Faculty of Sciences and Technology
Department of Informatics Engineering

Intelligent System for Fire Detection

Ana Abrantes de Abreu Madeira

Dissertation in the context of the Master in Informatics Engineering, Specialization in Intelligent Systems advised by Prof. Bernardete Ribeiro, co-advised by Prof. Alberto Cardoso and Prof. Catarina Silva and presented to the Faculty of Sciences and Technology / Department of Informatics Engineering.

January 2020

1 2 9 0

UNIVERSIDADE Đ
COIMBRA

This page is intentionally left blank.

Abstract

The early detection of a fire can largely mitigate its harmful consequences. With the developments in the area of image capture technology and the consequent improvement in image quality, it is now possible to develop systems for visual identification of fire indicators.

The present work aims to develop a fire and smoke recognition system in images captured by smartphone cameras. This system can later be integrated into an application that will allow the reporting of fires using crowdsourced data.

Therefore, different deep learning techniques will be implemented and tested, which will involve training and evaluating the performance of different models. The training of these models also involves the analysis and pre-processing of images from the training dataset.

The development of this project is part of the FireLoc project, as a system for validating the presence of fire or smoke in images.

Keywords

Deep Learning, Image Recognition, Convolutional Neural Networks, Transfer Learning

This page is intentionally left blank.

Contents

This page is intentionally left blank.

List of Figures

This page is intentionally left blank.

List of Tables

This page is intentionally left blank.

Chapter 1

Introduction

1.1 Context and Objectives

Identifying a forest fire in its early stages is essential for a faster and more effective response from fire and civil protection authorities. An effective initial response can decrease fire damage and, in certain cases, it can prevent the loss of lives and property, e.g., houses and other belongings.

Even considering the supplementary preventive measures used, such as the increasing number of firefighting means available, in summer, the number of reported occurrences is alarming [5]. Every summer, there are countless fire occurrences, often caused by humans, and aggravated by the typically high temperatures. The severity of the damage has been particularly visible in recent years. In 2019, Portugal had over 41,000 hectares of land affected by fire [6]. In the beginning of 2020, wildfires in Australia caused, in less than a week, the destruction of over 1000 homes and a visible cloud of smoke from space [2, 3].

Fire detection can be very helpful to the population if followed by a period of communication and report of the current situation to the competent authorities.

One of the problems that may affect fire report is the lack of means of locating the person reporting a fire via mobile phone. This situation is particularly worrying when it comes to forest fires where there is no surrounding urban landscape, and it is challenging to use known geographical references. In these cases, someone who wants to report an incident will need to know their exact location to seek assistance. To mitigate this risk, there have recently been more and more initiatives to allow a more accurate location of the call received by the emergency services in Portugal. In 2019, the Government of Portugal announced the implementation of advanced mobile location (AML) technology, which is applied to smartphones, to allow the recognition of calls to the emergency number 112. With this system, it is thus possible to activate geolocation services and automatically send the location coordinates to the 112.pt operational center [1]. However, when locating the person who reports a fire, the location of the ignition may not be immediate, a situation that the FireLoc project aims to resolve.

FireLoc project's primary goal is to develop a system that enables the report of forest fires by identifying, locating, and monitoring them, using crowdsourced data. This system should include a mobile application that should allow its users to send a photograph of the observed fire, as well as its geolocation data. Using this system, any user can submit a photo taken by their smartphone camera. The information sent may also contain the geolocation of the reported event, allowing a faster and more adequate intervention by the firefighters. Due to the possible increase in the number of contributions, when climacteric conditions are more prone to fire ignition, e.g., during summer, the development of a filtering mechanism becomes imperative, allowing for a faster and more accurate communication of the occurrence. With the increase of the volume of information received, processing it by human visual analysis of each reported situation, even if done by an expert, becomes too time-consuming, and therefore not useful for a rapid response. Therefore, the development of an intelligent system that can classify the collected images automatically is proposed. This system can then be applied to each report made using the FireLoc application, accessing if there is a current fire or not and discarding false reports, which may improve firefighter's intervention. This system would enhance the process of analyzing crowdsourced data, and allow specialists to reject false or irrelevant reports, ultimately enhancing firefighter's response.

This work is part of the FireLoc project, and its main objective is the development of a system capable of recognizing fire and its signs in the images submitted.

Like any image classification problem, there are some inherent risks to consider. Of these, it is possible to highlight the large volume of images needed to be able to develop and evaluate the behavior of the developed system, simulating real situations, as one of the biggest obstacles to overcome.

In the context of the FireLoc project, the main goals established regarding this work are:

- Study of methodologies applied to image fire recognition;
- Development of a fire recognition algorithm in images;
- Development of an algorithm for processing and classifying the images collected by the application;
- Development of a system for recognizing forest fires in images, to be used to classify images submitted by the app's users.

1.2 Expected Contributions

While participating in the development of the FireLoc project, the following contributions are expected:

- Documentation of the current state of the art regarding fire detection methodologies;
- Development of a fire detection system to be integrated in the FireLoc system;
- Proposal of a specific image dataset for forest fire recognition;
- Testing of the system developed under real conditions using real fire images and analysis of the results obtained in real-time;
- Publication of the results obtained.

1.3 Document Outline

This document is organized in a total of six chapters.

Firstly, this chapter provides a contextualization of the work developed, as well as its contributions to the FireLoc project.

Then, in the second chapter, state-of-the-art analysis of smoke and fire recognition in images and videos is presented. Concluding this chapter is an overview of current approaches and systems.

The third chapter describes the work methodologies and the approach used for the work developed in the first semester.

Chapter four contains the data used for the work developed as well as the results obtained and their discussion.

Chapter five introduces the planned and actual work plans for the first semester. The schedule for the work to be developed in the second semester is also presented.

Finally, in the sixth and last chapter, a conclusion is given regarding the work developed and the results obtained in the first semester. A summary of the work to be developed in the second semester is presented.

This page is intentionally left blank.

Chapter 2

Background and State of the art

In this chapter, state-of-the-art is addressed with regards to fire recognition in images. Current approaches to similar problems are then critically evaluated and analyzed in how they can be used in the construction of the fire detection system proposed in this thesis. First, several approaches for image and video classification and fire recognition are discussed. Then, an overview of possible learning approaches that have been applied to this area is presented and briefly discussed. Finally, similar systems for fire reports using crowdsourced data are analyzed, to contextualize the work developed, and identify possible difficulties.

2.1 Fire recognition methods

Fire or smoke recognition in images has been of increasing interest and, consequently, new methodologies have been developed, based on different paradigms.

In the following sections, we present some of these methods, distinguishing feature-based methods from methods based on the construction of classification models. A summary is also given, addressing the different methods that can be applied to video classification with respect to the presence of fire or smoke, as well as a distinction between these methods and the ones used for static image classification.

2.1.1 Feature-based approaches

Feature-based recognition relies on the extraction and identification of the most relevant features for image classification. For each methodology, the extracted features are then used to develop an image or video classification algorithm regarding the presence of fire. These correspond to the first documented approaches to visual fire recognition. The following methods use image characteristics such as color or texture to identify the presence of fire flames or smoke in static images or videos.

The problem to be resolved is the recognition of signs of fire in images, referring to the area of image recognition problems. Image recognition can be defined as the process of identifying and detecting an object or feature in an image. Image features can provide rich information on the image content and can also help identify different regions in an image. Correct discovery and analysis of image's most relevant features can help solve image recognition tasks.

One common step in feature-based approaches to this problem is the use of color features to isolate image pixels containing the fire. Some solutions have a pre-processing step with the purpose of eliminating the non-relevant information present in the image by filtering it and processing the colors present. The next phase consists on using color-feature extraction methods or texture identification algorithms [16]. This step includes an edge-detecting analysis to isolate the relevant information, making it possible to ignore background objects [12, 40]. One advantage of these methods is that they have a low computing cost.

In statistical color modeling approaches, background objects are extracted, and different sections of the image are segmented into fire and non-fire regions. This segmentation, combined with motion information, is a well-known algorithm for fire detection in video [13]. Most of these approaches only address fire detection, without concern for the presence of smoke. In the early stages of a fire, smoke detection can be of the utmost importance as it allows for faster and more appropriate response by appropriate means such as firefighters.

Using chromatic and dynamic features, in [15], fire detection was proposed by Celik et al, using a set of rules. The proposed algorithm thus proved the possibility of image classification based on an analysis of chromatic features to identify pixels containing fire or smoke using the RGB color space [15].

In 2008, Celik et al proposed a generic color model for fire pixel detection. The algorithm consisted of identifying pixels corresponding to the fire areas of an image. A set of rules was proposed in the YCbCr color space to detect the presence or absence of fire in a certain pixel [14]. Other color spaces can be used for fire detection, with the RGB color space proving to be less robust against illumination change.

Statistics or probability distribution analysis are also proposed as a color-based fire detection method. Typically, this type of method involves the calculation of probabilities of the images containing fire or not. Based on the constructed probability model, it is then possible to classify the images. For image classification, lookup tables or probabilistic models can be used. When using these tables, the goal will be to map the probabilities of belonging to each class and the color distribution present. It is assumed that pixels of the image in which smoke is present are distinguishable from the others, forming an isolated set regarding color distribution. For the construction of probabilistic models, it is necessary to calculate the probabilities of the images belonging to each class, based on their color distribution (coordinates in the considered color space). In both approaches, a prior analysis of the histogram of the color levels of the image is necessary to construct the table and calculate the probabilities of belonging to each class [26].

Even though these approaches are computationally efficient, color-based approaches have a high probability of failure with the presence of objects with a color similar to fire, red lighting conditions, presenting a high false-alarm rate [12, 21].

The use of features like color or texture for smoke detection presents additional difficulties. Due to the semi-transparent nature of fire in particular images, color and texture features cannot easily be identified. Smoke can often be mistakenly detected with the presence of clouds or snow, raising false alarms and, therefore, most of the algorithms that rely on color features only focus on fire detection. Furthermore, when using video, as there is temporal information, and it is possible to identify the presence of a fire by recognizing scenario changes.

The discovered features can also be used in classic computer vision approaches. Computer vision can be defined as the field that deals with how computers can understand and extract information from images or videos. The use of features for computer vision object detection tasks is, therefore, an essential step in computer vision methodologies. Computer vision algorithms for fire detection usually encompass moving object segmentation, fire pixel detection, and an analysis of the regions containing "fire pixels" detected in the first stages [14]. These approaches will be discussed in the next section, along with other model-based approaches.

2.1.2 Model-based approaches

Following the analysis of fire or smoke recognition methods in still images, a review of model-based approaches is now presented. These are methods assume the use of a model, such as a neural network, which is trained to recognize flames or smoke in images or videos.

