Federal University of Juiz de Fora
Institute of Exact Sciences
Bachelor's Degree in Computer Science

Fire detection through a combination of deep neural networks and graph cuts

Davi Magalhães Pereira

JUIZ DE FORA
DECEMBER, 2023

Fire detection through a combination of deep neural networks and graph cuts

Davi Magalhães Pereira

Federal University of Juiz de Fora

Institute of Exact Sciences

Departament of Computer Science

Bachelor's Degree in Computer Science

Advisor: Saulo Moraes Villela

Co-Advisor: Marcelo Bernardes Vieira

JUIZ DE FORA

DECEMBER, 2023

Fire detection through a combination of deep neural networks and graph cuts

Davi Magalhães Pereira

MONOGRAPH SUBMITTED TO THE FACULTY OF THE INSTITUTE OF EXACT SCIENCES OF THE FEDERAL UNIVERSITY OF JUIZ DE FORA, AS AN INTEGRAL PART OF THE NECESSARY REQUIREMENTS TO OBTAIN A BACHELOR'S DEGREE IN COMPUTER SCIENCE.

Accepted by:

Saulo Moraes Villela
D.Sc. in Systems Engineering and Computer Science

Marcelo Bernardes Vieira
D.Sc. in Image and Signal Processing Sciences

Luiz Maurílio da Silva Maciel
D.Sc. in Systems Engineering and Computer Science

Heder Soares Bernardino
D.Sc. in Computational Modeling

JUIZ DE FORA
DECEMBER 13, 2023

Abstract

Recent developments in computer vision techniques have markedly improved fire detection capabilities compared to conventional systems. This work introduces an innovative methodology that integrates deep neural networks for identifying instances and regions of fire, graph cuts, and color thresholding for a nuanced approach to fire segmentation. The incorporation of fire segmentation masks facilitates precise analysis, providing valuable insights into fire origins and propagation to proactively prevent future incidents. Our method, leveraging graph cuts segmentation with comprehensive color information, demonstrates enhanced accuracy and detailed fire detection. The results illustrate a notable improvement in recall, maintaining competitive precision, thereby establishing an efficient and effective fire detection framework.

Keywords: Fire detection; fire segmentation; image classification; graph cut; deep learning; color thresholding.

Resumo

Métodos recentes de visão computacional têm avançado significativamente em detecção de incêndios em comparação com sistemas tradicionais. Este trabalho apresenta uma metodologia inovadora que integra redes neurais profundas para identificar instâncias e regiões de incêndio, cortes de grafos e limiarização de cores para uma abordagem detalhada na segmentação de incêndios. A incorporação de máscaras de segmentação de incêndios facilita uma análise precisa, fornecendo informações valiosas sobre a origem e propagação de incêndios para prevenir futuros incidentes de maneira proativa. O método proposto, aproveitando a segmentação por cortes de grafo com informações abrangentes de cor, demonstra uma precisão aprimorada e uma detecção de incêndios detalhada. Os resultados ilustram uma melhoria notável na taxa de verdadeiros positivos, mantendo uma precisão competitiva, estabelecendo assim um *framework* eficiente e eficaz para detecção de incêndios.

Palavras-chave: Detecção de fogo; segmentação de fogo; classificação de imagens; corte de grafo; aprendizagem profunda; limiar de cor.

Acknowledgements

I would like to express my sincere appreciation to my advisors Saulo Moraes and Marcelo Bernardes, whose guidance and insightful feedback have been invaluable. Without your expertise, patience, and support, this work would not have been possible. Their mentorship has not only enriched my academic journey but has also inspired me to delve into the fascinating realm of this research field.

I also want to acknowledge the professors of the Department of Computer Science for their dedication to teaching and their contributions to my personal and professional growth during these years. Your passion for imparting knowledge has left a lasting impact on me.

To my friends, who stood by my side during the challenges and triumphs of this academic pursuit, your camaraderie has made this path all the more meaningful.

Finally, I extend my heartfelt gratitude to my family, whose unwavering support and encouragement have been a constant source of strength throughout my academic journey. Your belief in me has been my driving force, and I am profoundly grateful for your sacrifices and understanding.

"Chaos breeds life, where order breeds habit."

Henry Adams

Contents

List of Figures

List of Tables

List of Abbreviations

AI	Artificial Intelligence
CNN	Convolutional Neural Network
CV	Computer Vision
DL	Deep Learning
DNN	Deep Neural Network
FPR	False Positive Rate
HSV	Hue Saturation Value
MBConv	Mobile Inverted Bottleneck Convolution
ML	Machine Learning
NLP	Natural Language Processing
ReLU	Rectified Linear Unit
RGB	Red Green Blue
TPR	True Positive Rate

1 Introduction

1.1 Problem Overview and Context

Fire detection is a crucial aspect of fire safety in various environments, including homes, public buildings, industrial facilities, and forests. Traditional fire detection systems rely on sensors that detect smoke, heat, or flames to alert occupants about a potential fire. However, these systems can be limited in their accuracy and may produce false alarms. With the advent of neural networks and other forms of artificial intelligence (AI), there is a growing interest in developing more effective fire detection systems. This chapter provides the motivation for the development of new fire detection methods, outlines the main challenges associated with the problem, highlights the relevance of the work, and outlines the objectives.

Fire detection and surveillance systems have significantly evolved over time, becoming more advanced and efficient in preventing the spread of fires. These systems have become integral parts of many buildings, factories, and public places, providing early alerts and protection against potentially deadly fires. Analyzing the spread of fires is a crucial aspect of fire detection and surveillance systems, as it enables more efficient and effective fire prevention and control.

In recent years, surveillance systems have become increasingly interconnected and intelligent, utilizing a variety of technologies such as AI and the Internet of Things (IoT). These systems use AI algorithms to analyze real-time video data from cameras, identifying and flagging unusual or suspicious behaviors. On top of that, they can utilize not only real-time detection but also image analysis of sets of images to identify patterns of fire propagation, potential sources of fires, and vulnerable areas. By leveraging this information, necessary measures can be taken to prevent losses.

As a pivotal component of these advancements, AI refers to the simulation of intelligence in machines that are programmed to perform tasks that typically require intelligence, such as visual perception, speech recognition, decision-making, and natural

language processing.

While AI broadly encompasses the simulation of human intelligence in machines, its subfield, Machine Learning (ML), takes a specific approach. ML allows systems to learn and improve automatically from experience without being explicitly programmed. This involves the use of statistical and mathematical algorithms to analyze data, recognize patterns, and make predictions.

Additionally, a specialized form of ML known as deep learning (DL) has gained prominence. DL involves the use of deep neural networks (DNNs), composed of multiple layers of interconnected nodes, to analyze complex data. DL algorithms can be used for tasks such as image and speech recognition, natural language processing, and autonomous decision-making. However, these networks require a large amount of data to learn complex patterns and become robust to a wide variation of new data.

