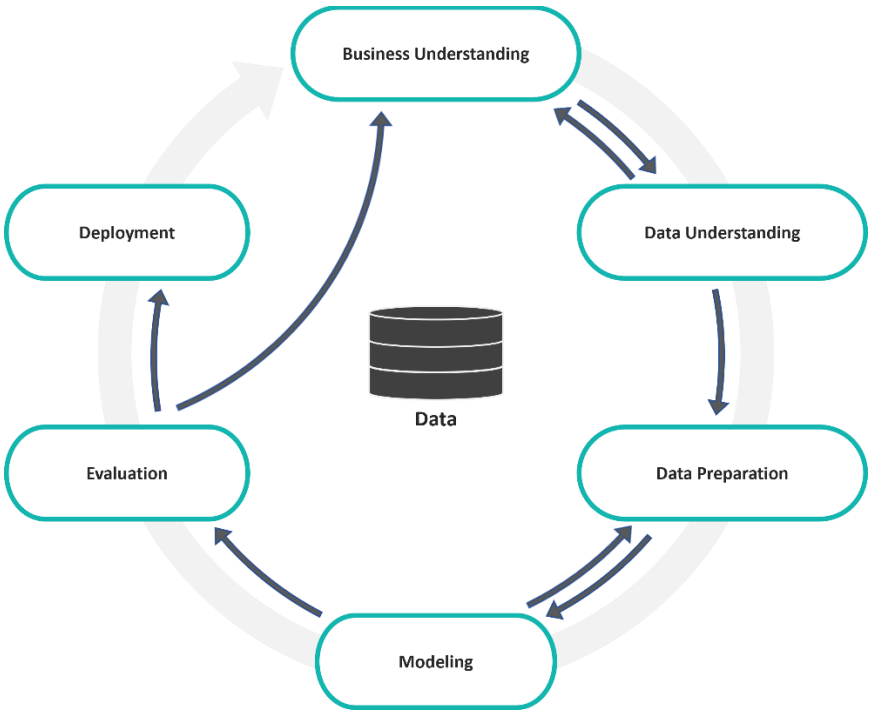

Figure 3 - The CRISP-DM process model

The Business Understanding stage focuses on establishing the project requirements and objectives from the business standpoint and serves as the cornerstone for the sequential steps and work to follow. This step of the process was achieved through meeting sessions with the team at INOV-INESC responsible for the CICLOPE project, discussing business needs, the current state of the system, potential solutions, and system and data constraints.

During the Data Understanding stage, collected image data from CICLOPE was analysed to examine feasibility of the previously discussed solutions, as well as identification of data-based constraints, data quality and diversity, further refining the problem scope and data preparation steps required. In the Data Understanding phase, it is important to return to Business Understanding to adjust business requirements with the reality of data and define a robust problem proposition.

The Data Preparation stage consisted of the sequence of data wrangling and transformation processes applied to prepare the data to be fed through subsequent models, described in Chapter 3.

In the Modelling stage, several network architectures were employed over the prepared datasets, as described in Chapter 4, where an iterative process is followed, tuning parameters and refining model architectures to achieve the Proof of Concept. As different models have distinct requirements from the data, a back-and-forth process takes place between adjusting the data preparation steps and the application of models.

The Evaluation stage consists of the analysis of the results obtained from the modelling stage and assessing them in the scope of the business requirements and initial objectives. Being an iterative methodology, the evaluation stage leads back to the Business Understanding stage, where it is essential to determine whether the solution is satisfactory, and if any business requirements were not considered. If the solution achieves the business objectives, the Deployment stage follows, while if there are improvements and considerations needed, the cycle is repeated.

The final Deployment stage includes preparing the solution for production, integrating with the current system, as well as establishing monitoring processes to evaluate live model performance and identify issues such as data drifts, concept drifts, or model degradation, ensuring model maintenance. This stage is not included in the scope of this dissertation.

CHAPTER 2

State of the Art Review

2.1. Background

Computer Vision (CV) can be defined as a subfield of Artificial Intelligence (AI) in which the goal is to automatically extract useful information from images. It is distinguished from Image Processing as the latter is defined by the creation of new images from the original through the application of different techniques to normalize, enhance, or transform images without gaining information about their content.

Information extracted from images can take many forms, and therefore we can differentiate several tasks associated with the problem of Computer Vision by the type of information obtained, with vast applicability in several real-world challenges. These can be grouped into three main categories – Image Classification, Object Detection, and Semantic Segmentation –, as demonstrated in Figure 4. In Image Classification, the aim is to classify an input image between predefined labels, while in Object Detection the goal is to locate objects within the image frame, and Semantic Segmentation applies pixel-wise classification to categorize the different objects in the image.

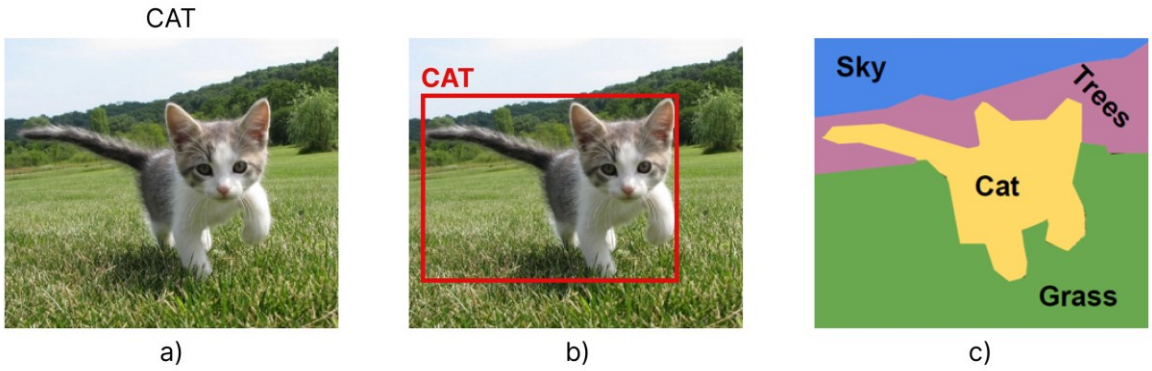

Figure 4 - Computer Vision problem examples: (a) Image Classification, (b) Object Detection, (c) Semantic Segmentation

In recent years, the topic of Computer Vision is often associated with Deep Learning (DL) as it has evolved exponentially and is widely used with successful applications, promising better performance, more automation, and reutilization of models with transfer learning [4].

Contrary to traditional Machine Learning (ML) techniques, in Deep Learning, features are extracted automatically from the image without the need for expert subject knowledge, and the training process involves learning these features and parameters to produce an output that solves a task. In the domain of Computer Vision, convolutional neural networks (CNN) have since been the favored architecture. In original CNN architectures, an image is inputted as matrices of pixel intensity values and is fed through a series of convolution and pooling layers. In convolution layers, filters are applied to the input matrices to extract specific features from the original image, and during the training process, the network will optimize the filter parameters to extract better features. In the pooling layers, the main objective is to reduce the dimensionality of the generated feature maps by, for example, extracting the maximum value in segments of the feature map, in the case of *max pooling*, or by computing an average [5].

Although early CNNs showed promise, with increased network complexity they demonstrated difficulty in scaling to larger problems, requiring large datasets and high computing power to train effectively. Since then, CNN models expanded, employing new strategies with many notable architectures developed requiring less training time and computing power even with limited data, and having been successfully applied in various tasks.

A problem that commonly affects model performance in CNNs is the lack of volume and variety in the training dataset, which affects generalization to different sets of images and inhibits scalability to real-world application. Models that produce good results on training metrics but show poor generalizability are overfitted to the training set.

One strategy to tackle the aforementioned problem is to apply data augmentation techniques in order to reduce overfitting. Several other strategies can reduce these effects, however, data augmentation takes a data-centric approach by focusing on the quality and diversity of the data used to train a model before applying changes to the model itself.

There are two main categories of data augmentation techniques – image manipulations and deep learning approaches. Image manipulation techniques can include geometric transformations such as flipping, cropping, or rotating images, as well as color space transformations, filtering, noise injection, and others, while Deep Learning approaches can involve the application of Generative Adversarial Network (GAN) models to produce new artificial images from the original dataset while retaining similar characteristics. Both of these strategies have shown an ability to improve model performance, although not all should be applied as it has also been proven that certain transformations can worsen overfitting effects. Thus, data augmentation techniques should be applied with consideration of the intrinsic bias of the original dataset [6].

As previously mentioned, one of the major hurdles in applying CNN models is the required processing time to effectively train the models to reach satisfactory performance. In the particular case of CNNs, this process involves optimizing the feature extraction part of the network in the convolution layers.

Transfer Learning consists of initializing a model with pre-trained weights that were previously optimized with a different dataset and task. By reutilizing pre-optimized weights on a new model, the filters generated to extract useful features from the previous training process are applied. In many cases, these weights are transferred from networks that have been trained on a very large and diverse dataset, extracting generic features that significantly speed up the training process. From here, the model can be trained entirely, optimizing the initial features for the new task, or trained in a portion, such as the densely connected layer [7].

This is particularly useful on very wide and deep networks with a large number of parameters where training from scratch is very time-consuming, computationally expensive, and sometimes unfeasible. By inheriting the pre-trained weights of state-of-the-art models, the new model can bypass the most cumbersome training stage and start from a base of generic features that can also improve the robustness and generalization of the model.

2.2. Related Work

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

SEARCH QUERY 1. *("smoke detect*" OR "wildfire detect*" OR "forest fire detect*") AND ("deep learning" OR "computer vision" OR "image classification" OR "semantic segmentation") AND PUBYEAR > 2009.*

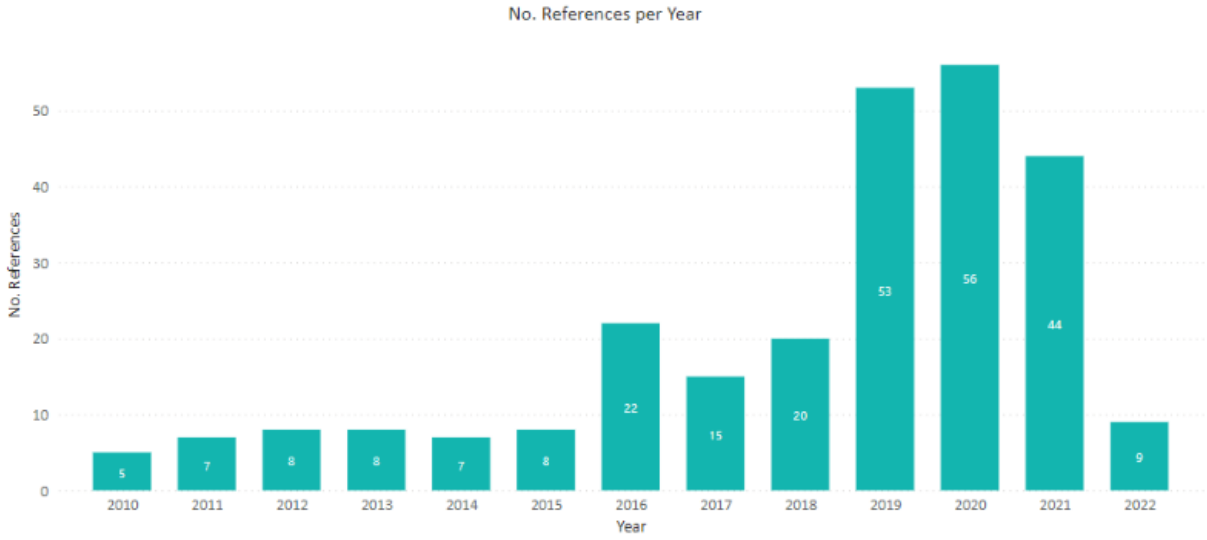

Figure 5 - Number of documents per year retrieved on February 7, 2022

The search query returned 262 documents, with the large majority dating to the last 3 years, as shown on Figure 5, which highlights the relative recency of deep learning studies for wildfire detection applications.

The identified references were organized and stored with Mendeley Reference Manager, and following the workflow displayed in Figure 6, a screening process was conducted through analysis of document title, abstract and keywords in order to determine applicability to the topic of this dissertation. Posteriorly, 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 20 papers used in this state-of-the-art review.

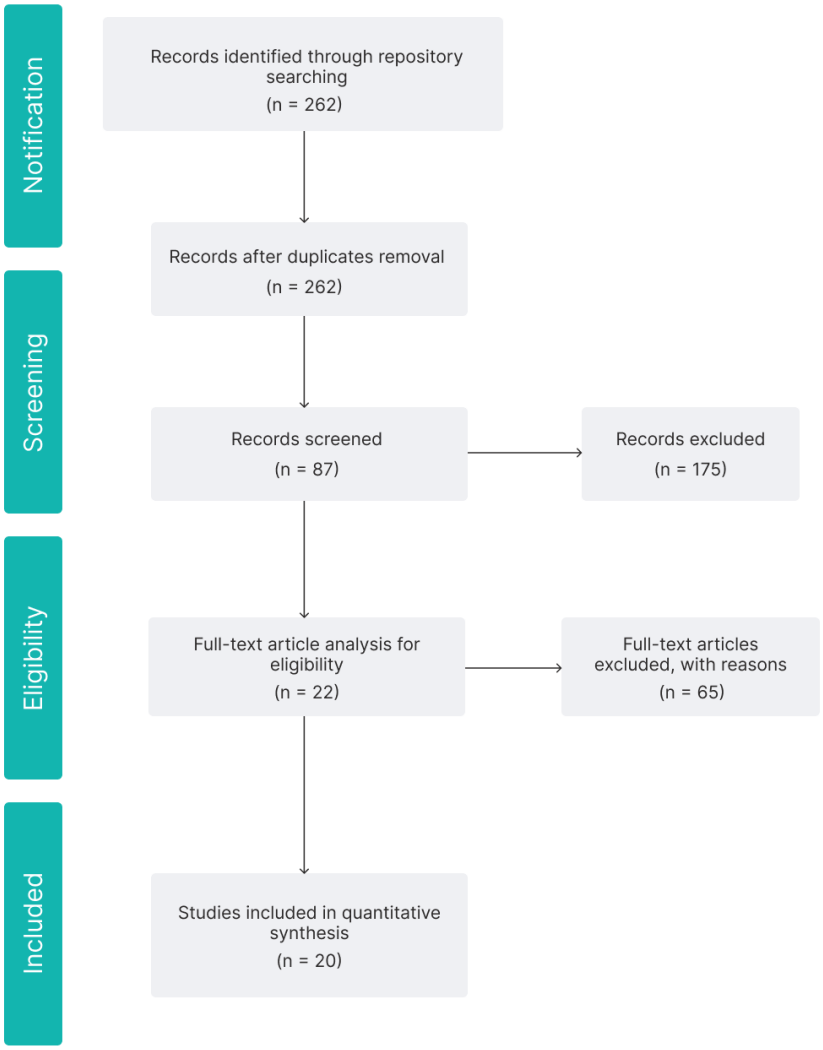

Figure 6 - PRISMA workflow diagram

The following section will present the findings which resulted from the systematic reviews process introduced, divided by the different sub-areas of Computer Vision methods applied to the problem of wildfire detection.