The known model-based approaches for fire and smoke recognition can be divided into traditional computer vision methods and deep learning methods. In conventional computer vision approaches, there is a feature extraction phase, previous to the model training phase, that relies on expert knowledge and is the base for the image classification. Deep learning approaches will be addressed in the following sections.

Most traditional methods imply the extraction of color, shape, or texture information and, in video recognition approaches, motion information features. Some studies consider the existence of static and dynamic features [31]. The extraction of dynamic features presupposes an analysis of the temporal sequence of events, i.e., the analysis of scenery changes over time. For static features extraction, it is only necessary to analyze static images or, in case of videos, the independent analysis of the frames.

As static features of flames, color, texture, and shape can be useful for identifying fire regions in an image. As dynamic features, in video, changes in the area affected and the edges of the flames, image shape-changing over time and overall movement can also be considered. Changes in the area affected by the fire are identifiable by calculating the number of highlight points, i.e., points where the brightness is above a certain threshold and the overall movement of the flames can be detected based on the video frame sequences [31].

After the extraction of these features, there can be the definition of a rule-model to identify image regions containing the fire, as presented above. Another alternative is the use of the extracted features for the construction of Feature Vectors, used as input to train the model.

The compilation of the gathered feature information into Feature Vectors is another crucial task that has to be completed before the training phase of the model. These vectors contain information that can help describe the scene present in an image, the present objects, and their most discriminant characteristics.

For these approaches, models such as backpropagation neural networks [31] have also been applied. With the backpropagation algorithm, during the training phase, the neural network's weights are updated according to the total loss and the error each node is responsible for. By backpropagating the total loss, the nodes responsible for higher error rates are given less importance, i.e., a lower weight value. This algorithm will thus allow the training of a model with good classification accuracy, i.e., the classification of images or videos close to the manually assigned classes.

For the improvement of classification accuracy, some strategies include an initial step of extraction of the color feature parameters to exclude classification interferences based on color [16, 31]. Combining the static and dynamic features calculated, a recent study proposes the construction of multidimensional feature vectors. These are then fed to a backpropagation neural network, allowing the recognition of flame [31].

Support Vector Machines or SVMs can also be an alternative for fire recognition. SVM is a machine learning algorithm that can be used in classification problems. The algorithm optimizes a hyperplane to separate the data, considering the problem's classes.

One of the proposed algorithms for fire recognition using SVMs begins with the analysis of color distribution values within the range in the image's color model, RGB for example. Following that, using a predefined threshold value, the image's possible fire regions are identified. This algorithm assumes that the surroundings have an orange or red color and that only fires in a more advanced stage present a white-color in their core and uses them to segment the image into fire and non-fire regions [40]. After obtaining the fire images, to discard false candidates, the identified regions pass through a filtering process based on static features, using a trained support vector machine (SVM). For each defined candidate region, features like color distribution, texture parameter, and shape roundness are extracted and analyzed, discarding possible false positives.

The static features considered include five color distribution features, five texture parameter features, and one shape roundness feature.

As for color distribution features, the ratio of white-yellow pixels, the ratio of red pixels, the ratio of orange pixels, and another two features calculated based on the color histogram of each candidate feature in different color channels.

For the texture features, as in the HSV color space, the H represents color information, the co-occurrence matrix is calculated from that. Texture features such as the angular second moment, the entropy, the mean, the contrast, and the inverse difference moment are then taken into the classification as static features. The co-occurrence matrix is a statistical method used to represent co-occurrent pixel color values. These features are, therefore, helpful in understanding the textures present in the image.

Shape roundness is a boundary analysis feature. It is used as a way to describe the complexity of the shapes present in the candidate region. The use of this feature assumes that more complex shapes will have a higher roundness value. An alternative to the use of this feature, explained under other algorithms presented, would be the calculation of flame edges, for example.

Combined with dynamic features related to changes throughout the frame sequence, these are used for SVM training. This makes it possible to identify a fire in video. Examples of the dynamic features used include Variation of contour and Flickering frequency. The calculation of these features uses the coefficients of the discrete Fourier transform in the various frames of the video. By calculating the variance between two consecutive descriptors, changes in the flames over time can be detected. The use of this feature is based on the fact that when watching a video fire, there are changes in the shape and region of the image representing flame. This distinguishes between fire and fire-colored objects.

For the Flickering frequency calculation, consecutive Fourier descriptors are used over a short time period of the video. Considering these descriptors, the sequence of variances with temporal wavelet is analyzed. This feature allows you to evaluate the behavior of flames over time in the video.

State of the art deep learning approaches imply the use of raw data to train the models and typically use CNN architectures. These approaches and architectures will be addressed in the following sections.

2.1.3 Video-based approaches

Traditional approaches to fire recognition in videos combine a set of static and dynamic features such as color, shape, texture, and motion orientation for fire recognition [39].

Most video-based fire recognition methods combine the temporal behavior of smoke and flames with its color and shape characteristics to recognize fire. With these features' information, it is possible to use or build a rule-based algorithm or a multi-dimensional feature vector. These feature vectors can then be used as input to a conventional classification algorithm such as Neural Networks or Support Vector Machines [17, 40].

Unlike image analysis, by using video footage, it is possible to obtain useful information on the development of the situation, i.e., the size and evolution of the fire. Adding to the static features that can be extracted from still images, in videos, it is possible to extract dynamic features based on temporal information [31]. It is also easier to identify the location and progression of the event in systems that use static cameras.

There are a few advantages in the use of video for early forest fire detection and report, as opposed to only using still images. While satellite imagery cannot be used for early wildfire detection, due to the temporal difference between collection and analysis or the low resolution, solutions based on areal video footage can be seen as a solution [38]. Compared to still images analysis, fire and smoke identification becomes less problematic. Accurate texture analysis to differ fire from similarly-colored objects in images may be challenging. This problem is mitigated in video analysis by using features such as edge, shape, or area changing of the identified area as a possible fire [31].

Although the use of video allows for an easier classification, videos usually present a higher data size, i.e., are more complex, and training of models can be more computationally costly than training for recognition in images. For the current problem, videos recorded with smartphone cameras do not fall under the category of "static cameras", and even more significant difficulties in extracting dynamic features may arise. Besides, detecting changes in scenery over time may not be feasible in shaky, blurred, or poor quality videos, which are very common in videos taken by people in crises.

2.2 Learning approaches

Following the analysis of known approaches to model-based fire and smoke recognition in section 2.1.2, state of the art computer vision approaches, and deep learning approaches will now be addressed.

In traditional or classic computer vision approaches it is possible to identify several classic phases of image classification. The figure 2.1 shows a step-wise division of the image analysis and classification process when using a classification model.

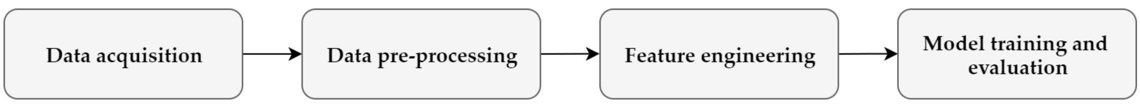

Figure 2.1: Phases of image classification

Data acquisition :

This step involves the acquisition images to be used in training and testing of the model as well as all the dataset preparation necessary. Each image needs to be analyzed and annotated according to its category.

There should be a large enough volume of data for the model to learn. In this problem, it should contain a large enough number of examples of images containing Fire, images containing Smoke and images containing forest areas, considered Neutral.

The acquired data should also be representative of all possible situations. In this problem, it should include images containing clouds as not Smoke or containing sunsets as not Fire, as they may be confused with the presence of Smoke or Fire.

Data pre-processing :

In some cases, an additional pre-processing step may be necessary before the images are given as input to the model. This step ensures that the input provided for training is in the expected format and ready to be used in the next stage.

A good pre-processing of the data can make model training easier and allow better classification accuracy.

Feature engineering:

In this stage, all the relevant features that can be used for classification are obtained from the training set of images. In Traditional Computer Vision approaches, feature extraction includes manual extraction and selection of features.

Model training and evaluation:

With the features extracted in the previous stage, the model is then trained, iterating over the available data. After training the model, metrics like a confusion matrix or accuracy can be used to evaluate its performance.

2.2.1 Convolutional Neural Networks

As an alternative to classic computer vision algorithms for image classification, state of the art algorithms use deep learning methods. One of the advantages of using deep learning methods is that feature extraction is done directly by the model, allowing end-to-end learning, i.e., the model only needs a set of annotated images as an input [27].

With the use of deep learning methods, it is possible to process large amounts of information to train very deep architectures with many layers. The extraction of discriminative features is done automatically. By processing large amounts of data for training, it becomes possible to obtain better generalization ability, and therefore, higher accuracy in the testing phase [28].

By using deep learning models, the patterns present in the data are discovered automatically, and the most descriptive features identified in each class are then used for classification. Convolutional Neural Networks (CNN) are the most successful type of models used for image classification and object detection tasks [27, 39].

CNN can extract topological properties from an image [17], very useful for image segmentation classification problems [18]. An image's topological properties study can provide a mathematical basis for its proper processing. These properties, in image classification problems, can be related to color, shape or texture features, for example.

Since these networks are able to automatically learn a set of visual features from the training data, the results don't rely solely on expert knowledge to build relevant feature extractors. It is, therefore, possible to obtain more accurate classifications [17].

Several approaches to the problem include the study and optimization of a model built and trained from scratch. However, the full training of a model requires the availability of large amounts of annotated data for training, validation, and testing [36]. The training of deep neural networks can take a very long time. However, an additional data pre-processing stage can make use of traditional computer vision techniques to facilitate the training of the model or to make it more robust [27].

As stated before, it is possible to use a CNN operating directly on raw data. Some state-of-the-art proposed algorithms for fire and smoke detection assume the training of the model with a raw RGB frame as input and no previous feature extraction phase. [39] [17]

When using CNN, there are some essential concepts to have in mind:

- **Local Perception :**

  In an image, local pixels have close correlations. Correlation between more distant pixels is not very obvious. Based on these concepts, CNN follows the principle that there is no need to have a global perception of the image by all neurons of the neural network. Each neural node can, therefore, focus on one part of the image and then integrate the information with the others. [28]

- **Filters :**

  Filters are used as feature extractors. By applying a filter in different convolutional areas, it becomes possible to extract the various features of an image.[28]

- **Sub-sampling :**

  Allows the reduction of computational complexity. By applying downsampling, it is possible to identify features that are invariant to translation. [28] Sub-sampling is performed by the network's pooling layers. There are usually several convolution and pooling layers in CNNs.