As the demand for efficient and accurate monitoring solutions increases, intelligent models like neural networks are gaining popularity for automating tasks traditionally carried out by humans. Thus, systems such as traffic monitoring and surveillance have integrated with these models to better meet the demand, as vast amounts of data come from security cameras and are difficult to be meticulously monitored by humans.

Intelligent surveillance systems can encompass a variety of features, including motion detection, facial recognition, object tracking, and automatic alerting. They can be applied to various applications, from traffic monitoring to detecting potential security threats in public spaces. One of the main benefits of using intelligent systems is the ability to quickly and accurately analyze large volumes of video data. Using AI and ML algorithms, these systems can detect patterns and anomalies that might go unnoticed by human operators. Additionally, they also enable reducing the workload of human operators, freeing them to focus on more complex tasks.

Moreover, analyzing how fires propagate is a critical aspect of fire detection and surveillance systems, as it enables the development of more effective strategies for fire prevention and control. For instance, understanding how a fire spreads allows for designing buildings and structures to avoid such accidents. Furthermore, analyzing fire propagation can aid emergency services in responding more effectively to fire emergencies by rapidly

identifying the fire's origin, assessing its severity, and mobilizing appropriate resources to contain and extinguish it. By gaining a clear understanding of fire spread, it is possible to minimize the risk to human life and property, as well as reduce the damages caused by fires.

In the context of emergency management, natural disasters, accidents, and emergencies occur frequently and have severe impacts. To prevent or mitigate these issues, various sensors, alarms, and security systems are implemented. Fires are incidents that can spread rapidly and pose a significant threat to lives and properties, thus addressing them in their early stages is crucial. According to a natural disaster report (RELIEFWEB, 2021), there were 19 forest fires in 2021, resulting in the loss of 128 human lives and 9.2 billion dollars. Additionally, the U.S. Fire Administration (USFA) states that there were 1,291,500 fires, 3,704 deaths, 16,600 injuries, and 14.8 billion in losses in the year 2019 in the United States (U.S. FIRE ADMINISTRATION, 2022). Festag (2016) investigated the rate of false alarms from sensitive sensors in Germany, which averages 86.07% per year, highlighting the inaccuracy of such systems. This underscores the need for autonomous systems to detect fires more efficiently so that firefighters can be dispatched promptly.

Traditional fire alarms are widely used to prevent significant losses. However, these solutions are based on optical and infrared sensors, requiring them to be placed near the fire, which is challenging in open areas. Furthermore, not all fires occur when someone is nearby, thus human monitoring is recommended to confirm the fire and assess its severity. Additionally, these systems are expensive to install and maintain and have a high false alarm rate.

As an alternative, other vision-based sensors can provide more information, such as the fire's location and severity. For instance, fire segmentation can offer the fire's size and location, enabling the assessment of growth rate through the analysis of a sequence of images.

A non-real-time fire detection and segmentation method can also be valuable for analyzing how fire spreads across a set of images. Analyzing a sequence of fire images can provide valuable insights into fire behavior and the effectiveness of firefighting efforts.

The problem addressed by this work involves not only identifying the presence of

fires in images but also locating the fire through pixel-level clustering.

Delving into the specifics, image segmentation emerges as a crucial aspect of the solution. Segmentation is one of the most well-known and longstanding problems in computer vision (CV), involving grouping pixels that belong to the same class or exhibit similarity in a given context (SZELISKI, 2011). The result achieved in this problem, where objects in the image are divided into classes, is commonly referred to as a mask. Figure 1.1 illustrates an example of an annotated image for semantic segmentation, where each object class is color-coded.

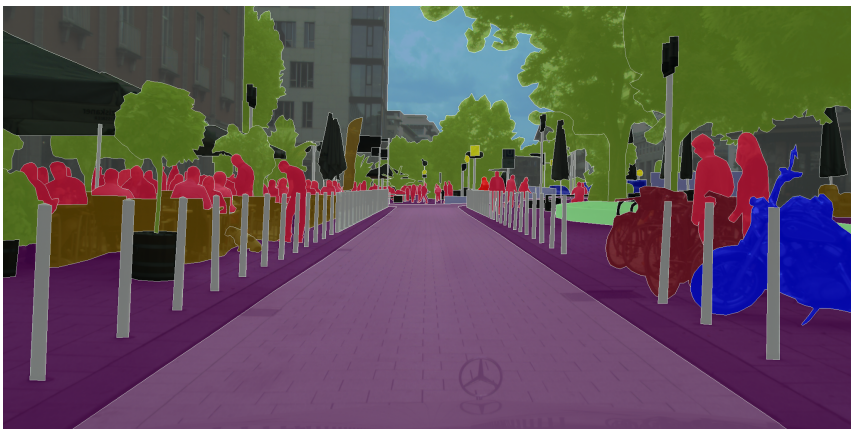


Figure 1.1: Example of semantic segmentation annotation taken from the Cityscapes dataset (CORDTS et al., 2016).

Therefore, segmentation techniques can be applied to the fire detection problem, providing much more information than just image classification with or without fire. The problem in fire detection can be described as separating fire pixels from non-fire pixels in a series of images. Once fire pixels are segmented, it is possible to track the progression of fire over time by analyzing changes in fire pixels from one image to the next. This approach can be particularly useful for understanding how a fire spreads across a large area or behaves under different conditions. By analyzing a series of images captured over time, patterns in fire spread can be identified, and factors like wind, temperature, and fuel load can be assessed for their impact on fire behavior.

Image processing techniques for segmentation range from classical algorithms like thresholding and edge detection to deep neural network models for semantic segmentation. However, as fire is not an easily distinguishable object for algorithms, the task requires more complex methods that can leverage color and geometry information. Moreover,

the fact that the color of fire can be present in a vast variety of other objects makes the problem even more challenging. In essence, across the entire color space, one needs to identify the most common fire colors while differentiating them from any other object sharing similar characteristics. As shown in Figure 1.2, the goal is to obtain a fine-grained mask that precisely segments the fire.

(a) Input image. (b) Ground truth.

Figure 1.2: Example of binary segmentation for fire taken from our test set (CHINO et al., 2015).

1.2 Motivation

Fires pose a significant threat to both human lives and properties, and swiftly and accurately detecting and mitigating them is crucial. The state of the art lacks methods with balanced classification, meaning that achieving good metrics for both the True Positive Rate (TPR) and the False Positive Rate (FPR) remains a challenge. Furthermore, the best results are often achieved by identifying what is not fire, the negative and majority class in the images (CHINO et al., 2015). For the fire detection problem, it is more critical not to miss fire cases, meaning false alarms are less detrimental than undetected fires. As a result, there is a pressing need for new fire detection and segmentation methods capable of better classifying what constitutes fire — the positive and most important class — and maintaining balanced results.

The benefits of such a system can be immense. In the short term, it can assist first responders in swiftly and accurately pinpointing the location and extent of fires, enabling them to respond more effectively and potentially save lives and properties. In the long term, it can enable better fire prevention strategies, offering more precise data about fire

patterns and behaviors when utilized to analyze sets of images and make data-driven decisions.