The first sub-section will summarize relevant works which approached the problem from an image classification perspective, where interesting solutions have been proposed, such as multi-class labelling strategies, dual-channel networks, deep convolution and normalization layers, as well as dilation and attention mechanisms for improved performance. The following sub-section will focus on object detection methods, emphasizing comparisons between single-stage and dual-stage detectors, and the application of ensemble methods with the combination of object detection and image classification models, followed by a sub-section on semantic segmentation approaches where concentration weight labelling is used for improved results. The sub-sections that follow focus on works that employed the previously introduced methods of transfer learning and data augmentation, analyzing their impact on smoke wildfire detection. Finally, we explore works that employed rule-based methods, particularly in combination with deep learning models, as it relates to the proposed framework detailed in this dissertation. A final summary table describes the performance metrics obtained from each work analyzed.

2.3. Computer Vision for Wildfire Detection

2.3.1. Image Classification

Considering the rapid evolution of computer vision applications based on deep learning methods, undertaking a literature review on the state-of-the-art developments for wildfire detection is relevant to determine the methods currently used and their success on this specific task.

In reference [8], 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, 5 applied image classification techniques, 7 applied object detection, and only 3 applied semantic segmentation. The authors concluded that smoke detecting models achieved better results than flame-based ones, especially in the early stages. However, smoke detection revealed higher difficulty in achieving good performance in nighttime images and in the presence of fog, clouds, and 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.

Reference [9] proposes a CNN model that can achieve good performance in both clear and foggy environments, suggesting a multi-class approach instead of binary classification, defining 4 labels – Non-Smoke, Smoke, Non-Smoke with fog, and fog. The authors applied a VGG16 type architecture pre-trained on ImageNet, which uses smaller filter sizes and shorter strides, and compared the model's performance to GoogleNet and AlexNet, achieving an accuracy of 97.72% and outperforming both models.

Another review paper [10] 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 [11], 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 *concat* layer. With this method the authors were able to combine 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, [12] 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 (DCNN) to various state-of-the-art architectures, as well as each component network, in terms of accuracy, detection rate, and false alarm rates over two different test sets. In both sets, the DCNN 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 Set 1 and Set 2. The best performing state-of-the-art networks on accuracy rates were Dense-Net (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 Dense-Net (1.08% and 1.10%). The proposed DCNN also performed better than each subnetwork alone, as SBNN achieved accuracies 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 DCNN demonstrates that a larger diversification of extracted features can produce a better generalizing model.

The aforementioned DNCNN was proposed in [13] and stands for Deep Normalization and Convolutional Neural Network. The authors replaced the traditional convolution layers in CNNs with normalization and convolutional layers, using batch normalization. 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 [14], 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.

Attention mechanisms have also been increasingly studied in their ability to improve performance in image classification tasks. A Convolutional Block Attention Module (CBAM) was proposed in [15] 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 extracted features, whereas SAM focuses on the most informative spatial location of the feature maps. In [16], the authors applied a similar mechanism in the proposed SmokeNet model and applied it to smoke detection in satellite imagery, classifying between 6 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.

2.3.2. 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 tend to be more computationally intensive than image classification models. Object detection models can have a two-stage or a single-stage architecture, where in the case of two-stage detectors the first stage selects regions of interest to be classified in the second stage, whereas single-stage detectors classify the image in one single pass.

In [17], the authors compared two state-of-the-art two-stage detectors (Faster R-CNN and R-FCN) and one single-stage detector (SSD), substituting 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 show that SSD is faster to process test images but 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 mAP of 56.04%.

Another common single-stage detector is the YOLO family of detectors. In reference [18], 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 the test image processing time.

Similarly, [19] 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 *1x1xn* 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%.

Reference [20] applied a dynamic background modeling mechanism to improve performance of an SSD detector with 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 consists of intersecting the SSD detection output with the extracted moving target in order 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 [21], 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, as well as 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.

2.3.3. Semantic Segmentation

Semantic segmentation approaches are more computationally intensive due to the classification of each pixel within the image set, and in the case of 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 [22] 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.

2.3.4. Transfer Learning

As mentioned, 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 [23], 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 [24], 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.

2.3.5. Data Augmentation

As also stated, data augmentation can too be beneficial, especially in the event of small and imbalanced datasets. In [25], 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 4 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 from 100.00% to 100.00% on video 4, suggesting that the applied data augmentation technique can improve detection ability on the same architecture.

The work in [26] 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%.

2.3.6. Rule-based Methods

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

Reference [27] 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 the test set 1 from 93.96% to 99.77%, and on the test set 2 from 93.37% to 99.06%.

A similar strategy was employed in [28], 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.

The dark-channel prior method was also applied in [29] 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 preprocessed images, and outperformed SSD and Faster R-CNN detectors, FireNet, rule-based optical flow and dark-channel preprocessing 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 reference [30], a different technique was adopted, applying multichannel binary thresholding and HSV colorspace thresholding over the original images. Binary thresholding is comprised of defining a fixed threshold value and minimizing or maximizing each pixel value based on if 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 depthwise 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%.

2.3.7. Summary

A summary of the selected studies is presented in Table 1, comparing each paper's results regarding method and approach followed, and performance metrics achieved.

In general, while many of the works analyzed introduce interesting solutions and positive outcomes, the emphasis is largely on the improvement of overall accuracy for wildfire detection, or detection quality through better IoU scores, in the case of object detection. However, as the case of CICLOPE shows, very good model accuracy does not imply a low rate of false alarms, which can be a difficult challenge to tackle. Therefore, this dissertation aims at filling this evidenced gap in the literature by focusing on model refinement through reduction of false alarms.

Table 1 - Comparative analysis of selected studies

Ref	Method	Backbone	Approach	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	FA Rate (%)	mAP (%)	mIoU (%)
[9]	Multi-class approach for foggy environments	VGG16	Image Classification	97.72	98	97	98	2.30		
[11]	Dual-Channel CNN	AlexNet + CNN	Image Classification	99.33						
		AlexNet with Transfer Learning		99.08						
[12]	Dual-Channel CNN	SBNN + SCNN	Image Classification	99.7		99.5		0.12		
		SBNN		98.3		97.3		0.96		
		SCNN		98.6		97.6		0.84		
[13]	DNCNN	ZF-Net based	Image Classification	97.83		95.28		0.48		
		ZF-NET		97.18		93.29		0.24		
[14]	Dilated CNN	CNN	Image Classification	99.53	98.27	97.46	98.92			
[16]	Spatial and Channel-wise Attention mechanism	SmokeNet	Image Classification	85.1				10.06		
		SCResNet		82.35				15.58		
		ResNet		73.51				36.67		
[17]	Comparison Feature Extractors	Faster R-CNN with Inception ResNet V2	Object Detection						56.04	
[18]	YOLO-SMOKE with channel attention and dropblock layer	YOLOv3 with darknet-53 based	Object Detection						81.95	
[19]	YOLO based with depthwise separable convolutions	YOLOv4 with CSPdarknet53 based	Object Detection	97.8	98.5	97.4	97.9	1.7		
[20]	Dynamic foreground extraction with ViBe	SSD with MobileNet	Object Detection						51.87	
[21]	Ensemble method	YOLOv5 + EfficientDet + EfficientNet	Mixed					0.3	79	
[22]	Concentration weight labeling	MobileNet + PSPnet	Semantic Segmentation							75.38
[23]	Transfer Learning	VGG16 with Transfer Learning	Image Classification	99.73						
		AlexNet		98.91						
[24]	Transfer Learning	MobileNetV2	Image Classification	99.3						
[25]	Data augmentation with synthetic smoke implantation	Faster R-CNN + augmented data	Object Detection			82.11				
		Faster R-CNN + real data				74.19				
[26]	Data augmentation with CycleGAN-generated samples	Dense-Net based	Image Classification	98.271	99.380	96.976	98.163			
[27]	Dynamic background subtraction and dark-channel prior	CNN	Image Classification	99.77		99.87		1.33		
[28]	Foreground extraction with Kalman filtering	CNN	Image Classification	84.38	86.36	90.49	88.37			
[29]	Lucas-Kanade optical flow and dark-channel prior	Inception V3	Object Detection	97.0	98.0	96.1	97.0			
[30]	Multichannel binary thresholding and HSV colorspace thresholding	Deep Separable CNN	Image Classification	93.60	91.65	98.10	95.77	5.28		

CHAPTER 3

Data Preparation

3.1. Datasets

The datasets used for wildfire detection in this dissertation are comprised of 4 batches of RGB images captured by the *Ciclope* wildfire surveillance system cameras mounted on watchtowers, as depicted on Figure 7, and represent real wildfire alarms signaled by the current smoke detection algorithm in operation, having been posteriorly manually classified as true alarms and false alarms.

Figure 7 - Example of a Ciclope surveillance camera mounted on a watchtower. Obtained from https://www.ciclope.pt/Gallery.aspx

Table 2 presents the image distributions across the 4 datasets used throughout this dissertation. Dataset *CasteloBranco_true_alarms_v2* represents 538 annotated images of true smoke alarms with bounding box identification of the alarm region as outputted by the current detection system. Dataset *Leiria_false_alarms_Floresta_e_campos_v1* contains a collection of 718 annotated images of false alarms with bounding box identification, also classified as false alarms of the type *"Fields and Forest"*, as the detected objects correspond to shadowing and light shifts phenomena occurring on natural fields and structures. Dataset *Leiria_false_alarms_Nuvens_e_nevoeiros_v1* also contains 3231 annotated images of false alarms with bounding box identification, though classified as belonging to the type *"Clouds and Fog"*, as the detected objects represent cloud objects identified as smoke, or fog occurrences. Dataset *Extracted_fires_2020_Ground* comprises a collection of 4504 annotated images of true alarms, without bounding box coordinates. 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, along with the bounding box coordinates.

To each dataset, a 70%-20%-10% split was applied to create training, validation, and test sets, so that each dataset has equal representation along each set. The training set will be the collection used to train/fit models, and therefore it contains the larger portion of the total collection of images, with a total of 6292 images in this set. This set of images is used during the models' training to optimize parameters in order to minimize the loss function. The validation set, sometimes also referred to as the development set, is used as a fine-tuning set to evaluate model performance during training, being comprised of 1796 samples. The test set is the final set used to evaluate model performance once the tuning process is completed. The reasoning behind the use of validation and test sets, as opposed to a simpler training/test split, is described in [31], and reflects that while the training set is directly used for parameter optimization, incorporating a level of bias, there is an indirect bias associated with validation sets as these are used as a control group to evaluate model performance during the training process, therefore a separate test set will provide a more accurate description of true model performance and generalizability.

Table 2 - Original datasets used

Dataset	Training	Validation	Test	Total
CasteloBranco_true_alarms_v2	376	107	55	538
Leiria_false_alarms_Floresta_e_campos_v1	502	143	73	718
Leiria_false_alarms_Nuvens_e_nevoeiros_v1	2262	646	323	3231
Extracted_fires_2020_Ground	3152	900	452	4504
Total	**6292**	**1796**	**903**	**8991**

3.1.1. Metadata Analysis

The collected images were gathered from 2018 to 2021, with the earliest image taken on 2018-10-03, and the latest recording 2021-10-27. In the span of total available dates, 195 days have associated images, which corresponds to 17.4% of all possible dates, as illustrated in Figure 8. The highest number of total images collected from a single date is 405 in 2021-10-01, while the lowest amount is of only 1 across several dates.

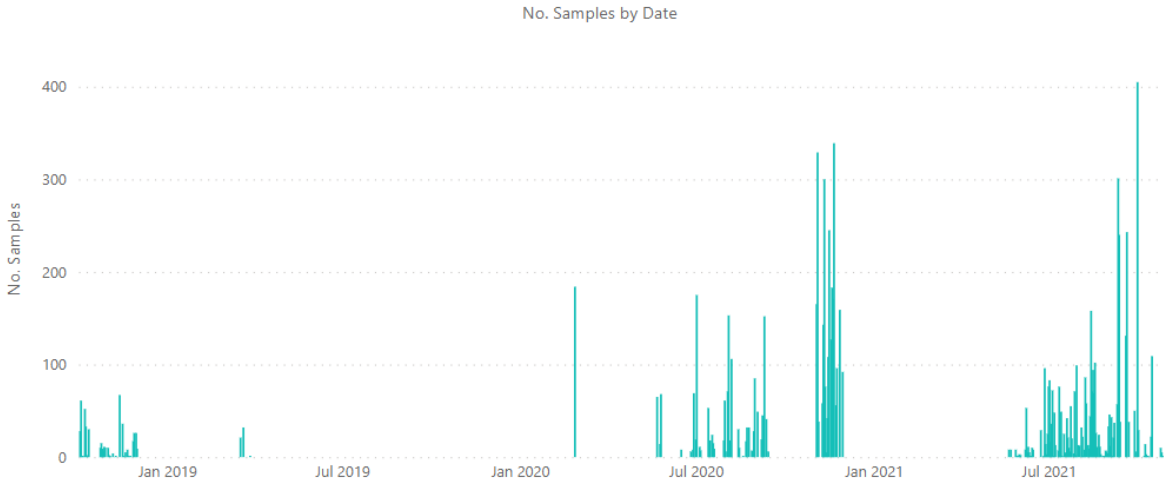

Figure 8 - Number of collected images by date

Figure 9 represents the distribution of images along months of the year, where in the case of False Alarms its distribution is concentrated in the span of months between May and October, with September being the month with the highest record of alarms, totaling 1297. In the case of true alarms, the month of November overwhelmingly leads with 2938 collected images, having otherwise a similar distribution from May to October, along with some samples gathered in February and March.