- **Full connection :**

  At the end of the network, the most correlated features are selected from all the abstract features extracted. The fully connected layer takes as inputs the features extracted by the previous convolution and pooling layers and predicts the correct label, classifying the image.

The figure 2.2 shows an example of a CNN architecture.

Convolution Neural Networks, or CNNs, are multi-layered feedforward deep learning neural networks, with different types of layers, widely used for image analysis.

The name of these networks is derived from the convolution blocks. A convolution operation can be defined as a mathematical operation that expresses what two different sets of information have in common. Each convolution operation, in CNNs, can be seen as filters that, during training, "slide over the input", generating a feature map. The output of the convolutional layers then passes through an activation function. The resulting feature maps, after all the convolution operations, are put together, resulting in the output of the convolutional layer [7].

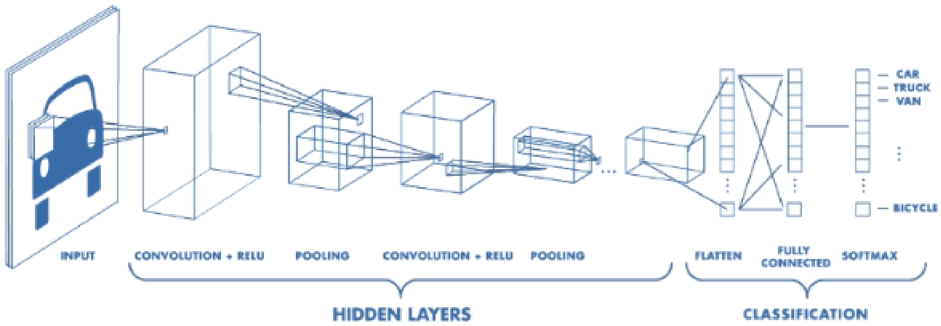

Figure 2.2: An example representation of CNN layers, from [7]

Max and average pooling are also important layers of a CNN. These are responsible for subsampling operations. In these layers, the input is divided into rectangular pooling regions. Then, max-pooling layers compute the maximum of each region, and average-pooling layers compute the average of each region [4].

Following the network's pooling and convolution layers, there usually is a fully connected layer. All of the neurons that compose a fully connected layer are connected to all the neurons in the previous layer. When solving classification problems, the number of neurons in the last fully connected layer corresponds to the number of classes considered. [4]

When choosing a model there are some very distinct architectures that can be considered. Depending on the problem that needs to be solved different paradigms can affect the results obtained, even if using the same training (and validation) data.

Training complex architectures can be a difficult task. Due to the complexity of the networks and the vanishing gradients problem, training a CNN can take a long time and demand a lot of computational capacity [36].

One common strategy to enhance the network's performance is to increase its depth, however the deeper the network the harder it is to train efficiently. With increased depth, optimization becomes more difficult and there is an increased probability of overfitting [22].

Overfitting happens when a function models a particular set of data too well, i.e., in a model, when the training set's noise or random fluctuations are learned by the model and used as a future reference for performing classification. The problem with overfitting is that these characteristics may be particular to the training set and not applicable to all possible cases, leading to a more significant classification error.

Activation functions are a significant part of a neural network's architecture, conditioning its performance and its ability to learn. These functions define the output of the nodes when given a particular input. They are what allow neural networks to comprehend and learn from data. When using deep learning, the effect of adding more layers and using certain activation functions can cause the gradient (error) to increase exponentially, resulting in a very slow training of the network's front layers. Another problem is that CNNs take a long time to train, due to having more training parameters [36].

Residual Learning architecture

As a solution to prevent overfitting, Microsoft has made advances in terms of optimization by using residual learning networks, ResNets. Unlike with other architectures, with these networks, it becomes possible to use deeper networks without compromising test accuracy [22].

Degradations in training deep neural networks are a known problem [34]. As the depth increases, there is a point where the training accuracy starts decreasing.

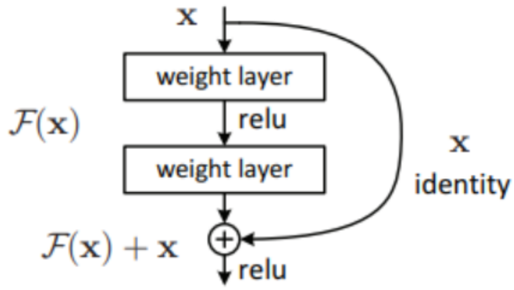

Figure 2.3: “Residual learning: a building block” [22]

In ResNet architectures, instead of only using convolutions, the degradation problem is addressed by using "shortcut connections" or identity mappings to increase network performance, represented in 2.3. The use of shortcuts as identity mapping consists of using residual learning "every few stacked layers". [22] After the training of each block of layers, the rectified linear units (ReLU) activation function is used instead of other less sensitive functions. By using the ReLU function after each block of layers, it is possible to evaluate if the input contains useful information and, therefore, decrease the error in comparison to other plain networks. In this case, when the result is negative after the training of the block, it is possible to consider it 0, avoiding the error propagation. [22][29] This also helps prevent the deterioration of the model accuracy. It also allows a faster and easier training phase [36].

Another common approach to enhance the model's performance is making the training set closer to the situation it will be used in. As the model will be used to identify forest fires, images containing urban settings and other backgrounds may be confusing and not as useful for training.

Another possibility is to use an unbalanced dataset. Some models trained for similar applications, i.e. identification of fire or smoke in images, use an unbalanced dataset for the training of the model. They propose that a higher number of neutral images compared to the number of fire images used can help decrease the probability of false-positive occurrence.

In the same study, it is also proposed that very deep convolutional models for fire recognition tend to overfit very easily. As a solution to that, they propose their own architecture, creating their own CNN architecture and training it from scratch. [24] Although they were able to obtain a good result, to train this network, a considerable amount of data was needed. Training also took a very long time, due to the number of parameters that needed to be optimized and the complex processing operations required.

Since its publication, in 2015, ResNet models have been applied in image classification and object recognition problems, achieving very promising results.

There are several versions of the ResNet, defined by the numbers 18, 34, 50, 101, and 152, with the numbers corresponding to the number of network layers. ResNet 18 and 34 use building blocks similar to the one shown in figure 2.3, two layers deep. The remaining ResNets 50,101 and 152 are made up of 3-layer building blocks.

The most important feature of these architectures, which distinguishes them from all others, is precisely the use of the "shortcut connections" mentioned above, at the end of each of these blocks. Making use of these properties, deep residual learning networks are the ones that can present greater depth without reflecting on their performance, among all the examples of architectures presented.

Residual learning neural networks are, therefore, the best solution for image recognition problems.

The lack of annotated datasets available and the required effort to manually create an annotated dataset, comprehensive enough for efficient training of a convolutional neural network, has led to the search for other solutions and algorithms to get around this problem.

Examples of state-of-the-art architectures, available with pre-trained weights for different known datasets are the residual networks or Resnets (mentioned above, R-CNN, MobileNet, and Vgg models.

Examples of state-of-the-art architectures for CNN are, for example, the above mentioned residual networks or ResNets, Inception, R-CNN, Mobilenet, and Vgg neural networks. Each of them has some singularities that set them apart from other previously used architectures. These are explained below to clarify the advantages and disadvantages of using each of them.

VGG architecture

VGG models were first proposed in 2014 by the Visual Geometry Group (VGG) from the University of Oxford. These architectures stand out from previous ones, mainly due to the concept they use to increase their depth [35].

For the construction of these models, it is assumed that by combining several smaller filters, it is possible to obtain results similar to those resulting from the use of larger filters. Therefore, very small convolutional filters are used as their depth is increased. By maintaining the benefits of using small filters, it is possible to decrease the number of parameters that need to be optimized throughout the workout. In these networks, the ReLU activation function mentioned above is used [35].

There are several versions of the VGG architecture, VGG16, and VGG19. While the VGG16 architecture has only 13 convolutional layers, VGG19 has a higher depth with 16 convolutional layers.

Although these networks get consistent results in terms of image classification, they use a considerable disk size and take a long time to train.

MobileNet

MobileNet's architecture was proposed in 2017 by Google, suitable for use contexts where there is no large computing power available. They stand out from the rest because, although they are lightweight models and occupy low disk space, they achieve good accuracy in image classification problems [23].

There are two available versions of MobileNets, called MobileNet V1 and MobileNets V2. From the first to the second version, some improvements have been implemented with respect to its performance, thus being better suited for systems where memory access limitations exist.

R-CNNs

Initially proposed in 2013, R-CNN architectures are suitable for image object detection problems and can, therefore, be adapted for image fire recognition problems. The solution presented by these networks to the problem can be divided into two main phases: region proposal and classification [20].

In the region proposal phase, the object found is detected by drawing a bounding box in the corresponding area of the image. In the next phase, the classification is made based on the first result. These models have as their main advantage over all the others that they can determine the exact location of objects in the image [20].

Due to the high computational cost and slowness of R-CNN training, changes have been proposed to reduce the training time of this network, such as Fast R-CNN [19] and Faster-CNN [33].

2.2.2 Transfer learning

As stated in the previous section, one of the disadvantages of training a model is the amount of data necessary to obtain good results. The most common strategies used to reduce annotation efforts are dataset augmentation and transfer learning.

The use of synthetic data to increase the training dataset has been a strategy used to combat the lack of annotated datasets in object recognition problems. [32] Some studies propose the use of synthetic data to train the model for the problem of smoke recognition in images, that is, the artificial addition of smoke in neutral images [39].

Other strategies for dataset augmentation are the use of horizontal or vertical image shifts, cropping or scaling the images, adding them to the original data to increase the available dataset [36]. This can not only be useful as a way to compensate for the unavailability of training data but can also boost model performance, making it more robust [36].

When the annotated data available for training isn't enough to obtain good results, it may be necessary to use a pre-trained model optimized to solve a different problem and retrain it. A known technique is the adaptation of a model previously trained for another domain, Transfer Learning.

The use of transfer learning encompasses a few issues. When using this technique, discovering what knowledge to transfer is an important task, i.e., knowing what part of the information should be preserved from the previous training. If done correctly, it will allow the knowledge information transfer to be useful for the current problem's domain classification [30]. When using transfer learning, usually, the "classification" (see figure 2.2) part of the CNN is re-trained for adaptation to the new problem domain. Within this part of the network, the issue is choosing what layers to retrain.

It is also necessary to know what pre-trained model to use and what layers to retrain in order to adapt it to the new problem's domain [30]. This method can be applied to fire recognition problems by using the pre-trained weights of a network for another target domain and adapting them for feature extraction taking into account the classes of our problem, which facilitates the learning of the common features in both domains [32].