In addition, accurate fire detection and segmentation can have broader societal implications. For instance, it can aid in the development of more effective climate models by providing data about the frequency and intensity of fires in different regions. It can also assist in identifying high-risk fire areas, allowing relevant authorities to take proactive measures to mitigate the risk.

Overall, there is a clear need for new and improved fire detection and segmentation methods, and the potential benefits of such a system are significant. With the power of DL and CV techniques, it might be possible to develop a more accurate and reliable system for identifying and tracking fires, with wide-ranging implications for emergency response, fire prevention, and societal well-being. While this method may not be a real-time application, it can provide valuable information for analysis and informing future strategies for dealing with fires. By understanding how fires spread and behave, it is possible to develop more effective fire prevention and suppression strategies, reducing the risk of loss of lives and properties due to fire.

1.3 Objectives

The main objective of this work is to develop a novel method for the balanced detection and segmentation of fire in images. The aim is to create a model capable of discerning both fire and non-fire areas, while having a higher emphasis to the positive class (fire). The exclusion of scenes without any detected fire prevents unnecessary processing and allows a more effective segmentation.

Subsequently, the method will be evaluated using the same dataset and image segmentation metrics employed in the investigations of related works. The research scope includes evaluating the use of deep neural networks (DNNs), CV methods, and ML to generate fire segmentation masks.

Hence, the objectives of this study include the following steps:

- Investigating the use of state-of-the-art neural networks for classification of fire

images and regions;

- Exploring various datasets utilized in related works;
- Developing a method for a patch-based segmentation, identifying fire regions in images;
- Experimenting with the use of a classification model prior to the segmentation stage, and assessing its impact on final results;
- Establishing a color range based on color distribution in the training set for constructing the color classifier;
- Examining color thresholding in both RGB and HSV color spaces;
- Experimenting with the use of graph cuts and their energy functions;
- Assessing final results by comparison with state-of-the-art methods for the given problem.

1.4 Contributions

The primary contributions of this study lie in advancing the domain of fire detection through an innovative methodology harnessing the capabilities of graph cuts for refined fire segmentation. The outcomes not only showcase the highest TPR when compared to contemporary state-of-the-art methods but also manifest enhancements in FPR, precision, and F_1-score in contrast to our prior research. This progress underscores the efficacy and promise of our approach in precisely detecting fires, reducing false alarms, and elevating overall performance. The significant improvements achieved resulted in the publication in two renowned conferences (PEREIRA; VIEIRA; VILLELA, 2022; PEREIRA; VIEIRA; VILLELA, 2023).

1.5 Organization

This work is organized as follows: Chapter 2 describes the fundamental concepts applied in the proposed approach. Chapter 3 summarizes how related works have addressed the fire detection problem. Chapter 4 details our method, including the integration of deep neural networks, graph cuts, and color thresholding. Chapter 5 describes the datasets used, the experimental setup, and presents the results of our experiments. Finally, Chapter 6 concludes the work and outlines future research directions.

2 Theoretical Framework

This chapter provides the necessary concepts to understand the CV techniques and DL models used in developing the method proposed in this work. Throughout the research, the use of each method and combinations among them will be evaluated. Firstly, the functioning of the k-means algorithm is presented, which is widely used for clustering tasks and also applicable for identifying clusters in images. Following that, the interpretation of images as graphs and the utilization of the classic graph cut method for object segmentation based on color and geometry similarity are detailed. Subsequently, the field of DL is introduced, discussing neural networks, their training, and potential training issues. Convolutional neural network (CNN) models, specialized neural networks for CV problems, are then described. Lastly, the emerging architecture in the natural language processing (NLP) field, also recently employed in CV, the transformers architecture, is discussed.

All these concepts are of paramount importance for the development of this work. They are directly related to CV problems, particularly image segmentation.

2.1 K-means

K-means is an unsupervised clustering algorithm used to partition a given dataset into k clusters, where k is the number of clusters specified by the user. It functions iteratively by assigning each data point to the cluster whose centroid is closest to it and then updating the centroid of each cluster based on the assigned data points (SZELISKI, 2011). The process continues until the centroids no longer move significantly or a maximum number of iterations is reached.

In the context of image segmentation, k-means can be employed to group similar pixels in an image into k clusters, where each cluster represents a different region in the image. Once clustering is complete, the resulting clusters can be used to segment the image by assigning each pixel to the corresponding cluster. This can be useful in

applications where regions containing the object of interest in an image can be separated from the background. However, k-means does not always produce the best segmentation results, as it assumes well-separated clusters. Figure 2.1 presents an example of color quantization using k-means.

Figure 2.1: K-means quantization example with k varying from 2 to 8. Source: (OPENCV, n.d.).

2.2 Graph Cut

Graph cut is a widely used technique in computer vision to solve optimization problems that can be formulated as graph problems. In the context of image segmentation, graph cuts can partition an image into foreground and background regions, minimizing an energy function (SZELISKI, 2011).

This energy function represents the cost associated with labeling each node in a graph. The energy function is typically defined as a combination of two terms: the term measuring similarity between input data and labels, and the smoothness term that encourages neighboring nodes in the graph to have similar labels. The goal of graph cuts is to find the labeling that minimizes the energy function, which can be formulated as an optimization problem and solved using efficient algorithms such as the maximum flow and minimum cut algorithms.

The maximum flow and minimum cut problem is straightforward: it involves sending the maximum possible flow between two specified nodes in a network (source and sink) without exceeding edge capacities (AHUJA; MAGNANTI; ORLIN, 1993). Algorithms to solve the problem model the network as a graph, with edges representing network connection capacities. The algorithm then tries to find the cut in the graph that minimizes the capacities of the edges crossing the cut, which is equivalent to finding the maximum flow.

The core idea behind graph cuts is to convert the image segmentation problem into a graph problem, where nodes represent image pixels and edges represent relationships between neighboring pixels. The weight of an edge represents the dissimilarity between the two connected nodes.

The segmentation problem is then formulated as a minimum cut problem in the graph, where the cut separates nodes into two sets of regions: the object of interest and the background. Figure 2.2 illustrates the graph cut process.

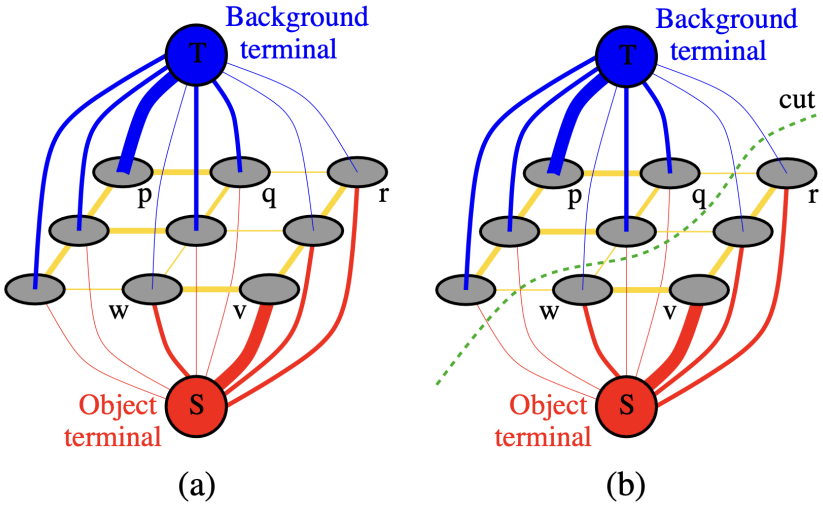


Figure 2.2: Graph cut applied in image segmentation (BOYKOV; JOLLY, 2001): (a) the graph is constructed from image pixels; (b) the image is partitioned based on the minimum cut. Source: Szeliski (2011).