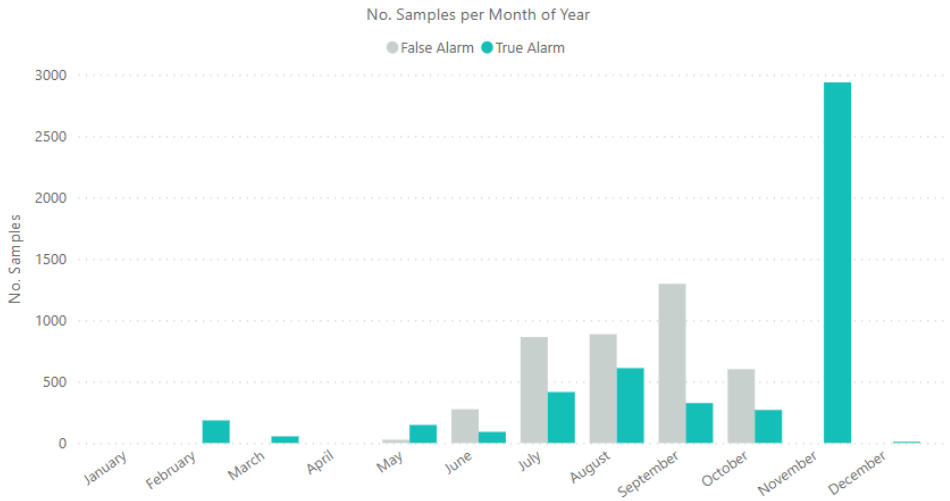

Figure 9 - Number of collected images by Month

Figure 10 illustrates the distribution of images across time of day while Figure 11 rounds up timestamps to the nearest hour to provide a clearer analysis of image distribution across hour of day. In both figures, a clear distinction can be asserted between the distribution of False Alarms and True Alarms, where the former predominantly occurs in the earlier periods of day, between 08:00 AM and 10:00 AM, whereas the latter exhibits a prevalence between 11:00 AM and 05:00 PM.

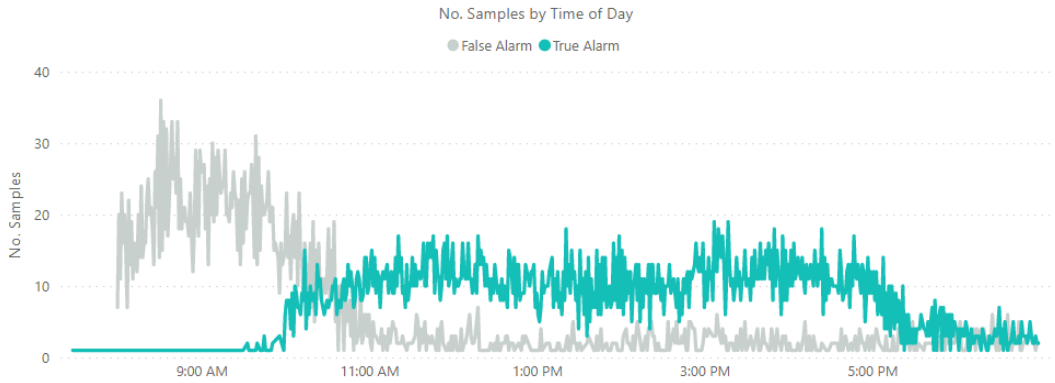

Figure 10 - Number of collected images by time of day

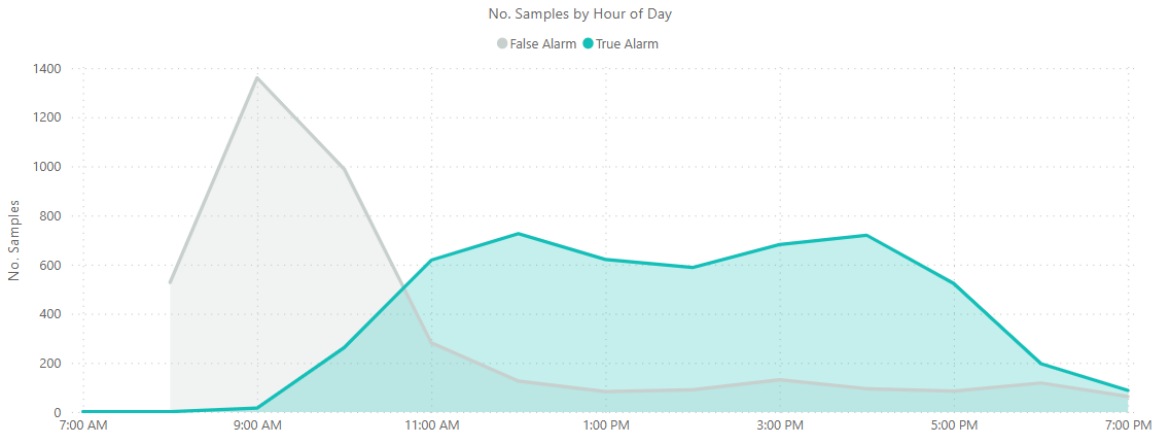

Figure 11 - Number of collected images by hour of day

An analysis of the metadata associated with the gathered images reveals that a considerable time span is considered in the collected images, and while monthly distributions are less informative regarding differences between the two classes, a stark distinction is discernible in terms of distributions along time of day, where many more false alarms have been captured during the early periods of day, while in the case of true alarms such occurrence is much rarer.

3.2. Classes

As previously detailed, in reference [9], a multi-class approach was implemented to deal with the challenge of smoke detection in foggy environments. Taking into consideration the characteristics of the datasets utilized in this dissertation, two separate datasets were created with distinct labeling strategies.

Dataset *two-classes* will employ a binary classification strategy, where *True Alarms* represent verified wildfire smoke occurrences and will be assigned a label of 1, while *False Alarms* congregate all other images without the verified presence of wildfire smoke, with an assigned label of 0.

Dataset *three-classes* will follow a multi-class approach and classify between *Smoke*, *Clouds and Fog*, and *Fields and Forest*. By comparing performances 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.

Table 3 - Distributions of Binary and Multi-class labeling datasets

Dataset	Class	Training	Validation	Test	**Total**
two-classes	True Alarm	3528	1007	507	**5042**
	False Alarm	2764	789	396	**3949**
three-classes	Smoke	3528	1007	507	**5042**
	Clouds and Fog	2262	646	323	**3231**
	Fields and Forest	502	143	73	**718**

As Figure 12 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 backgrounds 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, being often times 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.

Figure 12 - Sample images representative of Smoke, Clouds and Fog, and Fields and Forest classes

3.3. 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 [27] - [30], 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 abovementioned methods to the use case of this dissertation, where the images collected are gathered from a rule-based image processing algorithm that produces bounding box coordinates, enabling the extraction of the suspected region.

However, as previously stated, images collected from the dataset *Extracted_fires_2020_Ground* do not contain associated bounding box coordinates identification and were thus manually classified, resulting in a distinct bounding box definition as the rule-based system which produces very fine boxes that are small in area. This can be verified as the average bounding box area extracted from the detection system is 1929.8 square pixels, while the manually identified bounding boxes averaged 43094.2 square pixels in area.

In order to standardize bounding box dimensions, a padding constant was applied in the extraction function, which is defined as follows, where ***I*** represents the input image, and x_1, x_2, γ_1, and γ_2, stand for the coordinates along the x-axis and γ-axis, and *pad* takes the value of 5 pixels for manually annotated images, and 150 pixels for system annotated images:

$$f(x) = \begin{cases} 0, & x - pad < 0 \ \cap \ x \in [x_1, \gamma_1] \\ x - pad, & x - pad \geq 0 \ \cap \ x \in [x_1, \gamma_1] \\ length(\boldsymbol{I}), & x + pad > length(\boldsymbol{I}) \ \cap \ x \in [x_2] \\ x + pad, & x + pad \leq length(\boldsymbol{I}) \ \cap \ x \in [x_2] \\ height(\boldsymbol{I}), & x + pad > height(\boldsymbol{I}) \ \cap \ x \in [\gamma_2] \\ x + pad, & x + pad \leq height(\boldsymbol{I}) \ \cap \ x \in [\gamma_2] \end{cases} \tag{3.1}$$

Figure 13 - Examples of Bounding Box extraction

Two additional datasets *two-classes-bbox* and *three-classes-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*, as characterized in Table 4.

Comparing model performance across datasets allows for an evaluation on the best preprocessing 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.

Table 4 - Dataset distributions after creation of Bounding Box datasets

Dataset	Class	Training	Validation	Test	**Total**
two-classes	True Alarm	3528	1007	507	**5042**
	False Alarm	2764	789	396	**3949**
two-classes-bbox	True Alarm	3528	1007	507	**5042**
	False Alarm	2764	789	396	**3949**
three-classes	Smoke	3528	1007	507	**5042**
	Clouds and Fog	2262	646	323	**3231**
	Fields and Forest	502	143	73	**718**
three-classes-bbox	Smoke	3528	1007	507	**5042**
	Clouds and Fog	2262	646	323	**3231**
	Fields and Forest	502	143	73	**718**

3.4. 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 *two-classes* and *two-classes-bbox*, the imbalance is not significant as the majority class *True Alarms* represents 56.1% of the training set, and *False Alarms* makes up the remaining 43.9%. However, in the case of the multi-class labeled datasets *three-classes* and *three-classes-bbox*, the *Smoke* class represents 56.1%, while *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 reflect the sky and ground, respectively.

The horizontal flip operation, demonstrated in Figure 14, can be represented as the following reflection function:

$$g(x) = f(-x) \tag{3.2}$$

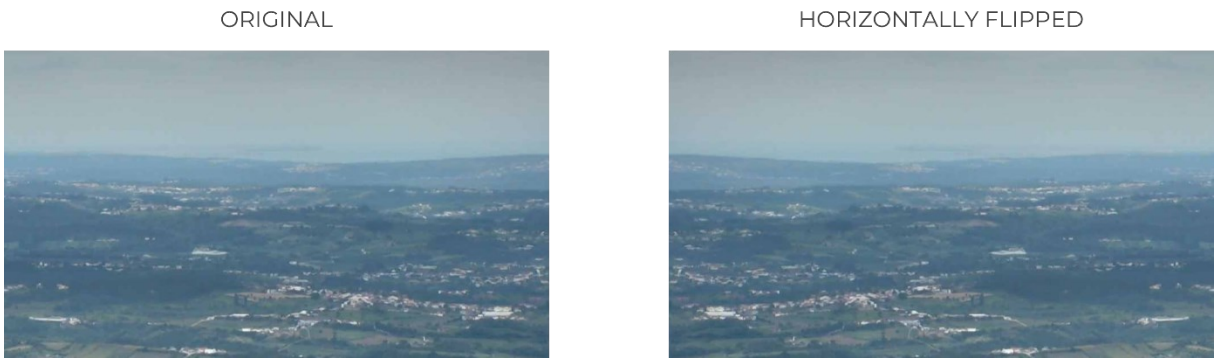

Figure 14 - Example of data augmentation through horizontal flipping

The flipping operation was realized only on the training set, to maintain the original distribution throughout the validation and test sets and each of the previously generated datasets was replicated so as 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 distributions of the final eight datasets created for model application are presented in Table 5.

Table 5 - Dataset distributions after creation of augmented datasets

Dataset	Class	Training	Validation	Test	**Total**
two-classes	True Alarm	3528	1007	507	**5042**
	False Alarm	2764	789	396	**3949**
two-classes-aug	True Alarm	3528	1007	507	**5042**
	False Alarm	3266	789	396	**4451**
two-classes-bbox	True Alarm	3528	1007	507	**5042**
	False Alarm	2764	789	396	**3949**
two-classes-bbox-aug	True Alarm	3528	1007	507	**5042**
	False Alarm	3266	789	396	**4451**
three-classes	Smoke	3528	1007	507	**5042**
	Clouds and Fog	2262	646	323	**3231**
	Fields and Forest	502	143	73	**718**
three-classes-aug	Smoke	3528	1007	507	**5042**
	Clouds and Fog	2262	646	323	**3231**
	Fields and Forest	1004	143	73	**1220**
three-classes-bbox	Smoke	3528	1007	507	**5042**
	Clouds and Fog	2262	646	323	**3231**
	Fields and Forest	502	143	73	**718**
three-classes-bbox-aug	Smoke	3528	1007	507	**5042**
	Clouds and Fog	2262	646	323	**3231**
	Fields and Forest	1004	143	73	**1220**

CHAPTER 4

Detection Framework

In this chapter the framework for wildfire smoke detection in the scope of the *CICLOPE* project is presented, describing the sequential workflow followed, results obtained for each stage, and the findings and justifications for decisions made throughout the development process.

The following sections will present an initial transfer learning approach, cross-evaluating a set of state-of-the-art models applied over the several datasets defined in chapter 3, with the goal of determining the most suitable data preparation strategy, and best performing network architecture.

The subsequent section introduces SCAM-SCNN, a novel detail selective network with spatial and channel attention modules, exploring the advantages of attention mechanisms for wildfire smoke detection, and assessing the performance of a fully trained selective model.