State-of-the-art models, as the ones mentioned in the previous section, can be used by applying pre-trained weights available, resulting from training models with any of the above-defined architectures to solve other classification problems or object recognition in images. With the initialization on the network using some pre-trained weights and using transfer learning techniques it is then possible their adaptation to the fire and smoke image classification problem's domain. It is also possible to combine both strategies, using synthetic data or other strategies for dataset augmentation to fine-tune a pre-trained model, applying transfer learning [32, 38].

2.3 Current approaches

This section provides an overview of the recent fire and smoke recognition methodologies and how they are used in current systems for fire and/or smoke recognition. Some of the proposed solutions for flame or smoke identification in images and videos are analyzed regarding their advantages and disadvantages.

These systems are divided between those that only use the extracted features information and those that use other types of strategies.

2.3.1 Feature-based approaches

Given the great advantages of early detection of fire detection, some systems have recently been developed and introduced for this purpose. The following are some of the systems whose algorithms have been previously measured, and their current uses in the real world are contextualized.

In the context of RESCUER project development ("Reliable and Smart Crowdsourcing Solution for Emergency and Crisis Management"), which, similarly to FireLoc, encompasses the development of a system that allows fire reports by crowdsourced data, several algorithms have been developed for fire recognition [8, 11, 16].

By using this system, anyone can send a photograph or video of the incident they are witnessing and reporting any visible fire or smoke in their surroundings. One of the most significant issues was dealing with the high volume of data collected. The number of images and videos submitted became impossible to be visually analyzed by experts in real-time. Therefore, the development of a real-time automatic validation system was imperative to identify which information might be useful, discarding false fire reports [8].

One of the proposed image validation systems was primarily used for fire recognition in social media images. Using this source to form the test dataset, actual system usage conditions were simulated. First, an analysis is made to the images of the different classes to extract features such as color, texture, and shapes present in the images. After calculating these features, Feature Vectors are constructed, aggregating them to be used in the classification of new images [11].

The algorithm used is based on *Instance Based Learning*, and consists of two steps. The first is the use of an evaluation function, which evaluates the proximity between pairs of vectors so that they can be aggregated into sets. Then a classification function is used to receive the vector sets for classification of new images, taking into account the classes considered. A concept description is also maintained to keep track of previous ratings. Fire recognition is then achieved by applying this algorithm to the obtained Feature Vectors [11].

In figure 2.4 is a representation of this algorithm, named *FFireDt*. "The Evaluating Module receives an unlabeled image, represents it executing feature extractor methods and labels it by using the Instance Based Learning Module. The system output (image plus label) interacts with the experts, who may also perform a similarity query". [11] In this image, *Feature Extraction Method* is referenced as FEM and *Instance-Based Learning* is referenced as IBL.

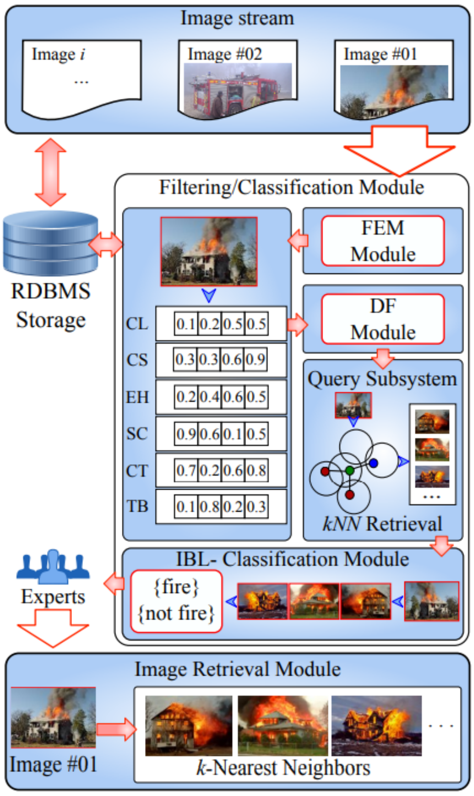

Figure 2.4: "Architecture of the FFireDt" [11]

Another algorithm proposed as a validation method for videos was BowFire and included a frame by frame analysis and, therefore, could also be applied to still images. The still image validation method, BoWFire, consisted of a color-based classification of the photos, keeping only the regions containing "fire" pixels. The result is then combined with another image, product a texture classification, only including the "fire-texture" pixels. If these images are coincident, the image is classified as fire [16].

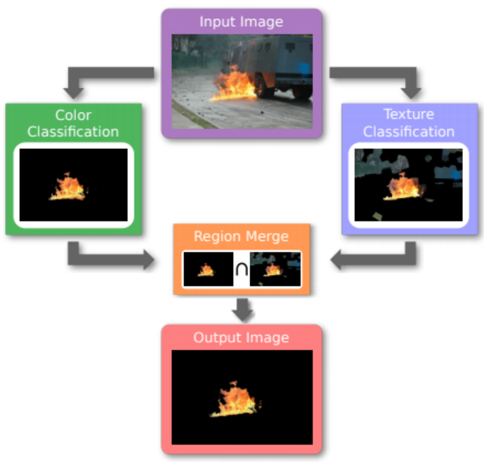

Figure 2.5: "Architecture of the BoWFire method" [16]

The application of traditional pattern recognition approaches to fire and smoke recognition is one of the methods that has been widely explored. However, these imply the additional step of identification and extraction of most discriminating features, which can be difficult for this problem [28].

2.3.2 Model-based approaches

In another approach, CNN was used for fire recognition. In this study, the network takes as input videos or images and gives as output the probabilities of it belonging to each class considered. All videos are first converted to images frames, and all images are then resized to fit the input size of the network. Each image is given to the network as a matrix with each RGB channel, and discriminating features are extracted. Parameter adjustment was achieved using backpropagation. The resulting model could be applied to both video and image inputs, classifying them as belonging to one of the classes considered, Fire or Not Fire [28]. In figure 2.6, from [28], is presented by the authors as a visual explanation of the algorithm used for data pre-processing, training, and classification using the model.

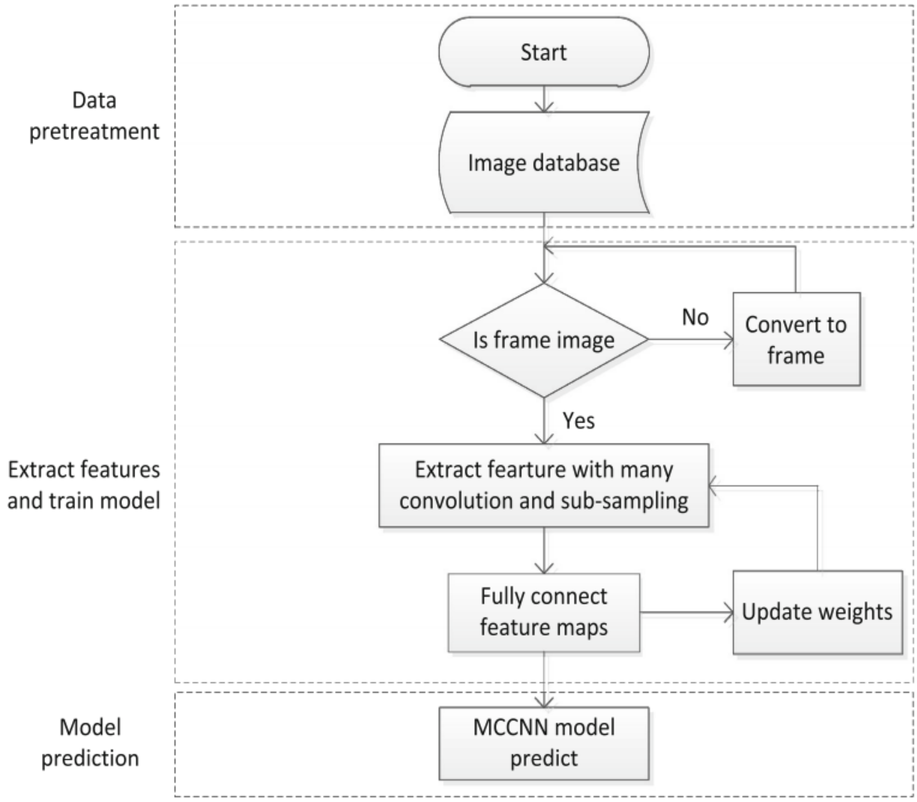

Figure 2.6: "Flowchart of fire recognition" [28]

Faster R-CNNs have also been successfully applied for smoke recognition tasks. The principle behind the use of these networks is its success and accuracy in object recognition tasks [33]. This study proposes the use of a Faster R-CNN to identify smoke regions, thus recognizing smoke in videos [39]. One of the advantages of using these types of models is the possibility of identifying the location, in the image, where the fire is recognized. In figure 2.7 is a diagram of the algorithm used for smoke detection using a faster R-CNN model, from [39].

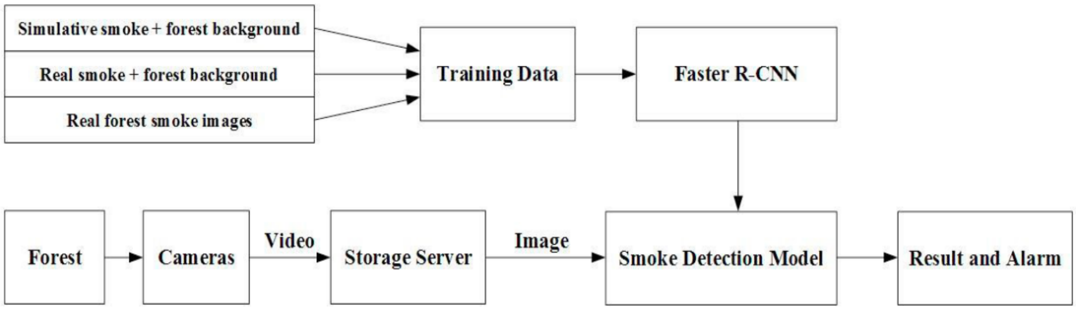

Figure 2.7: "Flowchart of forest smoke detection" [39]

Another recent system used "captioning and classification of dangerous situations" to autonomously detect anomalies, for robot applications. The adopted approach makes use of a CNN to identify anomalous situations in images. For this system's classification task, the Inception model was used, which allowed the construction of a trainable end-to-end architecture. In this system, the Inception module was used for recognition of other situations: "broken windows, injures people, fights, car accidents, guns and domestic violence" [10].

2.4 Final remarks

Given the state-of-the-art fire recognition systems nowadays, some relevant considerations are pointed below.

There is a great deal of interest today in developing alternatives for the automatic visual recognition of fire and smoke. However, many of these are not specific for the recognition of forest fires, and some of them just consider classifying images as either fire or non-fire-containing.