2.3 Deep Learning

DL is a subfield of ML that involves the utilization of deep neural networks to learn complex and nonlinear mappings between inputs and outputs. Neural networks are com-

putational models inspired by the human brain and consist of multiple layers of interconnected processing units called neurons (AGGARWAL, 2018). These models are particularly well-suited for handling complex and high-dimensional data, where traditional algorithms might struggle to identify patterns or make accurate predictions. They are also employed in tasks that require learning from large datasets, as they can automatically extract features and learn representations that enhance their performance.

Neural networks function by taking an input vector and passing it through a series of nonlinear transformations, with the output of each layer serving as the input to the next layer. The final output of the neural network is generated by the output layer, which typically employs a nonlinear activation function that maps the output of the final hidden layer to the desired output space. Such networks are commonly known as feedforward neural networks or multilayer perceptrons (MLPs), as data flows in only one direction, from the input layer through the hidden layers to the output layer. The term "feedforward" indicates that input data is fed into the network in a single pass without any feedback connections. Therefore, the goal of a feedforward network is to approximate a function f that maps an input $\mathbf{x}$ to an output value (GOODFELLOW; BENGIO; COURVILLE, 2016). Figure 2.3 depicts the architecture of a basic single-hidden-layer neural network. The operations illustrated in the output neuron, including weighted sum, bias addition, and activation, are likewise performed in all neurons a_i within the hidden layer.

Bias is a scalar value added to the weighted sum of a neuron's inputs. It is an additional parameter in the neural network that enables the model to shift its output in a specific direction. The bias term assists in shifting the activation function of a neuron in either the negative or positive direction. Thus, the following equation describes the computation performed in each neuron:

$$y = \sum_{i=1}^{n} w_i x_i + b, \tag{2.1}$$

where w_i represents the weight associated with the i-th input x_i, and b is the bias. A subsequent activation function is applied, such as the sigmoid or Rectified Linear Unit (ReLU), to capture the data's nonlinearity. Unlike sigmoid and other slower computing

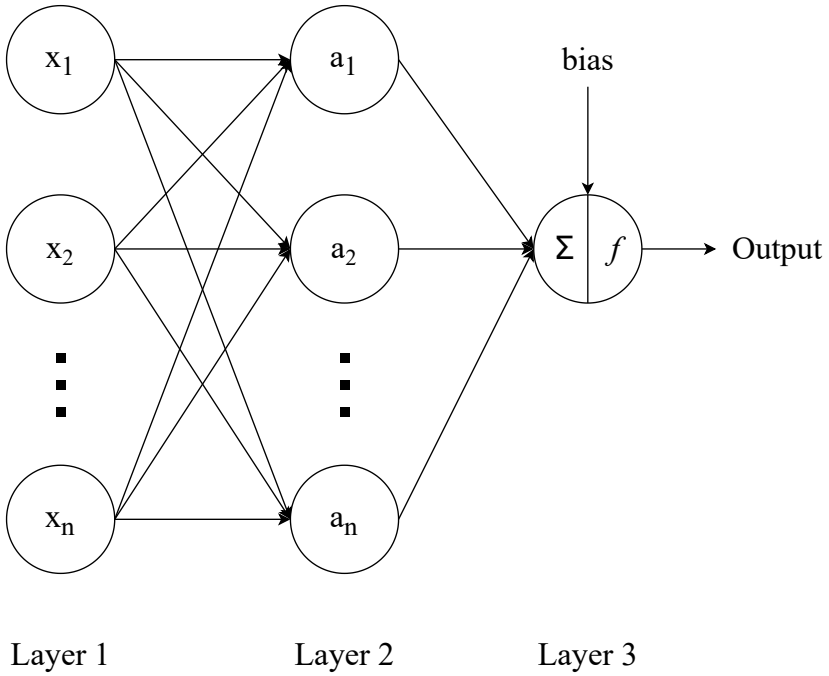


Figure 2.3: Architecture of a single-hidden-layer neural network.

functions, the ReLU activation function is simple and has proven effective in training deep neural networks (LECUN; BENGIO; HINTON, 2015). The ReLU function is defined as:

$$f(x) = \max(0, x), \tag{2.2}$$

where x is the input to the function. This function returns zero for negative inputs and the input value itself for non-negative inputs. One of the main advantages of the ReLU function is its computational efficiency.

The process of training a neural network involves feeding it with a dataset comprising inputs and expected labels for each entry, and adjusting the neuron weights to minimize the difference between the predicted and actual output. Learning occurs by adapting the network weights according to a loss function, which is achieved using a technique called backpropagation (RUMELHART; HINTON; WILLIAMS, 1986). Backpropagation utilizes gradient descent to minimize the loss in its classical version. The algorithm calculates the gradient of the loss function with respect to the network's weights and biases and uses it to update the parameters. The learning rate is a hyperparameter used in backpropagation, controlling the extent to which weights are updated in the gradient direction. A very high learning rate can cause the optimization process to diverge,

while a very low learning rate can lead to slow optimization and being trapped in a local minimum. Therefore, finding the ideal learning rate is crucial for effectively training a neural network. The use of adaptive learning rates is a popular practice to adjust the learning rate during training.

The primary challenge in ML is to create models that generalize well, meaning they perform well on new datasets (GOODFELLOW; BENGIO; COURVILLE, 2016). However, during the training process, it is important to be aware of the possibility of overfitting or underfitting the data. Overfitting occurs when the neural network is too complex for the provided data or is trained too well to the point that it starts memorizing training examples instead of learning patterns in the data, leading to poor generalization to new data. On the other hand, underfitting occurs when the neural network is not complex enough or has not been trained sufficiently to capture patterns in the data, resulting in low performance on both training and test data.

To address overfitting, regularization techniques such as L_1 and L_2 regularization, dropout (SRIVASTAVA et al., 2014), and data augmentation can be used to constrain the network weights and prevent it from memorizing noise in the training data. Dropout, for example, randomly selects and drops some neurons in a network layer during each training iteration. This forces the remaining neurons to learn more robust and generalized representations.

Conversely, the neural network can be made more complex by increasing the number of layers or neurons in the network or changing activation functions to capture more complex patterns in the data to combat underfitting. However, it is important to avoid making the network overly complex, as it may lead to overfitting. Appropriate validation techniques, such as cross-validation, can help identify and address overfitting and underfitting.