The final stage of the framework proposes a Dual-Channel CNN, combining the best performing transfer learning-based model with a selective network as two channels of a common network, with the goal of improving detection ability and prediction quality.

4.1. Initial Transfer Learning Approach

In the early stages of developing machine learning applications, starting off with simple approaches that can quickly return results can be very beneficial, as it enables a fast output that can be examined and orientate the workflow to follow, rather than investing too much time in a detailed approach that may lead to less conclusive results [31].

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 dissertation. The goal of this initial framework is to analyse the results obtained from the various models implemented regarding the defined datasets, understand the differences of each implementation and their impact on the problem, and ultimately select the most promising dataset and best performing model.

4.1.1. Transfer Learning Models

As stated before, transfer learning enables the initialization of models with pre-trained weights that were previously optimized over large, extensive collections of real images, thus extracting many useful general features that highlight distinct characteristics in an image, such as edges, textures, colorations, etc. Applying transfer learning can produce faster results that are sufficiently robust and facilitate a

better understanding of the underlying data, while training the same models from scratch would reveal very time and resource consuming, considering the complexity of such models.

The state-of-the-art models selected for this initial approach reflect the findings that resulted from the literature review performed and that exhibited satisfactory outcomes in the task of smoke and wildfire detection. The subsections that follow introduce the key properties of each model architecture and their attributes.

4.1.1.1. VGG16

The VGG16 architecture was initially proposed in 2013 [32], and has since been regarded as one of the most impactful architectures in the field of image recognition and widely used across numerous applications, having been implemented in [9] and [23].

Contrary to most industry standard architectures until then which applied larger initial receptive fields of 7×7, and even 11×11, with wide strides, VGG16 applies very small 3 × 3 filters that, when stacked in two and three convolution layers, effectively achieve 5×5 and 7×7 receptive fields, but introduce non-linear activations between each layer, improving feature discrimination. Additionally, it also greatly reduces the number of weights, as these can be defined as a function of the number of input channels $\boldsymbol{C}$, filter size $\boldsymbol{k}$, and the number of layers $\boldsymbol{L}$ as $\boldsymbol{L}(\boldsymbol{k}^2\boldsymbol{C}^2)$.

The model's architecture is represented in Figure 15. It is comprised of two blocks of two convolution layers followed by a max pooling layer, and three blocks of three convolution layers followed by max pooling layers, with the number of filters increasing with depth, amounting a total of 14,793,777 parameters.

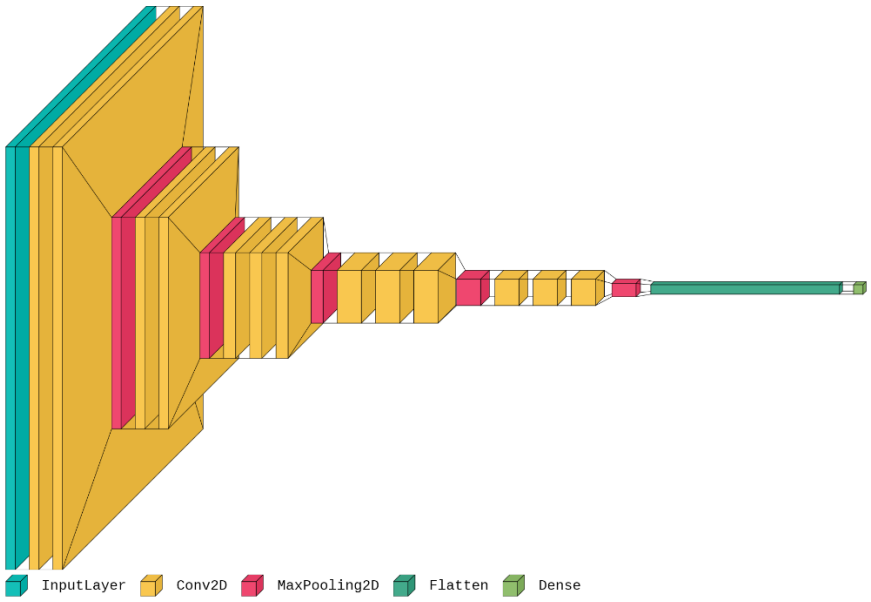

Figure 15 - VGG16 layers diagram

4.1.1.2. Xception

The Xception network was proposed in [33] in 2016, schematically inspired by VGG16, and improved upon the previous Inception architecture by introducing depthwise separable convolutions, decoupling the mapping of spatial correlations and cross-channel correlations, with residual connections, and is represented in Figure 16, amounting a total of 20,961,833 parameters.

Depthwise separable convolutions divide the traditional convolution into two separate steps – depthwise convolutions apply filter kernels on each individual input feature map one at a time, where ***M*** represents the number of feature maps, and subsequently pointwise convolutions aggregate the ***M*** generated maps by applying a 1×1 kernel, effectively completing the convolution operation. This produces comparable performance, while significantly reducing the computational costs of the convolution operation, which traditionally could be expressed as $\boldsymbol{h_i \cdot w_i \cdot d_i \cdot d_j \cdot k^2}$, where $\boldsymbol{h_i \times w_i \times d_i}$ represents the input height × width × depth of the input tensor, $\boldsymbol{k}$ represents the filter kernel size, and $\boldsymbol{d_j}$ the depth of the output tensor, while the cost of depthwise separable convolutions can be computed as $\boldsymbol{h_i \cdot w_i \cdot d_i}$ $(\boldsymbol{d_j} + \boldsymbol{k^2})$. In practice, a traditional convolution with a 3×3 filter kernel, a desired output of 64 feature maps, and an input tensor of $14 \times 14 \times 32$ would require 3,612,672 multiplications, while in the case of depthwise separable convolutions the operation would require only 457,856 multiplications.

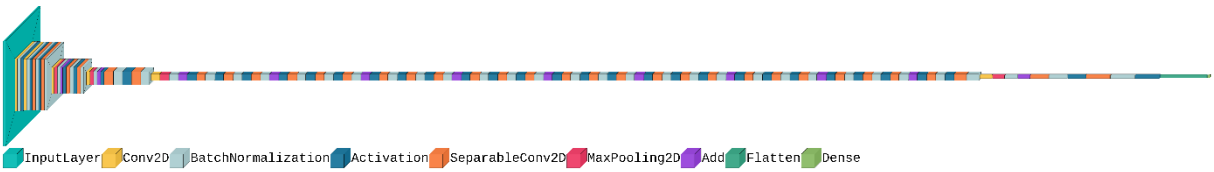

Figure 16 - Xception layers diagram

4.1.1.3. MobileNetV2

MobileNetV2, proposed in [34] in 2018, improved on the previous MobileNetV1 network which introduced a lightweight model designed for edge computing applications, by making use of depthwise convolutions as a means of reducing computing costs, inspired by Xception, having been implemented in [20], [22], and [24].

MobileNetV2 expands on its predecessor on two main concepts – Inverted Residuals and Linear Bottlenecks. Original residual blocks connect wide layers with a skip connection, with narrow layers in between, which performs 1×1 convolutions on the wide input and output layers, and 3×3 convolutions on the middle narrow layers, which reduces the number of parameters. However, MobileNetV2 applies Inverted Residual blocks where the input and output are the narrow layers and the middle layers are wide, based on the idea that there is information loss in the standard residuals block, as represented in Figure 17.

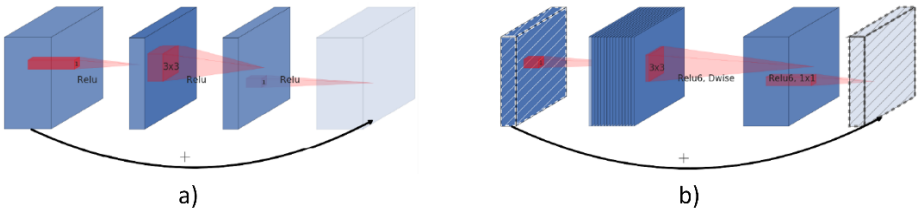

Figure 17 - (a) Original Residuals Block, (b) Inverted Residuals Block [34]

Non-linear activation functions introduce a loss of information, where in the case of the widely used rectified linear activation unit (ReLU) values below 0 are discarded, which becomes more evident with a lower number of channels in Inverted Residuals blocks. To solve this problem, MobileNetV2 inserts Linear Bottleneck layers into the convolutional blocks, where the last convolution layer in a residual block has a linear activation, before being added to the input activations.

The lightweight design of MobileNetV2 significantly reduces the total number of parameters to 2,320,705, while retaining very solid performance. The general layer diagram of the network is represented in Figure 18.

Figure 18 - MobileNetV2 layers diagram

4.1.1.4. DenseNet

DenseNets, proposed in [35] in 2016, were inspired by the architecture of ResNets which include skip-connections to bypass the non-linear transformations, and introduce a new dense connectivity to further improve the information flow, having been particularly successful in wildfire detection applications, as [26] shows.

In practice, the difference between ResNets and DenseNets can be grossly reduced to connection method between layers. In ResNets the inputs to each transformation layer are summed and can be defined as $\boldsymbol{H_l(x_{l-1}) + x_{l-1}}$, where ***H*** represents the non-linear transformation on the ***l***-th layer and ***x*** represents the output of each layer. In the case of DenseNets they are concatenated, which can be represented by the function $\boldsymbol{H_l(x_0, x_1, \ldots, x_{l-1})}$, where each transformation layer gets all previous concatenated outputs as inputs. This seemingly subtle architectural difference produces starkly different behaviours, as densely connected layers can receive additional implicit deep supervision through the propagation of feature maps, reducing feature redundancy, as well as feature reuse where features extracted by earlier layers are directly used throughout the dense block.

Counter-intuitively, the densely connected architecture of DenseNets do not imply a significant increase in parameters, amounting to only 7,087,061 total parameters. A diagram of the layer configuration in DenseNet can be observed in Figure 19.

Figure 19 - DenseNet layer diagram

4.1.2. Results Interpretation

In this section the results of the initial approach with transfer learning will be presented along with the implementation of the training and testing processes.

The previously identified 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.

The following algorithm 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, $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, and $\boldsymbol{\gamma}$ stand for the training, validation, and test sets, respectively, $\boldsymbol{\mu}$ and $\boldsymbol{\eta}$ represent the compiled model and the trained model, while $\boldsymbol{\pi}$ represents the predicted classes returned. The final classification report method will output a list of evaluation metrics that will be referenced for interpretation, these being Accuracy rate (***A***), Precision (***P***), Recall (***R***), F1-Score (***F1***), True Positives (***TP***), True Negatives (***TN***), False Positives (***FP***), False Negatives (***FN***), and False Alarm Rate (***FAR***).

ALGORITHM 4.1. Data pre-processing, training, and testing using the identified models

Input: Dataset directory ***D***, Model ***M***, class type ***C***

1. [***α***, ***β***, ***γ***] = Pre-process Images (***D*** + [train, val, test], target size = (224, 224), rescale ratio = 1./255, batch size = 128, ***C***)
2. ***μ*** = Compile Model (***M***, initial weights = 'imagenet', optimizer = Adam, ***C***)
   if ***C*** = binary
   then classification nodes = 1, activation = sigmoid, loss function = binary cross entropy
   else if ***C*** = multiclass
   then classification nodes = 2, activation = softmax, loss function = categorical cross entropy
3. ***η*** = Train Model (***μ***, training data = ***α***, validation data = ***β***, epochs)
4. ***π*** = Predict Classes (***η***, test data = ***γ***)
5. [***A***, ***P***, ***R***, ***F1***, ***TP***, ***TN***, ***FP***, ***FN***, ***FAR***] = Classification Report (***π***, ***γ***)

The evaluation metrics returned are defined by the equations that follow, where True Positives are predicted True Alarms that are real True Alarms, False Positives are predicted True Alarms that are actual False Alarms, True Negatives are correctly predicted False Alarms, and False Negatives are True Alarms incorrectly classified as False Alarms. Accuracy can be defined as the total fraction of correctly classified labels within the complete test set, while Precision represents the fraction of predicted True Alarms that are actual True Alarms, thereby indicating the quality of the model's predictions, while Recall identifies the fraction of predicted True Alarms as opposed to the total number of True Alarms within the test set, therefore demonstrating the model's sensitivity to true alarms. F1 Scores combine Precision and Recall in a single metric to allow for a more direct evaluation of the trade-off between prediction quality and sensitivity, while False Alarm Rate highlights the percentage of incorrectly predicted True Alarms, which can also be represented as $1 - P$.

$$A = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.1}$$

$$P = \frac{TP}{TP + FP} \tag{4.2}$$

$$R = \frac{TP}{TP + FN} \tag{4.3}$$

$$F1 = \frac{2 \times P \times R}{P + R} \tag{4.4}$$

$$FAR = \frac{FP}{TP + FP} \tag{4.5}$$

4.1.2.1. Dataset Selection

Table 6 presents a comparison of each model-dataset pairing in terms of accuracy and false alarm rates. The best performing model for each dataset 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. These results can be further interpreted more clearly in the heatmaps in Figure 20 which highlight the best performing pairs.

The binary bounding box strategies, with and without data augmentation, produced substantial better results, where Xception and DenseNet returned their highest accuracy rates on dataset *two-classes-bbox*, and VGG16 having its best result with dataset *two-classes-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 lowest scores.