There is also a lack of available datasets, annotated according to the classes Fire, Smoke, and Neutral, that is, without the presence of smoke or fire. In addition, many of the available datasets do not contain specific circumstances of wildfires, including urban situations with objects such as cars and houses, which can be confusing to classify.

This page is intentionally left blank.

Chapter 3

Proposed Approach

This chapter presents the approach used to develop the fire recognition system. As part of the FireLoc project, this work has as its primary objective the development of an intelligent fire and smoke recognition system in still images taken using smartphone cameras. For this system, image classification is performed, taking into account three different classes: *Fire*, *Neutral*, and *Smoke*.

The Neutral class includes images in which there is no smoke or fire flames present. To the Smoke class belong all the images where there is smoke presence, having no visible flame. The Fire class includes images with fire flames present.

A distinction is made between images containing fire and images containing only smoke since the presence of flames is more indicative of the presence of a fire. While in a picture with flames, it is safe to say that there is a fire and that can be used to infer the location of the occurrence. In images where only smoke is visible it is not possible to assertively locate the fire that gave rise to it.

By discarding false reports, which will be identified as images of the Neutral category, and determining which reports are sightings of flames in forest territory, it is possible to have better communication of incidents. The developed system will, therefore, help in the decision of intervention to be taken by the fire fighting means.

The first process of development was focused on studying previous approaches to similar problems, object detection, fire and smoke recognition in images, and existing systems developed for visual identification and reporting of fires in both urban and forest contexts.

In the development of the proposed system, the implemented methodologies will be based on Deep Learning approaches, focusing on state-of-the-art models for image object recognition, CNNs. Using these models makes it possible to automatically extract the most relevant features observed in the images and, consequently, obtain better classification results.

Deep learning approaches, such as the transfer learning technique applied in the first semester, will be studied for the development of the algorithm. Methods for improving existing datasets and developing the specific dataset will also be explored.

The developed system will have to go through a testing phase, using photographs taken in the context in which it will be used, that is, in forest fire conditions, to evaluate its performance. For the system to be reliable, a testing phase is required to show that no significant errors are observed. This means that for the system to be used in real conditions, it is ideal to obtain a low false-negative rate.

Possible difficulties that may interfere with project development include the lack of a benchmark dataset used for such problems. The lack of a benchmark dataset implies that a specific dataset must be established for this purpose. The used data should be representative of all possible situations and should provide a sufficient number of examples from each class to allow model training and evaluation to be possible.

The proposed approach for the development of the system consists of an iterative process of implementing and testing different deep learning approaches for recognizing smoke and fire in images. The tests performed to evaluate the performance of the models imply a later phase of analysis of the results obtained using images that represent the real conditions of use.

The main objective will be to obtain a fire and smoke recognition algorithm in images. For this, one or more models can be used, which will have to be optimized for the problem. The approach to the problem should also focus on the development of a specific dataset for fire and smoke recognition in the context of forest fires.

The developed algorithm will still have to obtain consistent results in real situations of use. As such, it may include the identification, in the space of the submitted image, of the location of the fire flames.

In the first semester, the ResNet models were studied and applied to this problem. A study of the different possibilities for image pre-processing was carried out, as it is a necessary step for them to be used as input for these models. Only with the pre-processing phase completed, these images would be used in the training and performance evaluation phases of the models.

Considering that ResNets are CNN models, they able to extract useful features from images and generally achieve good results in object classification and recognition tasks. In addition, due to the existence of "shortcut connections" for every n blocks in their architecture, the performance of these models does not degrade as significantly with the increase in the depth of the network. The tests carried out in the first semester aimed to study the feasibility of using these models in the fire recognition system.

Chapter 4

Current work and Preliminary Results

In this chapter, the work developed in the first semester is presented and the results are discussed. First, the setup of the fire recognition system in images is presented. The method used for training and testing the models is described, as well as the necessary steps to adopt this method for solving the problem. Then, the datasets used for training and evaluation of the models are also presented. The differences between them are discussed, as well as their adequacy to the problem of recognizing forest fires in images. The image pre-processing steps used to prepare the images for training and testing tasks are also presented.

4.1 Fire Recognition System Setup

As an initial approach for the development of the system, due to lack of available specific datasets, the Transfer Learning technique was used. This was done by applying the residual networks or ResNets architecture, inicialized with the weights optimized for recognizing the *ImageNet* dataset classes.

To adapt the model to the new problem domain, it was necessary to change ResNet's last fully connected layer, from the 1000 classes considered for the Imagenet dataset, to the three classes considered in this problem. This network layer is then retrained, adapting the classification to the domain currently considered for the *Fire*, *Smoke*, and *Neutral* image classes.

The development strategy then goes through a model training and validation phase, in which the network is initialized with the weights corresponding to the optimization for the *ImageNet* problem and fine-tuned to adapt to the new domain considered.

Given the context in which the system will be applied to, the images will be captured using different smartphones, and there will be no uniform resolution of the submitted photographs. In addition, the photos will be captured by humans present on the ground, indicating that there may be imperfections such as blurred parts or strange lighting conditions that can difficult classification.

In order to facilitate feature capture by the network, data preparation or pre-processing may be needed. Therefore, different formats, sizes, and qualities of images used in the training and performance evaluation of the model should be considered.

4.2 Datasets

In the initial tests, to allow for an evaluation of existing methods for fire recognition, an open-source dataset was used. This dataset is balanced, i.e., contains an equal amount of images from each category (*Smoke*, *Fire*, and *Neutral*), and has 300 images for training in each group and another 100 for validation, making a total of 1000 images, 900 for training and 100 for model validation. This dataset is named *Fire-Smoke-Dataset*.

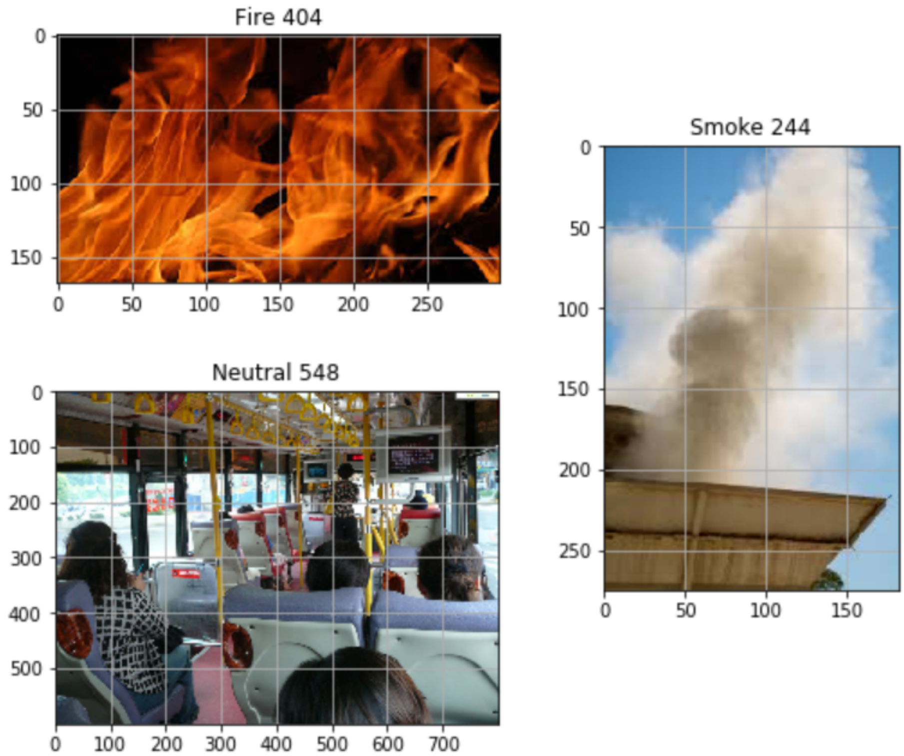

Figure 4.1: Images from the Fire-Smoke Dataset

During the semester, another set of photographs emerged, taken in a simulacrum, performed by firefighters. The pictures were taken using different smartphones and tablets, thus allowing them to simulate real conditions of use of the system. These images are all corresponding to the context of forest fires. A representation of all the classification classes was also taken into consideration, meaning that that in this specific image dataset, there are images with smoke, others with fire, and others without any evidence of fire.

The dataset of specific images collected so far only contains a total of 427 images, 165 identified as Fire, 77 identified as Neural, and 185 identified as smoke. These sets of images can be considered as a representation of real-world situations in which the system will be used, having varying sizes and proportions and being taken in different orientations, i.e., some are horizontal and others vertical. This dataset is named *Real-Images-Dataset*.

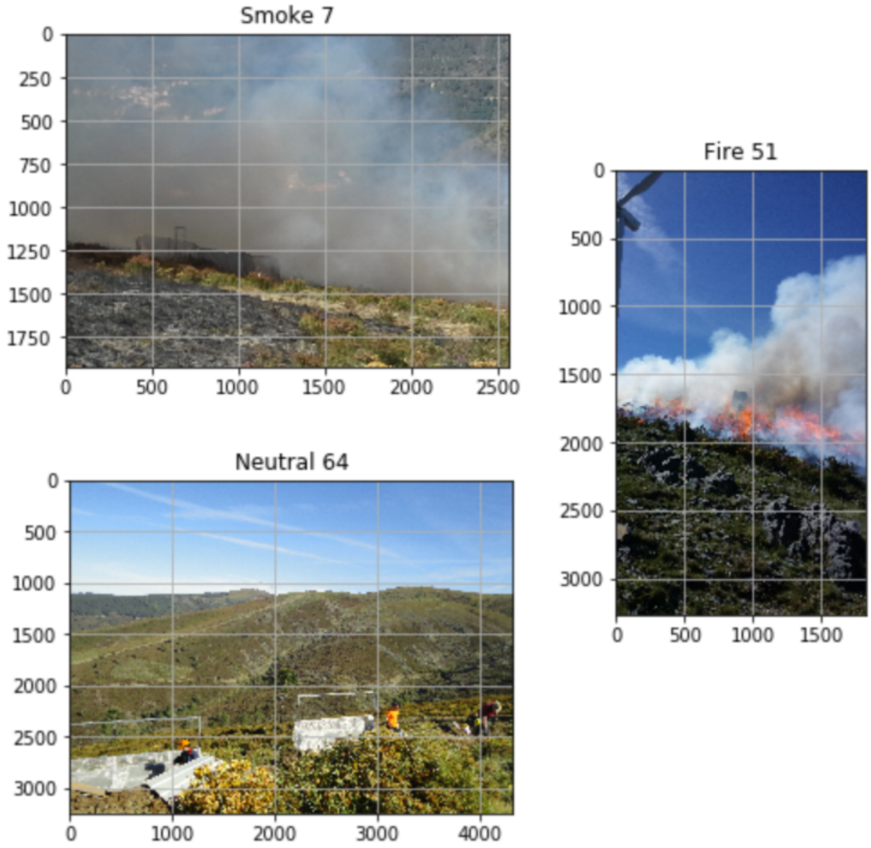

Figure 4.2: Images from the *Real-Images-Dataset*

	Max shapes			Min shapes		
Dataset	**Fire class**	**Smoke class**	**Neutral class**	**Fire class**	**Smoke class**	**Neutral class**
Fire-Smoke	300,429	307,427	2304,3072	108,168	118,96	145,176
Real-Images	4320,4320	4320,4320	3240,4320	1836,1836	1836,1836	1836,2560

Table 4.1: Dimension of datasets' image

There are some differences between both datasets in terms of size, presence of objects, and situations represented. In the first dataset, there are not only images corresponding to forest fires but also indoor ignitions, and accumulations of smoke. As an opposite to that, the photos taken in a fire drill context only consider wildland ignitions with fewer people and objects, different lighting conditions, clouds, and were all taken in a non-urban environment. However, due to the low amount of images available corresponding to each class, in this initial approach, the *Real-Images-Dataset* will only be used for model testing and performance evaluation.