A widely applied function at the output of neural networks is the softmax, often used in classification tasks to convert output values from the previous layer into probabilities for multiple possible classes (AGGARWAL, 2018). This probability distribution can be used to determine the most likely class for a given input. The softmax function is

defined as follows:

$$\text{softmax}(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{K} e^{x_j}}, \tag{2.3}$$

where x_i is the input to the i-th neuron in the output layer, K is the number of classes, and e is the base of the natural logarithm. The denominator of the softmax function is the sum of the exponential values of all inputs, ensuring that the output is a valid probability distribution with values between 0 and 1 that sum to 1. The fact that the output of the softmax function is a probability is valuable, as it allows understanding the neural network certainty level in its decisions (NIELSEN, 2015). Consequently, this probability can be utilized to evaluate the network using different thresholds to determine its output, i.e., defining from which probability the model's response for a specific class is accepted. This is useful for enhancing a model's predictions when it excels for a specific class and has not captured the pattern of others as effectively.

2.4 Convolutional Neural Networks

DL has become a popular technique for CV tasks such as image classification, object detection, and segmentation due to its ability to automatically learn complex visual features and patterns from vast amounts of data (LECUN; BENGIO; HINTON, 2015). Convolutional Neural Networks (CNNs) are a type of neural network designed specifically for CV tasks (GOODFELLOW; BENGIO; COURVILLE, 2016) and are increasingly prominent in the state of the art. These networks have revolutionized the field of CV, enabling accurate and efficient image classification, object detection, and segmentation. The distinctive component of CNNs is the convolutional layers, which perform the convolution operation on input data to extract features.

Convolution is a mathematical operation involving sliding a small window (referred to as a kernel or filter) over the input data, multiplying the values in the window by corresponding values in the filter, and summing the results (SZELISKI, 2011). The output of this operation is a single value representing the degree of similarity between the input data and the filter. By applying this operation to different regions of the input data, a feature map that captures various aspects of the input image can be generated.

Figure 2.4 demonstrates the convolution operation, with the blue pixels indicating the neighborhood in which the filter is applied in this step, generating the green pixel.

40	61	103	123	127	131	141	130
43	66	87	123	125	128	140	131
45	65	88	118	121	123	138	145
45	60	79	119	129	135	142	133
49	59	68	87	115	127	135	134
48	67	58	77	96	114	121	133
52	52	57	65	69	95	109	132
53	51	55	58	66	88	99	111

*

0.1	0.1	0.1
0.1	0.2	0.1
0.1	0.1	0.1

=

66	92	114	124	130	135
64	89	111	124	130	135
62	82	104	120	130	135
59	74	92	111	124	131
58	65	77	94	110	122
54	60	67	80	95	111

Figure 2.4: The image on the right is obtained by performing convolution of the filter over the image on the left.

Convolutive layers in CNNs apply these filters to input data throughout the network (GOODFELLOW; BENGIO; COURVILLE, 2016), generating feature maps that capture different types of features. These feature maps are passed through activation functions and optionally through pooling layers to reduce sample and data dimensionality. The values filling the filters in the convolution operations are learned by the network during training, allowing it to decide on the best values for certain features across its layers. Figure 2.5 presents the architecture of LeNet-5 (LECUN et al., 1998), the first proposed CNN.

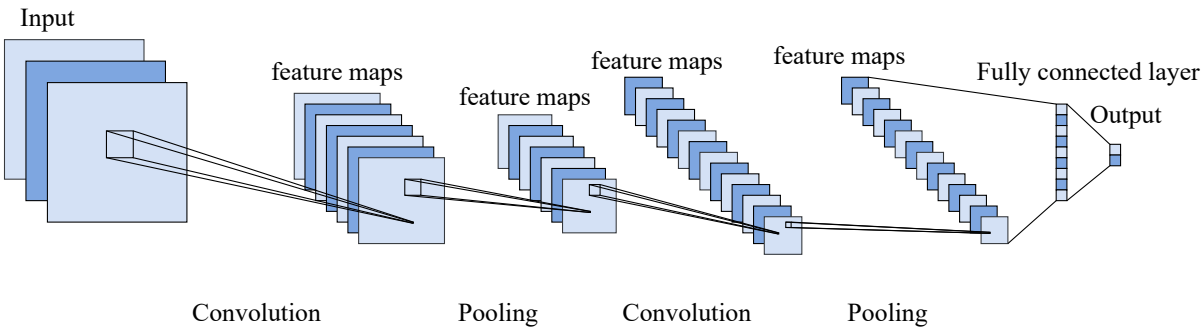


Figure 2.5: General architecture example of a CNN.

Pooling in CNNs is an operation for downsampling that reduces the spatial dimensions (height and width) of feature maps produced by convolutional layers and is a technique used in most CNNs (GOODFELLOW; BENGIO; COURVILLE, 2016). It is typically applied after the convolutional layer and the activation function. The pooling operation works by dividing each feature map into non-overlapping regions, referred to as

pooling regions or windows. Then, for each region, it calculates a single value representing the maximum (max pooling) or the average (average pooling) of the values in that region. This value is used to replace the entire window in the output feature map.

This operation is employed to reduce computational cost by decreasing the number of parameters and input data size, aiding in preventing overfitting. Furthermore, it helps to extract the most relevant features from the data. Figure 2.6 illustrates the max pooling operation.

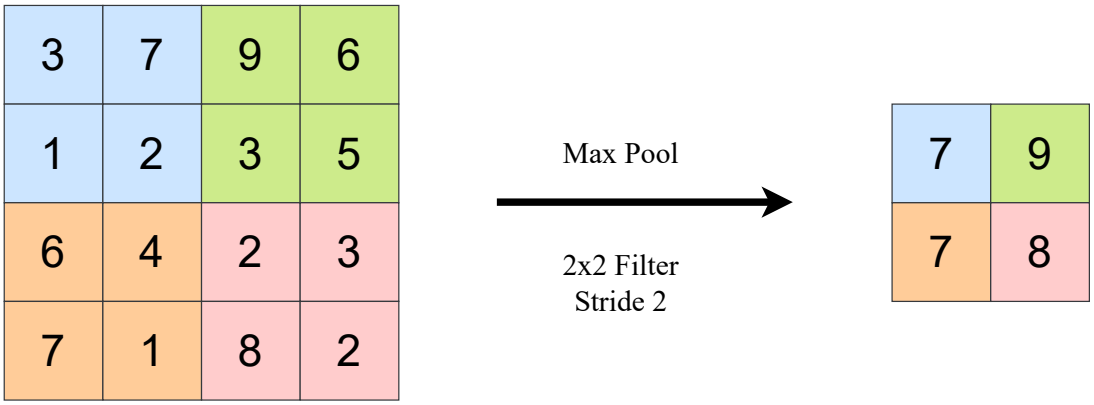


Figure 2.6: Max pooling operation applied with a 2x2 filter (window) and 2-pixel strides.

In a CNN, the initial layers learn low-level features such as edges and corners, while deeper layers learn more complex features composed of these low-level features, such as geometric shapes (AGGARWAL, 2018). This hierarchical structure allows CNNs to effectively model complex relationships in input data and achieve high accuracy.