Table 6 - Comparison of evaluation metrics over each dataset test set across applied models

Dataset	Metric	**VGG16**	**Xception**	**MobileNetV2**	**DenseNet**
two-classes	Accuracy (%)	94.5	97.1	97.4	**98.1**
	False Alarm Rate (%)	7.45	2.75	2.17	1.20
two-classes-aug	Accuracy (%)	94.8	97.2	97.4	**97.9**
	False Alarm Rate (%)	5.25	2.18	1.98	1.00
two-classes-bbox	Accuracy (%)	97.0	**99.2**	**99.2**	**99.3**
	False Alarm Rate (%)	1.81	0.98	0.40	0.79
two-classes-bbox-aug	Accuracy (%)	**97.9**	98.8	**99.2**	**99.2**
	False Alarm Rate (%)	2.34	0.80	0.40	0.40
three-classes	Accuracy (%)	93.1	95.2	95.6	**96.2**
	False Alarm Rate (%)	6.21	3.50	3.31	1.20
three-classes-aug	Accuracy (%)	93.9	95.0	95.7	**96.2**
	False Alarm Rate (%)	4.41	2.75	2.18	1.39
three-classes-bbox	Accuracy (%)	93.0	96.2	97.6	**97.8**
	False Alarm Rate (%)	2.53	0.79	0.20	0.78
three-classes-bbox-aug	Accuracy (%)	90.9	96.2	97.6	**97.8**
	False Alarm Rate (%)	7.52	0.99	0.79	1.36

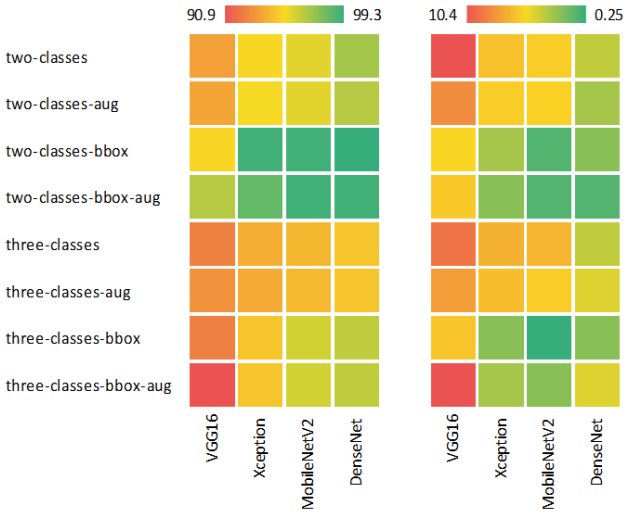

Figure 20 - Left) Accuracy rates comparison heatmap, Right) False Alarm Rates comparison heatmap

In terms of data preparation strategies, binary classes revealed considerably more suitable for this specific application, as with the multi-class approach models are forced to learn features that not only can be indicative of the presence of smoke but are also able to distinguish between false alarm types, which seemingly detracts from the model's ability to correctly identify the main target label. The Pearson correlation between the improvement in model accuracy and the multi-class strategy indicates a strong negative correlation with an R-Score of -0.7355, which is statistically significant for a P-Value of <.00001 at a significance level of 0.05.

The utilization of the extracted bounding boxes, on the contrary, are shown to be the better method when compared to model performance on entire image datasets. The removal of 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 Pearson correlation of R=0.5889 and a P-Value of 0.000391 (<.05).

In the case of augmented datasets, the strategy's effect is less conclusive as some models improved on these datasets, while others had worse or comparable performances. This can also be attributed to the low increase of samples in the minority label with augmentation, which may not have been sufficient to impact the label imbalance. The Pearson correlation of R=-0.0639 indicates a non-significant slight negative correlation, with a P-Value of 0.731942 (>.05).

Table 7 - Statistical Significance of Correlation between strategy and accuracy improvement

Strategy	R-Score	P-Value	Significant for p<.05
Multiclass - Binary	-0.7355	<.00001	Yes
Bounding Box - Entire Image	0.5889	0.000391	Yes
Augmented Data - Original	-0.0639	0.731942	No

Taking into consideration the analysis produced above on the effects of the implemented data preparation strategies, the following model selection analysis and further framework implementations will be applied over the *two-classes-bbox* dataset, as it shows to be more conducive to better model performance.

4.1.2.2. Model Selection

Examining the performance metrics of the various models applied over dataset *two-classes-bbox*, displayed in Table 8, it is 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-Scores present a single-value metric to combine both perceptions, and DenseNet achieves the highest score of 99.4%, signifying a better decision balance.

Table 8 - Comparison of evaluation metrics over dataset two-classes-bbox

Model	A (%)	P (%)	R (%)	F1 (%)	FAR (%)	AUROC
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**

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 abovementioned metrics were calculated over classifications generated using a cut-off value of 0.5, meaning prediction probabilities above or equal to 0.5 were labelled as True Alarms, and below 0.5 labelled 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) – also designated as Recall or Sensitivity – over the False Positives Rate (FPR) – also defined as *1 – Specificity* –, across various cut-off points or thresholds. This relationship can be better summarized in a single-value metric using the Area Under the Receiver Operating Characteristic (AUROC), represented on equation (4.6), which produces a value between 0 and 1. An AUROC 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, True Positives Rate is always 100%, and False Positives Rate is always 0%. An AUROC close to 1 reveals a very good ability to distinguish between classes, being a very important metric to evaluate.

$$AUROC = \int_0^1 TPR(FPR^{-1}(x))\, dx \tag{4.6}$$

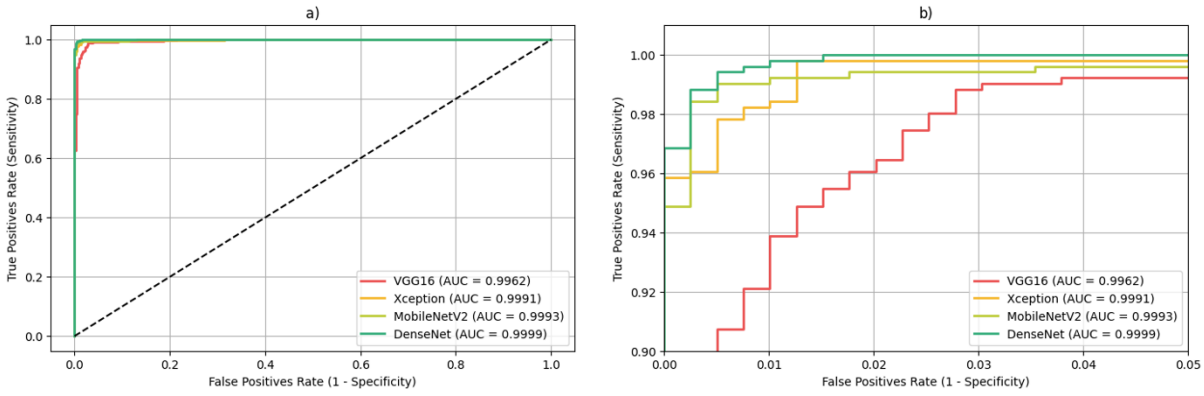

Figure 21 - a) ROC Curves, b) ROC Curves Detail with TPR between 0.9 - 1.0 and FPR between 0.0 - 0.05

The ROC curves displayed in Figure 21 show that Xception, MobileNetV2, and DenseNet all have a very high degree of discriminative ability between classes, as these can achieve very high True Positive Rates without compromising on False Positive Rates. The previously identified higher Recall scores for Xception and DenseNet in particular, match the observed curves, as both models can surpass 0.99 TPR maintaining FPR below 0.05, which is not the case with MobileNetV2.

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. For these reasons, DenseNet can be considered the best fitting model as it consistently displays a 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 AUROC score.

4.2. Spatial and Channel Attention Modularized Selective CNN

While the advantages of transfer learning have been previously explored, a key aspect of this method is the implementation of tendentially 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 dataset *two-classes-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.

4.2.1. Network Architecture

The Spatial and Channel Attention Modularized Selective CNN (SCAM-SCNN) was inspired by the architecture of SBNN [12], and the Convolutional Block Attention Module (CBAM) proposed in [15]. The network architecture is composed 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 portrayed in Table 9.

Table 9 - Layers structure and network parameters of SCAM-SCNN

Layer	Type	Network 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

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 by the following formula, where ***A*** and ***B*** represent the input and output tensors, respectively, and ***k*** the filter kernel.

$$\boldsymbol{B}_{ij} = (\boldsymbol{A} * \boldsymbol{k})_{ij} = \sum_{f=0}^{n_k-1} \sum_{h=0}^{n_k-1} \boldsymbol{A}_{i+f,j+h} \boldsymbol{k}_{i+f,j+h} \quad (4.7)$$

The resulting feature map ***B*** depends on the parameters of the applied filter kernel, and these can generate outputs that highlight specific features of the input, such as edges, textures, shapes, etc. A common edge detection filter kernel is the Sobel operator which can be represented as its horizontal edge detection kernel (4.8) and vertical edge detection kernel (4.9). An example of the outputs of these transformations is shown in Figure 22, and demonstrates how optimizing such parameters result in the extraction of features that are informative in the context of target identification.

$$\begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix} \tag{4.8}$$

$$\begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} \tag{4.9}$$

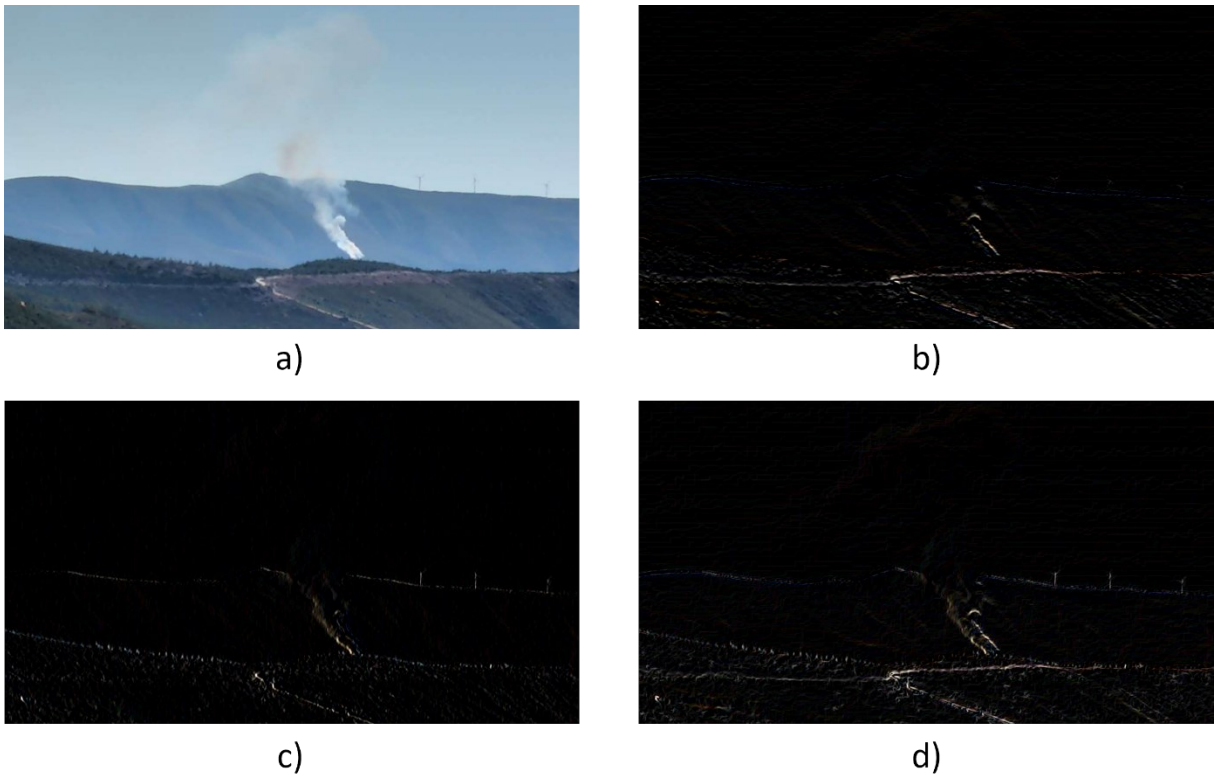

Figure 22 - Convolutions with the Sobel operator. a) Original image, b) horizontal edge detection, c) vertical edge detection, d) sum of horizontal and vertical edge detection outputs

With a normal convolution, the output tensor ***B*** shrinks after each operation, with its dimensions represented as $n_{\boldsymbol{B}} \times n_{\boldsymbol{B}}$ where $n_{\boldsymbol{B}} = (n_{\boldsymbol{A}} - n_{\boldsymbol{k}} + 1)$, and $n_{\boldsymbol{A}}$, $n_{\boldsymbol{k}}$ represent each dimension of input ***A*** and kernel ***k***. To preserve the original size of the input tensor, SCAM-SCNN utilizes *Same* padding in each convolutional layer, which applies a border of size $p = 1$ around the input image, and results in an output dimension $n_{\boldsymbol{B}} = (n_{\boldsymbol{A}} - n_{\boldsymbol{k}} + 2p + 1)$.