There is also a considerable difference between the dataset's image sizes, which can cause additional difficulties in classification. For a further idea of the size discrepancy, see table 4.1.

4.3 Data pre-processing

ResNet's model training requires an initial image pre-processing step to facilitate feature extraction. The weights with which the network is initialized correspond to the optimization result for object recognition in the ImageNet dataset. As such, feature extraction from input images follows a series of conventions, which must be kept for the images given to the network. These conventions include adapting each image to 244 x 244 size and splitting the image into RGB channels.

The first approach included a direct resize of each input image to the 244x 244 size and separation into the different RGB channels, as exemplified in figure 4.3.

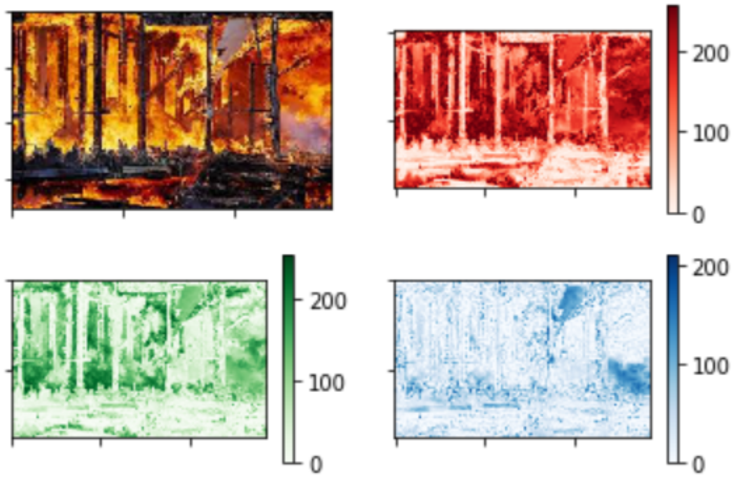

Figure 4.3: RGB components of an image

The image is also normalized to fit values within the range [-1, 1]. Thus it becomes possible to mitigate possible detrimental effects of the noise present in the images in the identification of features. This step can also facilitate the extraction of relevant features by standardizing the information on each image before training the neural network.

As a simplification for initial tests, the images of the different classes are organized in iterable batches of 32 images each. This simplification is used since the use of different batch sizes can affect the training performance of the model. In the future, this parameter should be optimized by varying it to obtain the best value. However, the tests presented set it at 32.

The datasets are also shuffled. This means that, at every epoch of training of the model, the images will be organized in a different order. Therefore, reshuffling the data at each iteration can contribute to better training of the model and help avoid early overfit due to possible patterns in class order.

4.4 Training and evaluating the model

As stated above, only the fully connected layer of the model is retrained. For that, the *Fire-Smoke-Dataset* will be used. After the pre-processing steps described in the previous section, the model iterates over the 900 images of each class, in the training set. A validation step with the other 100 images is done by the end of each iteration.

The models were tested with images from *Real-Images-Dataset*, in order to understand their behavior in real conditions of use.

For the optimization of ResNets models, there are two main optimizers used: Stochastic Gradient Descent (SGD) and Adam [25, 37]. These are efficient stochastic optimization methods and are widely used for optimization tasks.

As for the Adam optimizer, from [25]:

- "only requires first-order gradients with little memory requirement" [25];
- "individual computes adaptive learning rates for different parameters from estimates of first and second moments of the gradients" [25];
- "adapts learning rate scale for different layers instead of hand picking manually as in SGD " [25].

Adapting the learning rate can help CNN training. However, when adjusting the learning rate, errors may arise detrimental to the model's training [25, 37].

As an alternative, the Stochastic Gradient Descent with momentum algorithm emerged. Using momentum-based acceleration, using this algorithm can speed up the training of the network, without deteriorating the results [37].

The first tests were executed to compare the performance of trained models using both optimizers, and setting the initial learning rate to 0.001.

For the application of transfer learning, the ResNet models were initialized with the training weights for the Imagenet dataset [9], and only the last layer, fully connected, was retrained. As a simplification adopted, the initial tests were carried out using ResNet18.

4.5 Results and discussion

In this section, the preliminary test results are presented and discussed.

4.5.1 Evaluation Metrics

To evaluate the performance of the trained models, the following metrics were used: confusion matrix and f1-score.

Since the data in the test dataset is not balanced, the f1-score value was used to measure the models' performance. We can, therefore, interpret the f1-score measure as a weighted accuracy value, regardless of the number of images belonging to each class of the dataset.

The observation of the obtained confusion matrices will allow the verification, in greater detail, of the model performance in the classification of the test data. It can also be useful to understand the most significant classification errors and help to improve the training of the model.

4.5.2 Test Results

The first test was a comparasion between using the SGD optimizer with momentum or the Adam optimizer to train the model, for 500 iterations over the data, with ResNet18. The images were pre-processed before training the model. In addition to being separated into their RGB components, these images were also resized in their entirety to fit the size of the network input, 224 x 224.

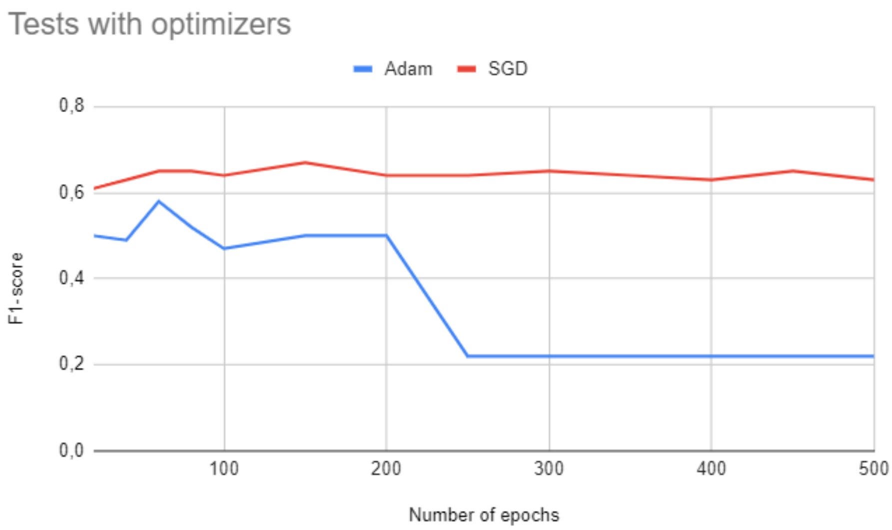

Figure 4.4: F1-score results

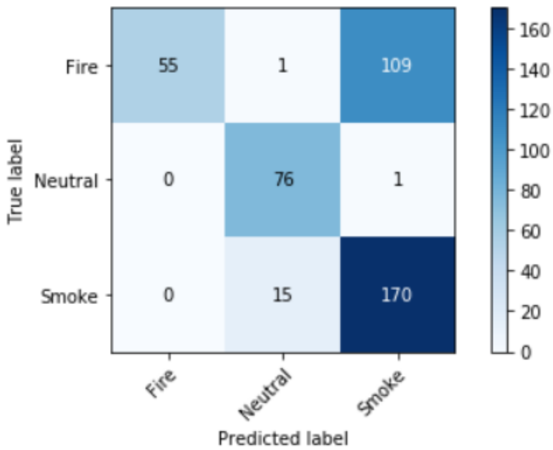

Figure 4.5: Confusion Matrix - best results

In general, using SGD with momentum, the model was able to obtain better results. The best result was achieved at epoch 150, see figure 4.4.

From epoch 200 forward, see figure 4.6, the model in which the optimizer Adam was used, classifies all input images as belonging to the Fire class. This result may indicate that, for training with this optimizer, additional steps for data pre-processing may be necessary. It may also be necessary to retrain more layers, before fully connected, in order to adapt to the new problem domain.

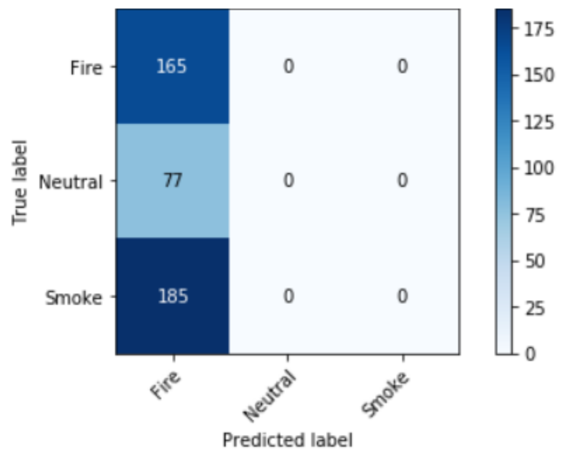

Figure 4.6: Confusion Matrix - using Adam

Other ResNet models

Using the SGD optimizer, in addition to ResNet 18, the other available ResNet models, 34, 50, 101, and 152 were then tested.