2.5 Transformers

Transformers (VASWANI et al., 2017) are a type of neural network architecture that has revolutionized Natural Language Processing (NLP) tasks. At the heart of the transformer architecture lies the self-attention mechanism, which enables the model to attend to different parts of the input sequence when generating an output. Self-attention calculates a weighted sum of the input sequence at each position, with weights determined by the similarity between the current position and all other positions in the sequence. The self-attention mechanism is used in both the encoder and decoder components of the transformer architecture. The encoder takes an input sequence of tokens (e.g., words)

and generates a sequence of embeddings, each representing a different position in the input. The decoder takes as input the encoder's output and its own previous decoder block outputs, producing an attention vector as output.

Each layer in the transformer consists of a multi-head self-attention mechanism followed by a feedforward neural network. The multi-head attention mechanism allows the model to attend to the input sequence in multiple ways, with each head learning a different attention distribution over the sequence.

In addition to self-attention, the transformer also includes positional encoding, used to provide the model with information about the position of each token in the input sequence. This is crucial because the self-attention mechanism only considers token similarity and not their positions.

This architecture has demonstrated remarkable performance across a wide range of NLP tasks, including machine translation, text summarization, and language modeling. Its success is largely attributed to its ability to capture intricate patterns within data. Figure 2.7 illustrates the architecture of a transformer.

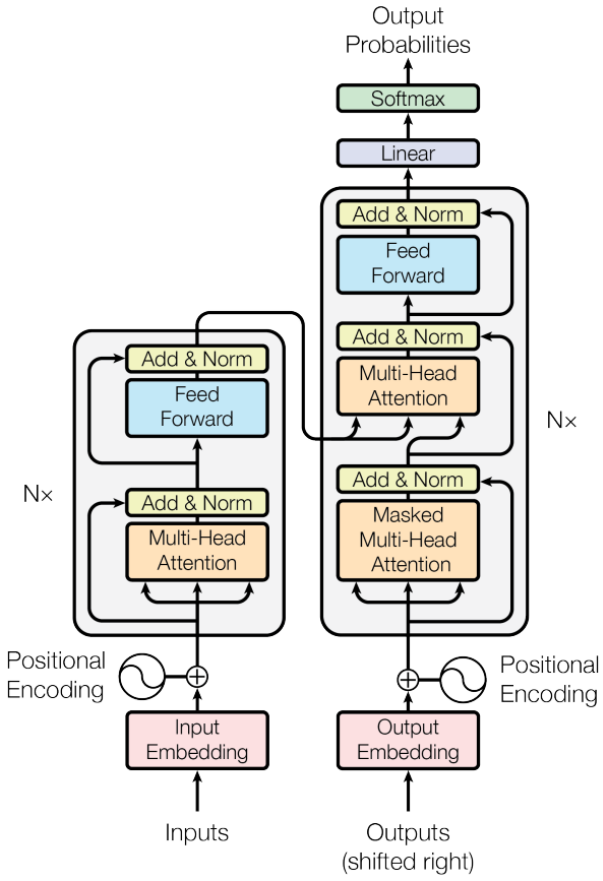


Figure 2.7: Architecture of a transformer model. Source: Vaswani et al. (2017).

Transformer networks have excelled in NLP tasks and have sparked significant interest in the field of computer vision. However, visual data demands specific network architectures and training methods. Therefore, various authors have implemented their versions of transformer models for vision tasks. Recently, the paper proposing the Vision Transformer (ViT) model (DOSOVITSKIY et al., 2020) experimented with using a standard transformer with minimal modifications. To achieve this, they treated small patches of the image as input tokens, flattened these patches, and added learned positional embeddings. Although ViT achieves remarkable results on recognition benchmarks when pre-trained on the JFT-300M dataset (SUN et al., 2017), it still struggles with limited data, falling behind CNNs in such cases. To overcome this issue, CoAtNet proposes a combination of deep convolution and self-attention in an attempt to merge CNN's generalization with the Transformer model's capabilities, as Transformers have shown a higher ceiling. To this end, their architecture is divided into 5 stages, 1 Convolution block, 2 Mobile Inverted Residual Bottleneck Convolution (MBConv) blocks, and 2 transformer blocks, as shown in Figure 2.8.

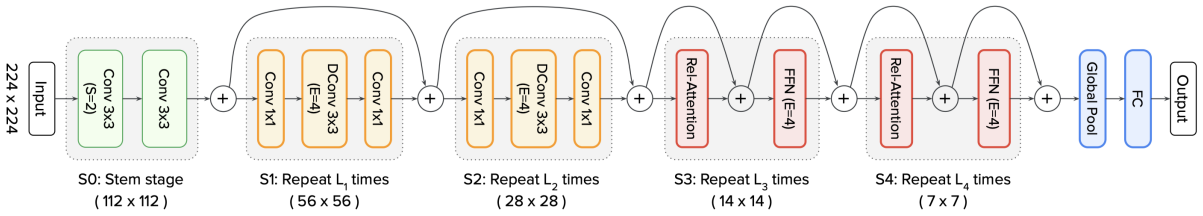


Figure 2.8: CoAtNet architecture overview. Source: Dai et al. (2021).

Unlike convolutional blocks, MBConv blocks (SANDLER et al., 2018) are based on the idea of inverted residuals, involving the use of 1×1 convolutions to expand the number of channels, followed by 3×3 depthwise convolutions, and then another 1×1 convolution to compress the output. This approach reduces the number of parameters and computational complexity while maintaining high accuracy. Additionally, a residual connection adds the input's feature maps to the block's output.

The idea of residual blocks (HE et al., 2015) stems from the difficulty of training deep neural networks due to the vanishing gradient problem, where gradients can become too small as they propagate through the network. Residual blocks were introduced to address this by introducing shortcut connections that allow gradients to flow directly to

earlier layers.

These shortcut connections enable residual blocks to learn a residual mapping, or the difference between the input and the block's output. This approach has proven to ease the training of very deep neural networks, allowing them to learn more complex functions without succumbing to the vanishing gradient problem.

3 Related Work

Fire detection can be achieved in numerous ways, depending on design decisions that define the main objectives. For instance, if the goal is simply to identify and report fires, the problem may boil down to image classification or object detection. There is a wide range of studies addressing the fire detection problem, and thus, the solutions proposed by the authors also vary.

Researchers who began exploring the use of vision for fire detection focused their studies on using motion and temporal features in videos (KIM; KIM; JEONG, 2014) or investigated using color information through different color spaces (CELIK; DEMIREL, 2009; CHEN; WU; CHIOU, 2004).

More recent works have implemented Convolutional Neural Networks (CNNs) in their approaches, utilizing them for image classification and fire localization. These have shown promising results by combining the power of CNNs with superpixel clustering (THOMSON; BHOWMIK; BRECKON, 2020) or by extracting feature maps from models to generate a mask (MUHAMMAD et al., 2019).

In this chapter, different methods used by state-of-the-art studies will be presented, as well as the proposed approaches that have served as baselines for various articles.