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 [36]. 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 ***B*** can be redefined as follows, where ***s*** represents the size of the stride.

$$n_B = \left\lfloor \frac{n_A - n_k + 2p}{s} - 1 \right\rfloor \tag{4.10}$$

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 [37]. This normalization step is defined as follows, where x^* represents the new value of a single component, $E(x)$ is its mean within a batch, $Var(x)$ is its variance within a batch, γ is a learned scaling factor, ε is a small constant, and β is a learned offset factor.

$$x^* = \gamma \cdot \frac{x - E(x)}{\sqrt{Var(x) + \varepsilon}} + \beta \tag{4.11}$$

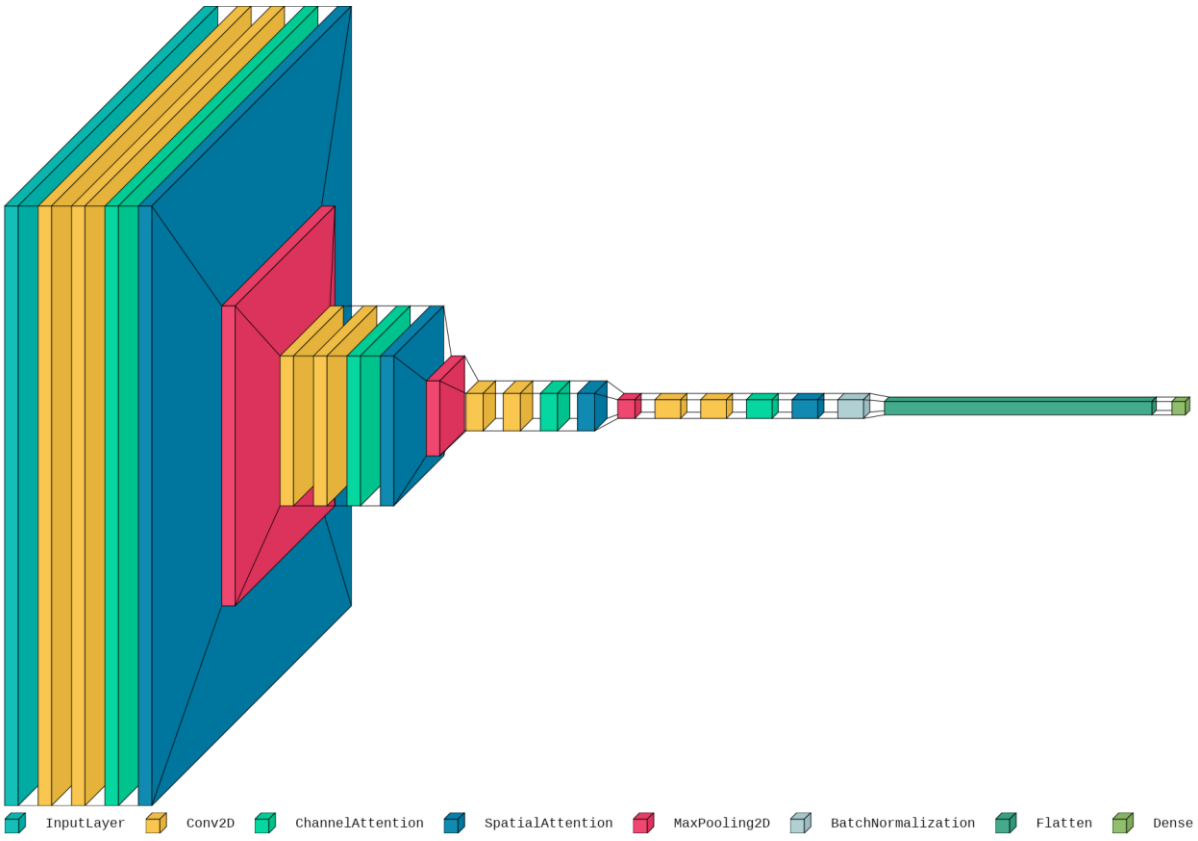

Figure 23 - SCAM-SCNN layers diagram

4.2.1.1. Spatial and Channel Attention Modules

Attention mechanisms have been increasingly studied as a method to improve performance of CNNs, attempting to approximate the role of attention in human perception. In the case of identifying wildfire smoke objects, this can be noted where humans might pay attention to certain colours 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 [15], composed by CAM and SAM sequentially, as shown in Figure 24.

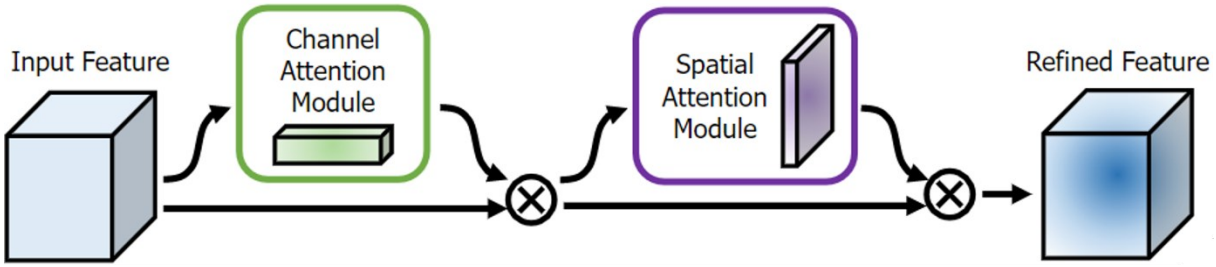

Figure 24 - Overview of CBAM [15]

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 $\boldsymbol{F^c_{max}}$ and $\boldsymbol{F^c_{avg}}$ 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 $\boldsymbol{f}$, while in the hidden layer these are set by a parameter ratio as $\lfloor \boldsymbol{f} / ratio \rfloor$, 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 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 Figure 25, and can be defined by the following equation, where σ represents the sigmoid function.

$$\boldsymbol{M_C} = \sigma\left(MLP\big(MaxPool(\boldsymbol{F})\big) + MLP\big(AvgPool(\boldsymbol{F})\big)\right) \tag{4.12}$$

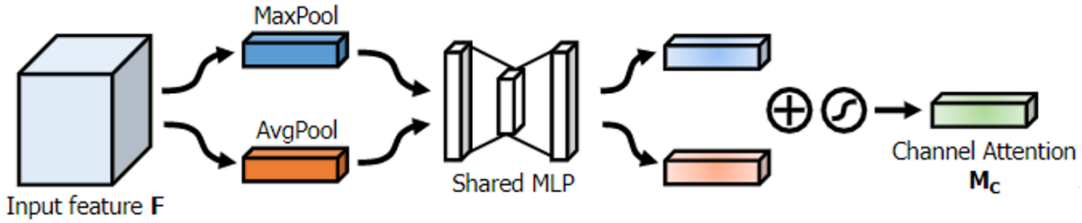

Figure 25 - Channel Attention Module [15]

In 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 $\boldsymbol{F^s_{max}}$ and $\boldsymbol{F^s_{avg}}$ 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 trough 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, that is then multiplied over input ***F***, similarly highlighting the most informative regions of the feature block. This module is characterized in Figure 26, and can be defined by the following equation, where $Conv^{7\times7}$ represents the convolution operation.

$$\boldsymbol{M_S} = \sigma\left(Conv^{7\times7}([MaxPool(\boldsymbol{F});\ AvgPool(\boldsymbol{F})])\right) \tag{4.13}$$

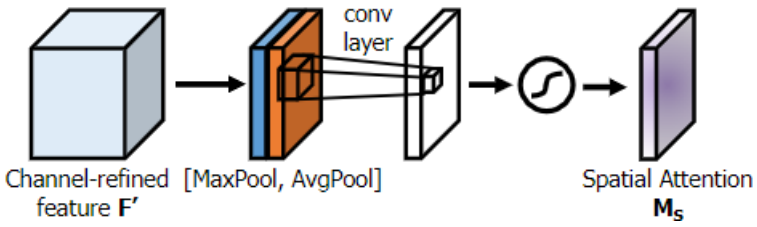

Figure 26 - Spatial Attention Module [15]

4.2.2. Implementation and Results

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

Model compilation and processes followed was done as presented previously during the initial transfer learning approach, applying the pipeline presented in algorithm 4.1 to pre-process, train and test each model.

Table 10 represents the evaluation metrics outputted from the classification report of SCNN and SCAM-SCNN, displaying a noticeable improvement in model performance 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 AUROC 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.

Table 10 - Comparison of evaluation metrics for SCNN and SCAM-SCNN

Model	A (%)	P (%)	R (%)	F1 (%)	FAR (%)	AUROC
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**

In order to visualize network activations, *GradCAM* (Gradient-weighted Class Activation Mapping) [38] 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. Figure 27 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. 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 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.

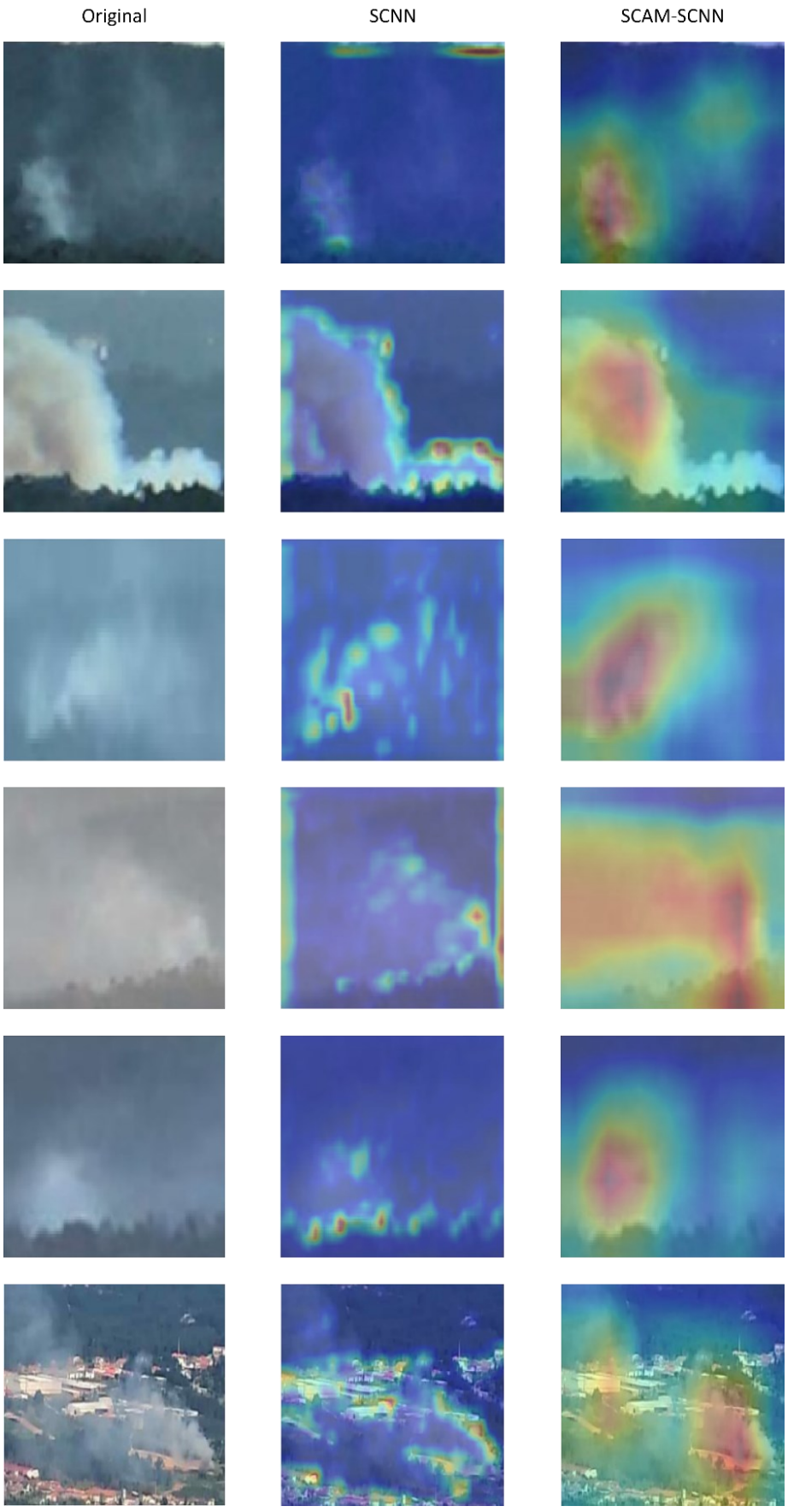

Figure 27 - Comparison of network activations with GradCAM

4.3. 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 $DenseNet_0$ and $SCAM\text{-}SCNN_0$ represent the segment of each network with the fully-connected layers removed. The concatenated features are then passed to a new classification layer as represented by the diagram in Figure 28, and Table 11.

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.

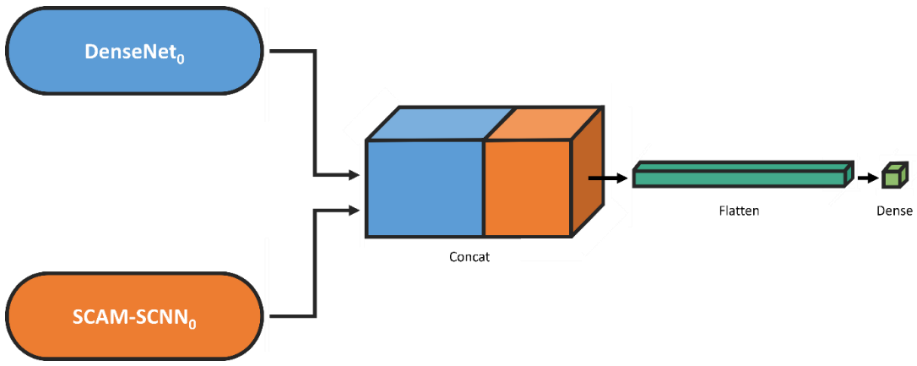

Figure 28 - Diagram the basic structure of the Dual-Channel CNN

Table 11 - Layers structure of the Dual-Channel CNN

Layer	Type	Network Parameters	
L1	Concat	Input: $DenseNet_0 \oplus SCAM\text{-}SCNN_0$	
L2	Output	Neurons number: 1	Activation function: Sigmoid

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 over a sample image input, as depicted by Figure 30.

As the outputs of the first convolution layer still maintain a noticeable resemblance to the original input image shown in Figure 29, 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 information base used in the classification layer to produce better predictions.