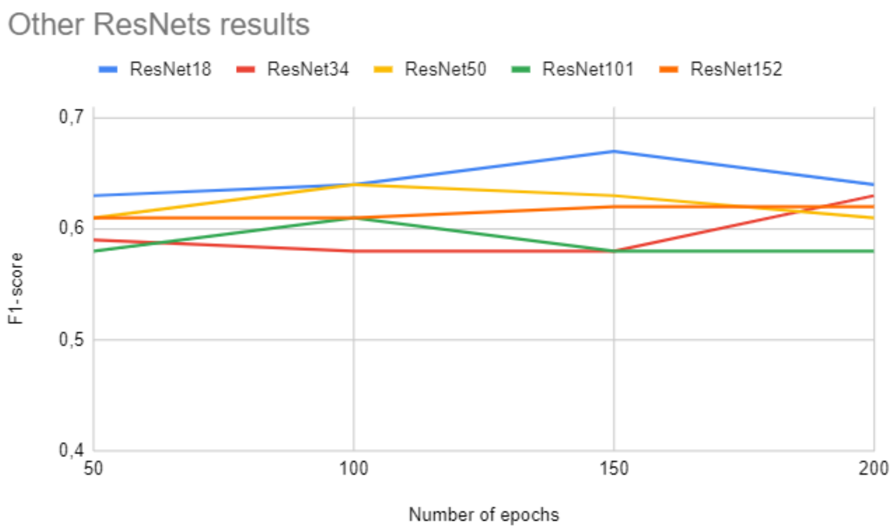

Figure 4.7: Results obtained with other ResNet models

The results obtained, in figure 4.7 suggest that the most suitable model for the problem will be a ResNet18. However, additional testing will be needed to see if less deep models can perform better.

Additional tests

Some additional tests were also carried out, using the images in their original size, but fragmented into portions of 224 x 244 dimensions.

('Image of:', 'Class: Smoke', 'Confidence score: 0.576171875')

Figure 4.8: Model classification of full resized image

('Image of:', 'Class: Fire', 'Confidence score: 0.5458984375')

Figure 4.9: Model classification of part of the original image

For some of the observed situations, as in the example of 4.8, when the resized image is classified, the model returns the result of belonging to the *Smoke* class. When dividing the image and performing the classification on the part of the image where there are flames, see figure 4.8, the model classifies it as belonging to the class *Fire*.

This difference in classification may be due to the loss of information detail in the pre-processing operation of direct adjustment of the image size. However, the division of images can cause additional errors in the classification of neutral images in which there is fog, clouds, or snow. By making a more detailed analysis of the image, without a global context, the model can confuse these phenomena with the presence of smoke, worsening the classification performance.

There is, therefore, the need to conduct additional tests to conclude whether, when using this method, it is possible to make a better distinction between Fire class and Smoke class images by models.

To perform these tests, it is, therefore, necessary to consider:

- If any part of the image is classified as *Fire*, the image contains fire;
- If there are no parts classified as fire but some are classified as Smoke, the image contains smoke;
- Only in cases where no part of the image is classified as Fire or Smoke the image belongs to the Neutral class.

Considering the objectives of the project, these first results are promising. Future work should focus on improving the confusion between the Fire and Smoke classes, which is observed 4.5.

Chapter 5

Work Plan

In this chapter, the timeline of the work followed in the first semester is presented. A work plan for the second semester is also presented.

5.1 Work performed in the first semester

The figure below presents a *Gantt* chart, referent to the temporal planning of the work developed in the first semester.

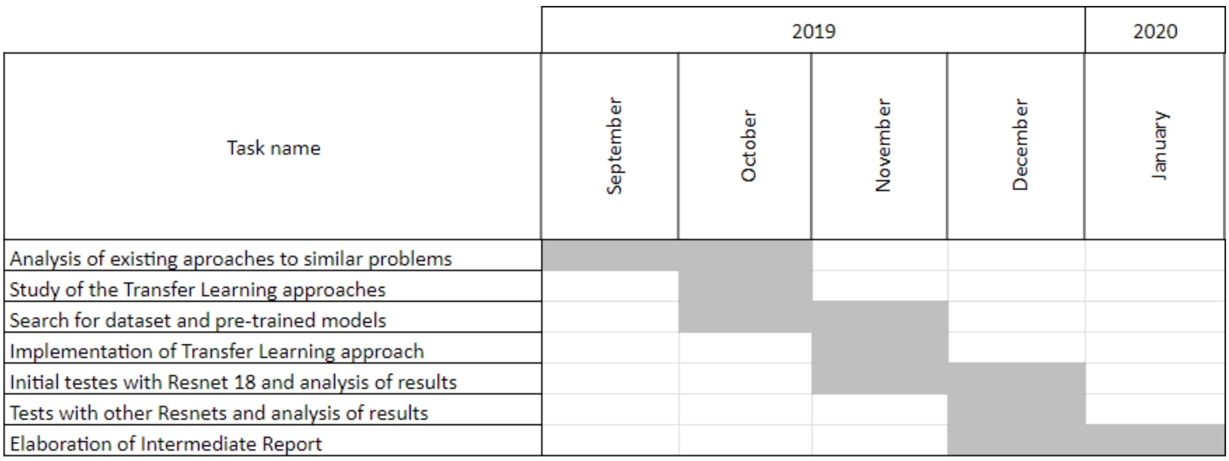

Figure 5.1: First semester *Gantt* chart

At the beginning of the semester, the focus of the work was the study of approaches and methodologies currently used to detect fire and smoke in images. The choice of the Transfer learning technique was mainly due to the unavailability of large amounts of available annotated data, specific to the problem.

For this reason, an alternative dataset was used, allowing to test the approach, even under conditions that do not fully correspond to the actual use of the system. Some photographs were also made available, taken in a simulacrum, by smartphones and tablets. In order to be able to evaluate the performance of the models, it was necessary to manually annotate these images, to be used in the tests performed.

Additional tests were also planned, using other models and comparing the results obtained with ResNet. However, due to development delays, it was not possible to complete all the work initially planned.

5.2 Work plan for the second semester

In this section, the time planning of the tasks to be carried out in the second semester is presented below.

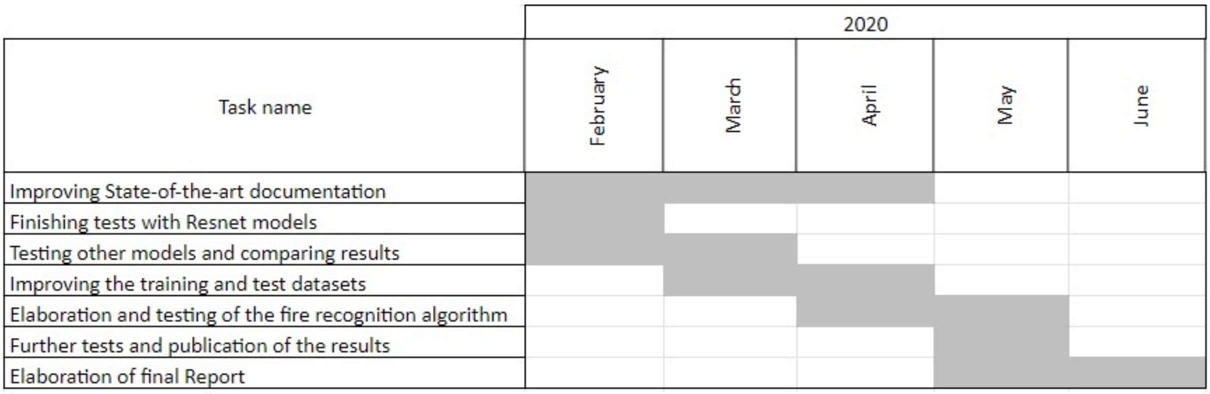

Figure 5.2: Second semester *Gantt* chart

Taking into consideration the work developed in the first semester, and the results obtained, it is safe to say that additional tests will be necessary to optimize the Transfer Learning technique studied.

Additionally, it might still be necessary to improve the dataset used in order to achieve better results. For this purpose, other images collected during the semester, in other simulations, for example, or some of the techniques studied in the state-of-the-art for data augmentation, may be used.

In the final phase of the second semester, the algorithm developed, as well as the results obtained, will be documented, and the dataset achieved will be made available. The final dissertation report will also be written.

5.3 Risks

In this section, some risks that may affect the work to be carried out in the next semester are analyzed. The analysis of each risk takes into consideration its probability of occurrence and level of impact on the work developed, according to table 5.1. Mitigation actions for each risk are also proposed.

	Probability	**Impact**
1	Low	Marginal
2	Medium	Critical
3	High	Catastrophic

Table 5.1: Scale used for Risk evaluation

Table 5.3 is a risk matrix, providing visual analysis of the impact of each risk, showing Risk 2 as the most severe.

Risk ID	Risk - Consequences	Probability	Impact
R1	Training deep learning models can take a long time and might lead to delays in getting results	3	2
R2	Not enough available annotated data for training	2	2

Table 5.2: Risk identification and analysis

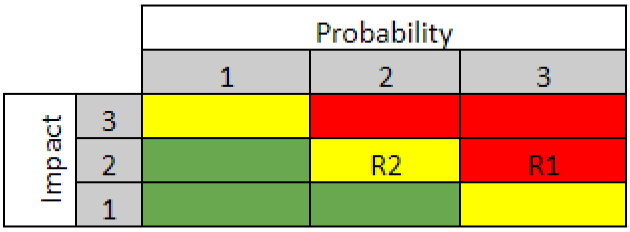

Figure 5.3: Summary of the risk analysis

For the risks identified, possible mitigation strategies are:

- Risk 1 - Use of techniques to make training more efficient, using tools like Google Colab
- Risk 2 - Construction of an annotated dataset, using images from different sources, suitable for the problem

Chapter 6

Conclusion

In the first phase of the project's development, state-of-the-art systems for recognizing fire and smoke in images were studied. There were also studied as different approaches to deep learning for problems of object recognition in images that can be used for recognition of fire and smoke.

By using the learning transfer technique, it was possible to achieve some initial results, which allowed the understanding of the most significant problems faced regarding fire and smoke recognition. The work performed with an existing dataset allowed the study of image pre-processing techniques that can be used to improve model training.

The work plan for the second semester is based on the results that have been achieved so far. Additional tests are needed to optimize the different models considered in order to assess their feasibility. Further tests will be carried out using other models, and its results will be compared with the results obtained in the initial approach.

The next phase of development will focus on the study of other models similar to ResNet, e.g., VGG [35] or MobileNets [23], applied to the fire recognition problem. Alternatives that allow the location of the flames identified in the image space, as R-CNNs [19, 20, 33], will also be evaluated.

Based on the performance evaluation results of the models obtained, it will be possible to study techniques for improving datasets and further develop the fire recognition algorithm in images.

References

[1] AML - Advanced Mobile Location. https://www.sg.mai.gov.pt/Noticias/Paginas/Demonstra%C3%A7%C3%A3o-real-do-AML---Advanced-Mobile-Location.aspx. accessed 2020-01-17.