3.1 Video-Based Fire and Smoke Detection

Chen, Wu and Chiou (2004) propose a video-based approach. The algorithm begins by segmenting moving regions from captured image sequences, which are considered candidates for fire and smoke detection. Chromatic features are employed to extract fire pixels and smoke pixels from these moving regions.

To differentiate fire and smoke pixels from corresponding false positive pixels, dynamic features like growth and clutter are employed. These features validate the extracted fire and smoke pixels. It is important to note that if both fire and smoke pixels

satisfy dynamic features, it indicates the presence of a real fire. However, if only fire pixels satisfy these dynamic features, it may suggest the burning of certain fuels that generate nearly transparent smoke, which is not captured by the video camera.

This method combines moving region segmentation, extraction of fire and smoke pixels based on chromatic features, validation through dynamic features, and evaluation of fire alarm trigger conditions to achieve early fire detection. Experimental results demonstrate the effectiveness of the proposed algorithm.

Chino et al. (2015) constructed a dataset and conducted evaluations, comparing various methods. When assessed using the dataset introduced by Chino et al. (2015), the proposed method showed satisfactory results in terms of FPR. Nevertheless, it is crucial to emphasize that the effectiveness of the method in detecting fires was limited.

One of the advantages of the method is its video-based approach, allowing the detection of moving regions and identifying candidates for fire and smoke presence. Additionally, the use of chromatic features for extracting fire and smoke pixels is an interesting strategy. The validation of these pixels through dynamic features such as growth and clutter also contributes to real fire detection.

However, the method has a significant limitation. It is not effective for fire detection, being more efficient in smoke detection. This could be attributed to the lack of consideration for other relevant aspects of fire detection, such as temperature variation and flame patterns. Additionally, the incapacity to capture nearly transparent smoke generated by certain fuels is a drawback, as it can lead to false negatives and compromise detection accuracy.

Despite these limitations, the method showed good results in terms of FPR when evaluated on a specific dataset. This indicates that the algorithm might be useful in specific contexts or scenarios where smoke detection is the primary concern. However, for comprehensive fire detection, it is necessary to consider other approaches or supplement the proposed method with other techniques that address the identified limitations.

3.2 Stereo Vision-Based Fire Segmentation

Rossi, Akhloufi and Tison (2011) introduce in their paper a novel stereo vision-based instrumentation system for fire segmentation in outdoor conditions. In the proposed approach, images are captured and processed using specialized algorithms. These algorithms enable the modeling of fires in 3D and the extraction of geometric features such as volume, surface area, motion direction, and length. Experiments were conducted in outdoor scenarios, and the obtained results showcase the effectiveness of the proposed system.

The method proposed extracts geometric features of fire from videos using stereo vision. The method involves a clustering approach to locate the fire region. Initially, the image is divided into two clusters based on the V channel of the YUV color space. The cluster with the highest V value is identified as the fire cluster.

To classify pixels more accurately, a 3D Gaussian model is employed. This model assigns a probability to each pixel based on its color and spatial position information. Based on these probabilities, the method can differentiate between fire pixels and non-fire pixels. It is important to note that this method was specifically developed for fires in controlled environments and might have limitations in outdoor fire emergency situations, as discussed by Chino et al. (2015).

When analyzing the results of Rossi, Akhloufi and Tison (2011) on the dataset proposed by Chino et al. (2015), a slight improvement in TPR can be observed with the application of the proposed method. However, it is crucial to emphasize that the results still do not reach levels considered satisfactory.

A benefit of this method is the use of stereo vision, which enables the extraction of geometric features of fire such as volume, surface area, motion direction, and length. Furthermore, the method employs a 3D Gaussian model to classify pixels more accurately, taking into account color and spatial position information. The results obtained from experiments demonstrate the effectiveness of the proposed system.

Nevertheless, it is important to consider some limitations of the method. It was specifically developed for fires in controlled environments, which means it might not be as effective in outdoor fire emergency situations. Additionally, despite showing an improvement in the detection rate compared to other methods, the results still fall

short of levels considered satisfactory, as discussed by Chino et al. (2015). Therefore, while the proposed method has its advantages, improvements are necessary to enhance its robustness and applicability in different fire scenarios.

3.3 Clustering-Based Fire Detection

Rudz et al. (2013) proposed another clustering-based method. They calculate four clusters using the Cb channel of the YCbCr color space. The one with the lowest Cb value is classified as the fire region. Subsequently, false positive pixels are discarded using a reference dataset. The method handles small and large regions differently: small regions are compared with the average of a reference region, and large regions are compared with the reference histogram. They execute this process for each color channel.

The results on the dataset from Chino et al. (2015) demonstrated an improvement of over 20% in TPR. However, they still have not reached a significant level of TPR.

An upside to this strategy is that the method utilizes the Cb channel of the YCbCr color space to identify fire regions, which can be effective in detecting fires under specific lighting conditions. Additionally, the method handles small and large regions differently, allowing for a more precise analysis of fire characteristics.

Even so, the method also has some problems. For instance, classifying the fire region based on the lowest Cb value can lead to false positives in situations where other elements in the scene possess similar Cb values. Furthermore, the discarding of false positive pixels relies on the availability of a reference dataset, which may restrict its applicability to specific scenarios where such data is available.

Another consideration is that the method performs the detection process for each color channel separately, which can increase computational complexity and processing time. Therefore, while the method presents interesting approaches for fire detection, it is necessary to evaluate its effectiveness under different conditions and consider its limitations for proper application.

3.4 BoWFire: A Color and Texture-Based Fire Detection Method

The method proposed by Chino et al. (2015) consists of three steps: generating a segmentation mask through a color classifier, generating a second mask through a texture classifier, and finally, performing an intersection between the two produced masks.

The color classification step aims to differentiate fire regions from non-fire regions based on color information. Color classification is performed in the YCbCr color space, as it provides better discrimination for fire colors (CHINO et al., 2015). The process begins by converting each pixel of the input image to the YCbCr color space. The pixel values are represented as (Y_i, Cb_i, Cr_i), where Y_i represents the luminance component and Cb_i and Cr_i represent the chrominance components.

A pixel color classification is then applied to each converted pixel using a training set of colors and a Naive Bayes classifier. The color training set consists of labeled examples of "fire" and "non-fire" pixels, which are used to train the color classifier.

If the color classifier categorizes a pixel as fire, the pixel is considered part of the fire region and is used to construct the output mask. Otherwise, the pixel is discarded. The use of a color classification step in the BoWFire method avoids the need for a large number of parameters. By classifying each pixel individually, the method achieves a finer granularity in its segmentation mask. However, this also results in false positives due to the lack of consideration of neighborhood information.

The texture classification step aims to improve the accuracy of fire detection by considering local image characteristics. Since different emergency situations can result in various types of fire images, global features might not efficiently capture small fire regions. Using the Simple Linear Iterative Clustering (SLIC) algorithm, the authors divide the image into superpixels. Superpixels are image regions obtained by grouping pixels based on their similarity in color, texture, or other image attributes. The goal of superpixel segmentation is to divide an image into visually coherent and meaningful regions, reducing the complexity of subsequent image processing tasks. For each superpixel, a feature extraction process is applied to extract a feature vector, and finally, the k-NN algorithm

with Manhattan distance classifies the found features, determining whether the superpixel is a fire region or not.