Figure 29 - Original bounding box image used as example for feature map visualization

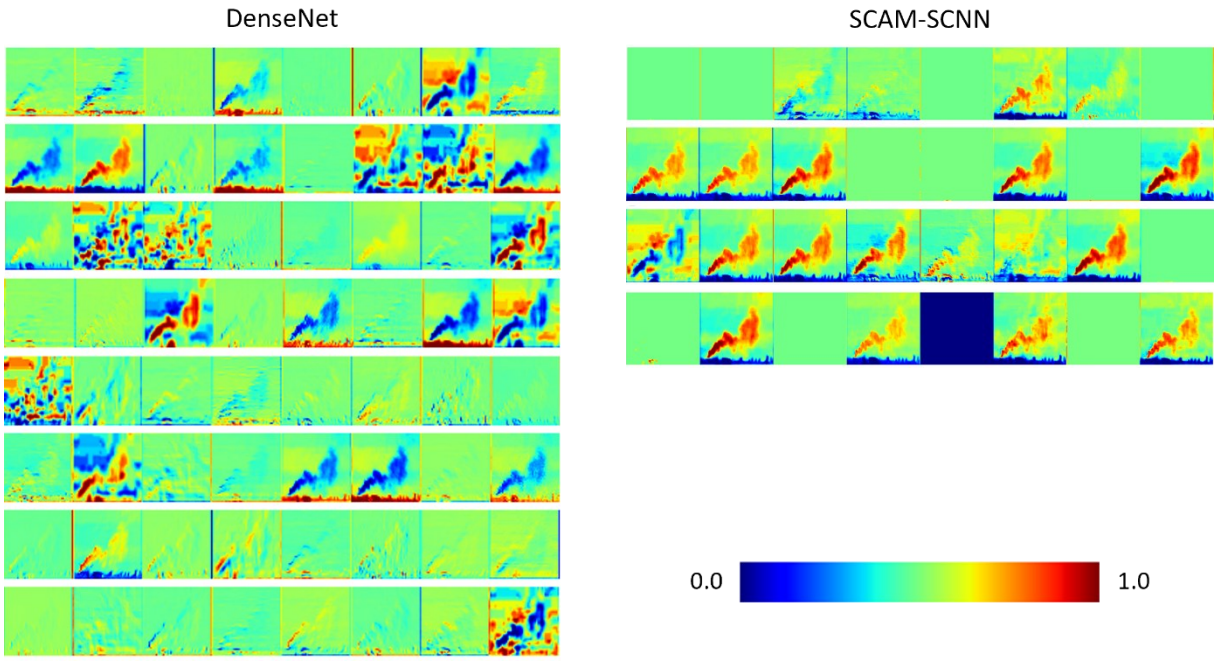

Figure 30 - Feature map visualization of first convolution layer. Left) Output from DenseNet, Right) Output from SCAM-SCNN

4.3.1. Experimental Results

In this section the performance of the Dual-Channel CNN is analysed and compared to that of its branch networks, having similarly applied dataset *two-classes-bbox* in training the classification layer portion of the network, evaluating the results obtained and effectiveness of the dual-channel strategy.

Table 12 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 False Alarm Rate 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 AUROC matches the highest score of .9999 obtained with DenseNet.

While DenseNet showed particularly good coverage of true alarm samples, with a Recall score of 99.6%, but a slightly 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.

Table 12 - Comparison of evaluation metrics between the Dual-Channel CNN and each branch model

Model	A (%)	P (%)	R (%)	F1 (%)	FAR (%)	AUROC
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
Dual-Channel CNN	**99.7**	**99.8**	**99.6**	**99.7**	**0.20**	**.9999**

4.3.2. Time-based Decision Function Adjustment

As previously evidenced in Figure 10 and Figure 11, false alarms occur within the early morning hours of the day with higher frequency, predominantly between 08:00 and 11:00, while true alarms are much more common between 10:00 and 17:00. As detailed in section 3.1.1, 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, and shadowing effects from the presence of hills and vegetation are caused by a lower altitude of the Sun.

The insights obtained from the metadata analysis performed can be utilized to alter network predictions with time-based conditions in order to increase model reliability. Figure 31 displays the plot of prediction probabilities outputted from the Dual-Channel CNN model across the recorded time of day associated with each image, where red dots represent true alarms, and blue dots represent false alarms. With a linear decision function of *p=0.5*, the model fails to correctly classify 1 false alarm occurring between 08:00 and 09:00, as well as 2 true alarms occurring at 15:00 and 16:00.

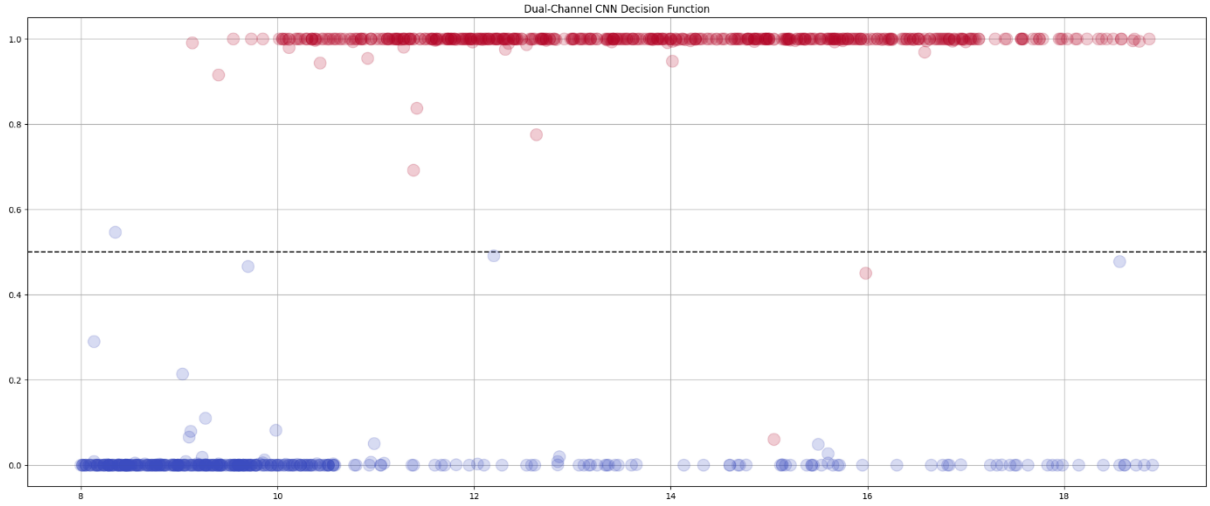

Figure 31 - Plot of prediction probabilities and decision function for the Dual-Channel CNN across hour of day

Considering the previously stated findings pertaining to the frequency of false alarms in morning hours, the sensitivity to true alarms can be reduced by increasing the cut-off value of the decision function during said time window in order to minimize the rate of false alarms, improving the prediction quality of the model.

Figure 32 displays the time-adjusted decision function applied over the prediction probabilities output of the proposed model, which can be defined by the following equation, where x represents the hour of day.

$$f(x) = \begin{cases} 0.6, & x \leq 10.0 \\ 0.5, & x > 10.0 \end{cases} \tag{4.14}$$

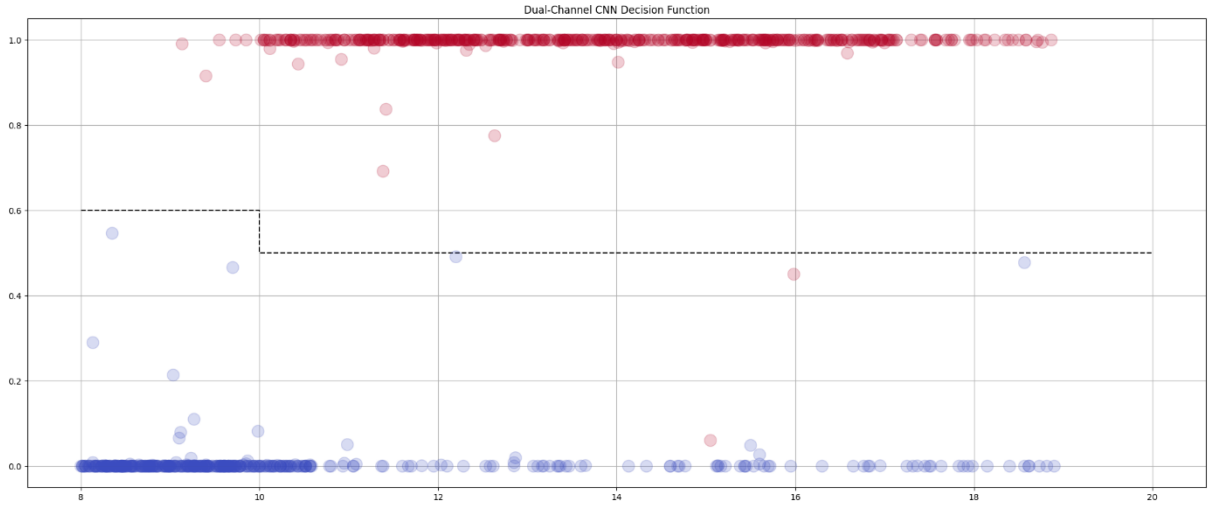

Figure 32 - Plot of prediction probabilities for the Dual-Channel CNN with Time-adjusted decision function

Applying a time-based condition to the Decision function (DF) can improve model robustness as an exogenous variable that 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 13 presents a comparison of the performance metrics of the proposed Dual-Channel CNN with time adjusted decision function to a linear decision function, as well as the branch models. 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 Accuracy rate was further improved to 99.8%, while False Alarms represented 0.00% of predictions.

Table 13 - Comparison of time-adjusted Decision Function to Dual-Channel CNN performance

Model	A (%)	P (%)	R (%)	F1 (%)	FAR (%)	AUROC
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
Dual-Channel CNN with linear DF	99.7	99.8	99.6	99.7	0.20	.9999
Dual-Channel CNN with time-adjusted DF	**99.8**	**100.0**	**99.6**	**99.8**	**0.00**	**.9999**

4.4. Discussion

The detection framework followed throughout this chapter evidenced several aspects pertaining to the problem of wildfire smoke detection defined in this dissertation, where many insights can be extracted, contributing to an increase in knowledge and information, described in this section.

In the initial transfer learning approach presented in 4.1, a set of state-of-the-art architectures composed of VGG16, Xception, MobileNetV2, and DenseNet, are applied to several datasets that employed distinct data preparation strategies, focusing on multi-class labelling, as opposed to a binary class scheme; application of bounding boxes opposite to full image inputs; and the usage of data augmentation techniques to reduce class imbalances.

Figure 33 presents the Accuracy rates obtained from the cross-evaluation of each model with every dataset, where a clear predominance of DenseNet can be observed throughout all data preparation strategies.

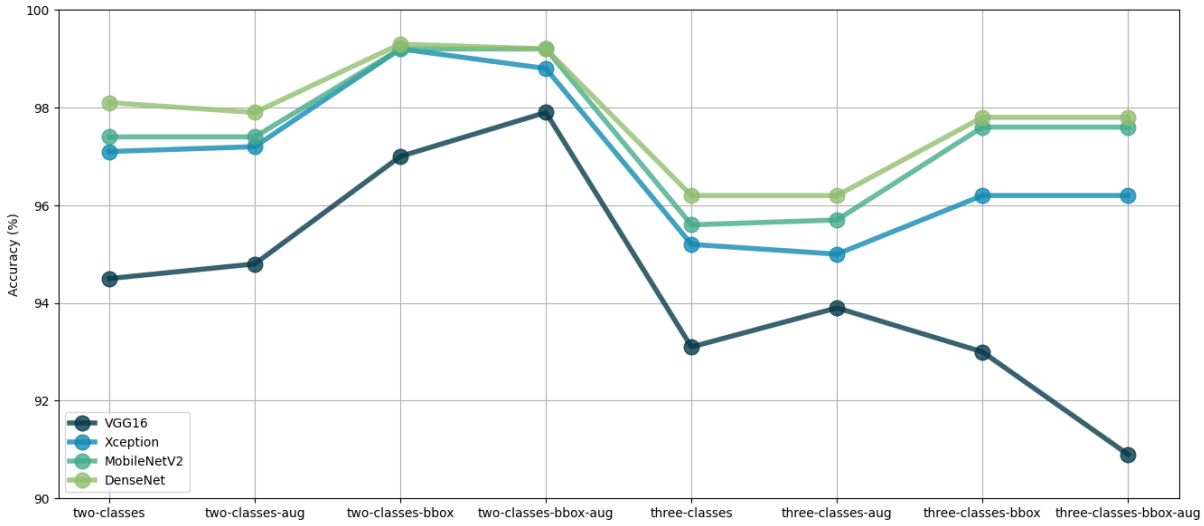

Figure 33 - Accuracy rates obtained with each model across different data preparation strategies

The results obtained evidenced that the usage of bounding box images and a binary class strategy returned better outcomes in the task of wildfire smoke detection, where the impact of data augmentation techniques was less conclusive, as it becomes less relevant with binary classification as the class imbalance is slight.

While employing multi-class labelling to distinguish between the different types of false alarms could present benefits, as these conform to distinct visual features, the observations made show that models performed significantly better when only classifying between true and false alarms. Binary classification allows models to focus on the features of the target label, where the three-class scheme required the model to learn the distinct features of all alarm types.

The superiority of bounding box inputs over full images corroborates the findings identified in the literature review, where several approaches employed some form of a process to extract suspect regions within the image as a mechanism to improve performance. Such a strategy comes to great benefit as the identification of the suspected region can be adopted from the current rule-based detection system employed with *CICLOPE*. The reduction of contextual visual noise is particularly advantageous as the original images found in the dataset cover large forest areas where several objects such as vegetation, hills, and surrounding cloud formations, are present, hindering the models' ability to detect the actual suspected object.

In terms of the application of state-of-the-art networks with transfer learning, the results obtained represent satisfactory outcomes achieved with simple direct implementations. Across the various models, DenseNet consistently maintained the highest detection rates for every dataset, with Accuracy scores as high as 99.3% with dataset *two-classes-bbox*, followed by MobileNetV2 with 99.2%.

Whereas transfer learning-based models offer very satisfactory outcomes with little overhead development, extracting comprehensive features that are more generic and applicable in several different problems, training a novel network can also be beneficial, particularly in very specific use cases such as wildfire smoke detection, where the target label is very consistent and well-defined.

SCAM-SCNN is a selective network fully trained on dataset *two-classes-bbox* exclusively, thus being focused on the extraction of detailed wildfire smoke features, employing spatial and channel attention modules.