[2] Australia's Deadly Wildfires in Photos: The View from Space. https://www.space.com/australia-wildfires-satellite-images-2019-2020.html. accessed 2020-01-19.

[3] Bushfire-destroyed homes should not be rebuilt in riskiest areas, experts say. https://www.theguardian.com/australia-news/2020/jan/19/bushfire-destroyed-homes-should-not-be-rebuilt-in-riskiest-areas-experts-say. accessed 2020-01-19.

[4] Convolutional Neural Networks (CNNs / ConvNets). http://cs231n.github.io/convolutional-networks/. accessed 2020-01-17.

[5] Fatalities due to the most significant wildfires worldwide as of 2016. https://www.statista.com/statistics/234723/fatalities-due-to-the-most-significant-wildfires/. Accessed: 2020-01-10.

[6] Incêndios. Portugal tem 41.017 hectares de área ardida em 2019. https://expresso.pt/sociedade/2019-10-01-Incendios.-Portugal-tem-41.017-hectares-de-area-ardida-em-2019. accessed 2020-01-19.

[7] An intuitive guide to convolutional neural networks. https://www.freecodecamp.org/news/an-intuitive-guide-to-convolutional-neural-networks-260c2de0a050/. Accessed: 2020-01-17.

[8] Project RESCUER – new communication platform to save lives. http://www.rescuer-project.org/. accessed 27-12-2019.

[9] TORCHVISION.MODELS. https://pytorch.org/docs/stable/torchvision/models.html. accessed 2020-01-18.

[10] Octavio Arriaga, Paul Plöger, and Matias Valdenegro-Toro. Image Captioning and Classification of Dangerous Situations. nov 2017.

[11] Marcos V.N. Bedo, Gustavo Blanco, Willian D. Oliveira, Mirela T. Cazzolato, Alceu F. Costa, Jose F. Rodrigues, Agma J.M. Traina, and Caetano Traina. Techniques for effective and efficient fire detection from social media images. *ICEIS 2015 - 17th International Conference on Enterprise Information Systems, Proceedings*, 1(June):34–45, 2015.

[12] Sam G Benjamin. A Comparative Analysis on Different Image Processing Techniques for Forest Fire Detection. *IJCSN International Journal of Computer Science and Network*, 5(1), 2016.

[13] T. Celik, H. Demirel, and H. Ozkaramanli. Automatic fire detection in video sequences. *European Signal Processing Conference*, (January), 2006.

[14] T. Celik and Kai-Kuang Ma. Computer Vision Based Fire Detection in Color Images. pages 258–263, 2008.

[15] Thou-ho Chao-ho Chen, Ping-hsueh Wu, and Yung-chuen Chiou. An Early Fire-Detection Method Based on Image Processing. pages 1707–1710, 2004.

[16] Daniel Y. T. Chino, Letricia P. S. Avalhais, Jose F. Rodrigues, and Agma J. M. Traina. BoWFire: Detection of Fire in Still Images by Integrating Pixel Color and Texture Analysis. jun 2015.

[17] Sebastien Frizzi, Rabeb Kaabi, Moez Bouchouicha, Jean Marc Ginoux, Eric Moreau, and Farhat Fnaiech. Convolutional neural network for video fire and smoke detection. *IECON Proceedings (Industrial Electronics Conference)*, pages 877–882, 2016.

[18] Rickard Brüel Gabrielsson. Topological Data Analysis of Convolutional Neural Networks' Weights on Images. 76(1):2007, 2009.

[19] Ross Girshick. Fast R-CNN. *Proceedings of the IEEE International Conference on Computer Vision*, 2015 Inter:1440–1448, 2015.

[20] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 580–587, 2014.

[21] Oluwarotimi Giwa and Abdsamad Benkrid. Fire detection in a still image using colour information. Technical report.

[22] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. dec 2015.

[23] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. 2017.

[24] Arpit Jadon, Mohd. Omama, Akshay Varshney, Mohammad Samar Ansari, and Rishabh Sharma. FireNet: A Specialized Lightweight Fire & Smoke Detection Model for Real-Time IoT Applications. may 2019.

[25] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. dec 2014.

[26] D Krstinić, D Stipaničev, and T Jakovčević. Histogram-based smoke segmentation in forest fire detection system. *Information Technology and Control*, 38(3):237–244, 2009.

[27] Niall O Mahony, Sean Campbell, Anderson Carvalho, Suman Harapanahalli, Gustavo Velasco Hernandez, Lenka Krpalkova, Daniel Riordan, and Joseph Walsh. Deep Learning vs . Traditional Computer Vision. (Cv).

[28] Wentao Mao, Wenpeng Wang, Zhi Dou, and Yuan Li. Fire Recognition Based On Multi-Channel Convolutional Neural Network. *Fire Technology*, 54(2):531–554, mar 2018.

[29] Vinod Nair and Geoffrey E. Hinton. Rectified linear units improve restricted boltzmann machines. In Johannes Fürnkranz and Thorsten Joachims, editors, *ICML*, pages 807–814. Omnipress, 2010.

[30] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010.

[31] Yan Qiang, Bo Pei, and Juanjuan Zhao. Forest Fire Image Intelligent Recognition based on the Neural Network. *Journal of Multimedia*, 9(3), 2014.

[32] Param S. Rajpura, Hristo Bojinov, and Ravi S. Hegde. Object Detection Using Deep CNNs Trained on Synthetic Images. jun 2017.

[33] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1137–1149, 2017.

[34] Prasun Roy, Subhankar Ghosh, Saumik Bhattacharya, and Umapada Pal. Effects of Degradations on Deep Neural Network Architectures. pages 1–11, 2018.

[35] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, pages 1–14, 2015.

[36] Lei Sun. ResNet on Tiny ImageNet. Technical report.

[37] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. Technical report, 2013.

[38] Wonjae Lee, Seonghyun Kim, Yong-Tae Lee, Hyun-Woo Lee, and Min Choi. Deep neural networks for wild fire detection with unmanned aerial vehicle. pages 252–253, Jan 2017.

[39] Qi Xing Zhang, Gao Hua Lin, Yong Ming Zhang, Gao Xu, and Jin Jun Wang. Wildland Forest Fire Smoke Detection Based on Faster R-CNN using Synthetic Smoke Images. In *Procedia Engineering*, volume 211, 2018.

[40] Jianhui Zhao, Zhong Zhang, Shizhong Han, Chengzhang Qu, Zhiyong Yuan, and Dengyi Zhang. SVM based forest fire detection using static and dynamic features. *Computer Science and Information Systems*, 8(3):821–841, 2011.

Appendices

This page is intentionally left blank.

Images from *Fire-Smoke-Dataset*

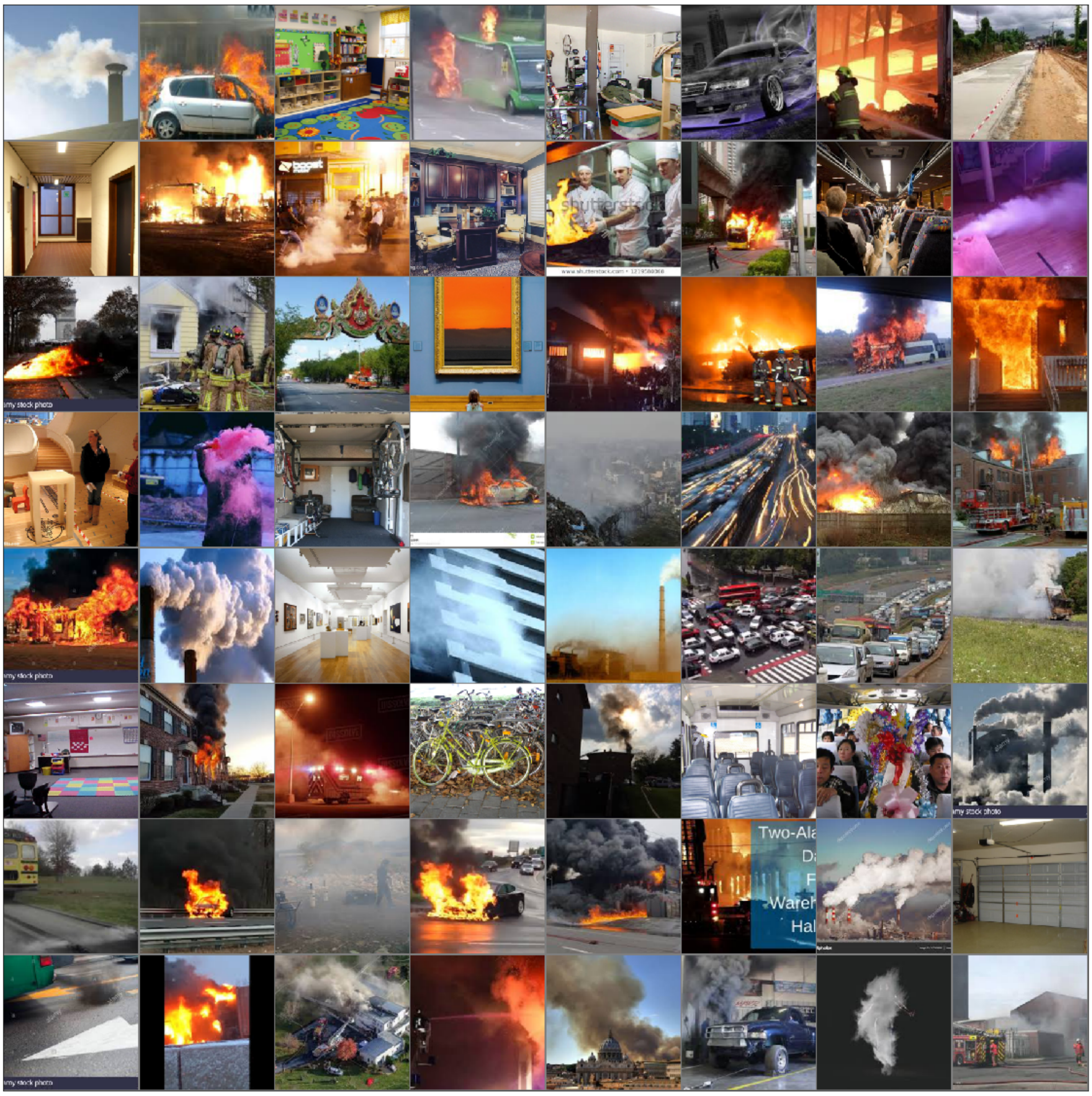

Figure 1: Example Fire-Smoke-Dataset images

This page is intentionally left blank.

Images images from *Real-Images-Dataset*

Figure 2: Example images from *Fire-Smoke-Dataset*