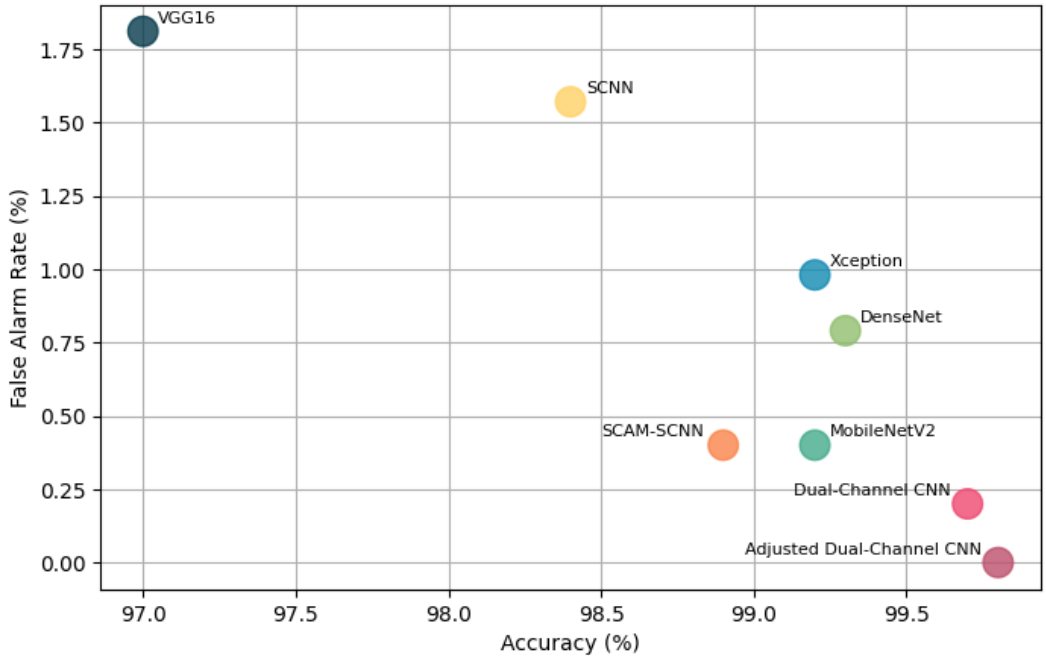

Figure 34 - Comparison of Accuracy and False Alarm Rates across all models presented

Figure 34 shows the combination of Accuracy and False Alarm Rates for all models applied over dataset *two-classes-bbox*. A remarkable increase in model performance is noted between the SCNN network to SCAM-SCNN, where attention mechanisms are applied, providing a strong case for the advantages of spatial and channel attention modules to increase detection ability and enhance the most informative features of the target label, reaching an Accuracy score of 98.9% and approximating the performance of state-of-the-art models.

The different approaches between transfer learning-based networks such as DenseNet, and novel selective networks such as SCAM-SCNN, is explored in the proposed Dual-Channel CNN. References [11] and [12] reported significant advantages to dual-channel architectures, particularly when utilizing networks oriented to distinct visual features. The proposed Dual-Channel CNN combines the comprehensive features of DenseNet with the selective detailed smoke features extracted with SCAM-SCNN, resulting in a more robust capable model, and surpassing the performance of each individual branch model, with an Accuracy of 99.7% and False Alarm Rate of 0.20%, further evidencing the benefits of this implementation when applied to wildfire smoke detection.

As identified in the metadata analysis section of this dissertation, the introduction of exogenous variables such as the recorded time of day associated with an image can also correlate to the frequency of true and false alarms. This is exploited with the Adjusted Dual-Channel CNN by re-defining the decision function as a function of time, where images collected between 08:00 and 11:00 reflect the majority of false alarms, presumably associated to weather and climatic events such as the occurrence of morning fogs, low clouds, and a shift in the Sun's altitude from the horizon, resulting in stronger shadowing events. The results obtained show that the introduction of said variables can improve prediction quality and overall detection ability, further reducing False Alarm Rates.

CHAPTER 5

Conclusion

5.1. Main Achievements

The work developed in this dissertation answers the initially posed research question – "Is it possible to improve the overall accuracy and to reduce the false alarm rate of the CICLOPE 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.

To answer the research question, the following dissertation objectives were defined:

The first objective – "To advance knowledge on Deep Learning solutions for image-based wildfire detection" – was achieved as the state-of-the-art review performed highlighted several implementations that evidenced satisfactory results for wildfire detection. Additionally, the experimentations with several data preparation strategies extracted valuable insights on the effects of binary and multi-class labelling, full image and bounding box image inputs, as well as data augmentation techniques, highlighting important considerations on its impact in model performance. On the perspective of modelling strategies, the comparisons between transfer learning approaches and fully-trained models from scratch also demonstrated the advantages and drawbacks of each implementation, primarily on its impact in feature extractions and performance implications.

The second defined objective – "To analyse the feasibility and applicability of a Deep Learning solution that can integrate with the CICLOPE wildfire surveillance system" – was achieved by studying the system and data constraints of the CICLOPE use case and identifying and testing potential integrations with Deep Learning models and its impact in improving the overall system. One key aspect of this solution is the integration of a rule-based detection algorithm with a posterior Deep Learning model. The literature review process identified several implementations of a similar rule-based process to extract suspect smoke regions as a mechanism 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 model specificity identified in the CICLOPE detection system, the outputs of the several models tested show that the application of Deep Learning models over the universe of detected alarms can act as a secondary filtering stage to reduce the number of false alarms without compromising true alarms recall, thus the solution is not only applicable, but also feasible as it does not disrupt the current detection system by being employed as an attached module.

The third objective – "To develop a Proof of Concept (POC) based on Deep Learning architectures for wildfire detection in the scope of the CICLOPE system, that can be deployed by INOV" – was accomplished with the proposed Dual-Channel CNN presented in Chapter 4. It complies with the previously defined objectives and respective findings as it can be integrated with the CICLOPE automatic detection system by taking any suspected alarm images and its respective bounding box coordinates as inputs and perform a secondary confirmation without impacting the original algorithm, representing a prototypical solution for the addition of a Deep Learning model to the CICLOPE system.

The final objective defined – "To evaluate the performance of the proposed Proof of Concept and its potential to improve overall wildfire detection rates and reduce False Alarm rates" – was also reached through extensive experimentation and analysis of performance metrics. The key measures of success for the applicability of the proposed solution were the reduction of False Alarms to improve model specificity whilst retaining True Alarm recall, improving overall accuracy. The proposed solution is deemed to be suitable for the business requirements, reporting a test accuracy of 99.7% while maintaining a very low False Alarm rate of 0.20%, without time-based decision function adjustments.

Additionally, the work presented in this dissertation serves as the basis for a scientific paper of the same title submitted to *IEEE Access* for publishing.

5.2. Future Work

Despite the positive results obtained with the proposed solution, further explorations could prove beneficial to an even better performance, as well as concerns with production deployment and application in a live environment. This section highlights a few aspects that can be explored in future iterations of the work developed in this dissertation.

As the training and testing of models were done over a finite subset of the data collected from the CICLOPE system, the question about the generalization to live smoke detection in production arises. In particular, concerns with common production machine learning issues such as concept drift, which pertains to an external change in the definition of the target label, and data drift, which pertains to an internal change in the data, should be taken into account when preparing a model to a production environment.

In order to successfully perform this transition, it is important to start by deploying the model in shadow mode, running in parallel with the current system, but not being used to make the final decisions during this stage. This deployment would entail building a monitorization pipeline where model performance can be examined in a live setting in order to inspect the occurrence of the frequent issues mentioned and determine the model's ability to generalize over a live feed of inputs.

On a secondary deployment stage, deploying the model as an assistant to the current system could be a beneficial way of introducing direct intervention without interfering with the original system, by way of having the model output displayed as a suggestive classification of the original alarm without entirely erasing the identified false alarms. Such a deployment pattern would enable building confidence during an intermediary integration stage. Finally, if the model provides satisfactory outputs, deployment to a fully automated stage can be followed, completing the integration of the solution within the CICLOPE automatic detection system.

Another exploration that should be taken into account is the expansion of the model to include other feature sources, such as infrared imagery and exogenous variables such as weather data, both of which are within the abilities of the Remote Acquisition Tower network. Just as image metadata analysis displayed a correlation between time of day and the presence of false alarms, studying the impact of exogenous variables can be beneficial to increasing the system's ability to adapt to different data contexts.

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," 2020. ISBN:978-92-76-18182-8.

[2] "Encyclopedia of Wildfires and Wildland-Urban Interface (WUI) Fires," *Encyclopedia of Wildfires and Wildland-Urban Interface (WUI) Fires*, 2020, doi: 10.1007/978-3-319-51727-8.

[3] P. Kourtz, "The Need for Improved Forest Fire Detection," *https://doi.org/10.5558/tfc63272-4*, vol. 63, no. 4, pp. 272–277, Aug. 2011, doi: 10.5558/TFC63272-4.

[4] J. Brownlee, *Deep Learning for Computer Vision - Image Classification, Object Detection and Face Recognition*, V1.4. 2019.

[5] H. H. Aghdan and E. J. Heravi, *Guide to Convolutional Neural Networks - A Practical Application to Traffic-Sign Detection and Classification*. Tarragona, Spain: Springer, 2017. doi: 10.1007/978-3-319-57550-6.

[6] C. Shorten and T. M. Khoshgoftaar, "A survey on Image Data Augmentation for Deep Learning," 2019, doi: 10.1186/s40537-019-0197-0.

[7] U. Michelucci, *Advanced Applied Deep Learning - Convolutional Neural Networks and Object Detection*. Dübendorf, Switzerland: Apress, 2019. ISBN:978-1-4842-4976-5.

[8] 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 Processing*, vol. 190, 2022, doi: 10.1016/j.sigpro.2021.108309.

[9] 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, 2019, doi: 10.1109/JIOT.2019.2896120.

[10] 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, 2021, doi: 10.1007/s10694-020-01064-z.

[11] 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," *Multimed Tools Appl*, vol. 79, no. 45–46, pp. 34587–34603, 2020, doi: 10.1007/s11042-019-08551-8.

[12] 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, 2020, doi: 10.1109/TMM.2019.2929009.

[13] 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.

[14] Y. Valikhujaev, A. Abdusalomov, and Y. Im Cho, “Automatic fire and smoke detection method for surveillance systems based on dilated cnns,” *Atmosphere (Basel)*, vol. 11, no. 11, 2020, doi: 10.3390/atmos11111241.

[15] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, “CBAM: Convolutional Block Attention Module”, doi: 10.48550/arXiv.1807.06521.

[16] 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 (Basel)*, vol. 11, no. 14, 2019, doi: 10.3390/rs11141702.

[17] 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 *17th International Conference on Machine Learning and Cybernetics, ICMLC 2018*, 2018, vol. 1, pp. 184–189. doi: 10.1109/ICMLC.2018.8527037.

[18] W. Cai, C. Wang, H. Huang, and T. Wang, “A Real-Time Smoke Detection Model Based on YOLO-SMOKE Algorithm,” 2020. doi: 10.1109/CSRSWTC50769.2020.9372453.

[19] 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*, 2022, doi: 10.1007/s10694-021-01199-7.

[20] G. Wang, J. Li, Y. Zheng, Q. Long, and W. Gu, “Forest smoke detection based on deep learning and background modeling,” in *2020 IEEE International Conference on Power, Intelligent Computing and Systems, ICPICS 2020*, 2020, pp. 112–116. doi: 10.1109/ICPICS50287.2020.9202287.

[21] 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, pp. 1–17, 2021, doi: 10.3390/f12020217.

[22] 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 (Switzerland)*, vol. 10, no. 21, 2021, doi: 10.3390/electronics10212675.

[23] Y. Li, A. Wu, N. Dong, J. Han, and Z. Lu, “Smoke recognition based on deep transfer learning and lightweight network,” in *38th Chinese Control Conference, CCC 2019*, 2019, vol. 2019-July, pp. 8617–8621. doi: 10.23919/ChiCC.2019.8865302.

[24] H. Wu, H. Li, A. Shamsoshoara, A. Razi, and F. Afghah, “Transfer Learning for Wildfire Identification in UAV Imagery,” 2020. doi: 10.1109/CISS48834.2020.1570617429.

[25] 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 *2017 8th International*

Conference on Fire Science and Fire Protection Engineering, ICFSFPE 2017, 2018, vol. 211, pp. 441–446. doi: 10.1016/j.proeng.2017.12.034.

[26] 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 (Basel)*, vol. 12, no. 22, pp. 1–16, 2020, doi: 10.3390/rs12223715.

[27] Y. Luo, L. Zhao, P. Liu, and D. Huang, "Fire smoke detection algorithm based on motion characteristic and convolutional neural networks," *Multimed Tools Appl*, vol. 77, no. 12, pp. 15075–15092, 2018, doi: 10.1007/s11042-017-5090-2.

[28] 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, 2021, doi: 10.1007/s11554-021-01094-y.

[29] D.-K. Kwak and J.-K. Ryu, "A Study on the Dynamic Image-Based Dark Channel Prior and Smoke Detection Using Deep Learning," *Journal of Electrical Engineering and Technology*, vol. 17, no. 1, pp. 581–589, 2022, doi: 10.1007/s42835-021-00880-9.

[30] S. Dutta and S. Ghosh, "Forest Fire Detection Using Combined Architecture of Separable Convolution and Image Processing," in *1st International Conference on Artificial Intelligence and Data Analytics, CAIDA 2021*, 2021, pp. 36–41. doi: 10.1109/CAIDA51941.2021.9425170.

[31] Andrew Ng, *Machine Learning Yearning*. 2018.

[32] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks For Large-Scale Image Recognition," 2015. doi: 10.48550/arXiv.1409.1556.

[33] F. Chollet, "Xception: Deep Learning with Depthwise Separable Convolutions," 2017. doi: 10.48550/arXiv.1610.02357.

[34] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," 2018. doi: 10.48550/arXiv.1801.04381.

[35] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely Connected Convolutional Networks." [Online]. Available: https://github.com/liuzhuang13/DenseNet.

[36] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, "Striving for Simplicity: The All Convolutional Net," 2015. doi: 10.48550/arXiv.1412.6806.

[37] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," 2015. doi: 10.48550/arXiv.1502.03167.

[38] 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, Oct. 2016, doi: 10.1007/s11263-019-01228-7.