If the feature classifier categorizes the superpixel as a fire region, all pixels belonging to the superpixel are considered part of the fire region and are used to construct the output mask. Conversely, if the feature classifier does not categorize the superpixel as a fire region, the pixels of that superpixel are discarded.

By considering local texture features, the BoWFire method efficiently captures specific patterns related to fire regions, enhancing the accuracy of fire detection.

A positive aspect of this method is the color classification step, which uses the YCbCr color space to discriminate fire regions from non-fire regions based on color information. This allows for more accurate detection, as fire colors are better discriminated in this color space. Furthermore, color classification is carried out individually on each pixel, providing a finer granularity in the segmentation mask.

Another positive aspect is the texture classification step, which aims to improve fire detection by considering local image characteristics. The use of superpixels, obtained through the SLIC algorithm, allows for pixel grouping based on color and texture similarity, capturing specific patterns related to fire regions. This increases the accuracy of fire detection, especially in situations where global image features would not be effective.

Nonetheless, it is important to highlight that the BoWFire method also relies on accurate superpixel segmentation and correct texture feature extraction. If there are inaccuracies in these steps, fire detection can be compromised.

The results obtained in this study demonstrated a significant improvement compared to previous works. Both the isolated color classification approach and the combination of texture and color showed superior performance. These results underscore the effectiveness of the proposed method compared to previous approaches.

3.5 Deep Learning-Based Fire Detection and Localization with CNNs

In more recent works, with the advent of the field of DL, new approaches have emerged proposing the use of Convolutional Neural Networks (CNNs) for fire detection. Muhammad et al. (2019) explore in their work the detection and localization of fires through networks based on SqueezeNet (IANDOLA et al., 2016) and AlexNet (KRIZHEVSKY; SUTSKEVER; HINTON, 2012). The model is trained to detect fires in various scenarios, including indoor and outdoor environments, without the need for pre-processing or selecting more relevant features in the images. The trained model assigns a label to input images based on probability scores calculated by the network.

If the network does not detect a fire in the image, the image is simply discarded. Otherwise, the fire localization step is carried out, and the segmentation mask is produced. To locate the fire in an image, the process involves additional processing. The *feature maps* from different layers in the CNN are analyzed to identify maps sensitive to fire regions. These maps are combined and binarized to segment the fire, resulting in a binary mask that indicates the fire region.

The segmented fire region is then used for two purposes. Firstly, the severity level or degree of burning of the fire is determined based on the number of pixels in the segmented region. Secondly, the Zone of Influence (ZOI) is found by subtracting the segmented fire regions from the original input image. The resulting ZOI image is passed through the original SqueezeNet model, which predicts its label from 1000 objects. This information helps identify the situation associated with the fire, such as a fire in a house, forest, or vehicle. The severity of the fire and situational information can be reported to the fire department for appropriate actions.

Overall, the proposed method employs a deep CNN to detect and localize fires in images, eliminating the need for manual pre-processing or feature engineering. It provides fire detection and localization, as well as additional information about the severity and situation of the fire. The method outperformed the state of the art at the time, showing better precision and recall results while maintaining low computational cost.

On one hand, the use of Convolutional Neural Networks (CNNs) allows the model to automatically learn relevant features for fire detection. This facilitates the application of the method in different scenarios, both indoors and outdoors.

On the other hand, the fire localization step involves additional processing. Analyzing the feature maps from different CNN layers is necessary to identify fire-sensitive regions. While this process can provide accurate fire segmentation, it adds computational complexity to the method.

3.6 Compact CNNs and SLIC-Based Clustering for Fire Detection and Localization in Video Frames

Similarly, Thomson, Bhowmik and Breckon (2020) investigate the use of more compact versions of NasNet-A-Mobile (ZOPH et al., 2018) and ShuffleNetV2 (MA et al., 2018) for detecting and localizing fires in images extracted from video frames. The proposed CNNs are applied to find frames containing fires. From the frames classified as containing fires, an iterative clustering is performed using the SLIC algorithm. Similar to the work of Chino et al. (2015), the superpixels generated by SLIC are also classified by Thomson, Bhowmik and Breckon (2020) to identify fire regions, but they propose this classification through their CNN models. Good results were obtained on their image dataset with these simplified networks. However, the work only evaluated the networks' performance for image classification and did not produce segmentation masks.

One of the main advantages of the proposed method is the investigation of using compact versions of CNNs, such as NasNet-A-Mobile and ShuffleNetV2, to detect and localize fires in images from video frames. These more compact networks might be more efficient in terms of computational resource utilization, allowing for faster fire detection and localization.

Another positive aspect is the application of CNNs to find frames containing fires. This helps reduce processing in areas of the image that are not relevant to the specific problem, saving time and computational resources.

However, despite the promising nature of employing the SLIC algorithm for itera-

tive clustering of frames identified as fire-containing, there is potential for further analysis. Evaluating the effectiveness of this segmentation method across diverse datasets could provide valuable insights into its segmentation capabilities. A more thorough assessment of the segmentation accuracy achieved by SLIC, especially in comparison to alternative fire segmentation methods, would be crucial.

3.7 Fire Detection with DeepLabV3 Semantic Segmentation

Mlích et al. (2020) compiled a dataset with polygon annotations and examined the performance of the DeepLabV3 semantic segmentation architecture (CHEN et al., 2017) to address the presented problem. In creating the dataset, the authors focused on including a substantial number of challenging images from the negative class ("non-fire") to ensure the network's robustness across diverse situations. This method demonstrated impressive enhancements in the FPR metric compared to other state-of-the-art works.

On a positive note, the authors curated a comprehensive dataset with polygon annotations, enabling a precise and detailed evaluation of the DeepLabV3 semantic segmentation model's performance (CHEN et al., 2017). Furthermore, their emphasis on incorporating a substantial number of challenging negative class images into the dataset aims to bolster the model's resilience in various and demanding scenarios.

Conversely, it is crucial to consider certain limitations. While the model achieved remarkable improvements in the FPR metric, its performance needs assessment across other relevant metrics such as the TPR.

Despite the mentioned limitations, the proposed method represents a substantial advancement in the field of fire detection, leveraging a well-established semantic segmentation architecture and constructing a comprehensive and challenging dataset. With improved FPR and a robust approach, this method holds the potential to contribute to enhancing the safety and efficiency of fire detection systems. However, further research and comprehensive evaluations are required to assess its overall performance and applicability in real-world scenarios.

4 Proposed Method

We introduce the proposed architecture depicted in Figure 4.1, where boxes with a blue marker share the same architecture. This method encompasses a prior classifier, a patch classifier dedicated to locating fire regions within an image, and employs graph cuts and a color thresholding process to enhance the precision of the fire shape in the segmentation.

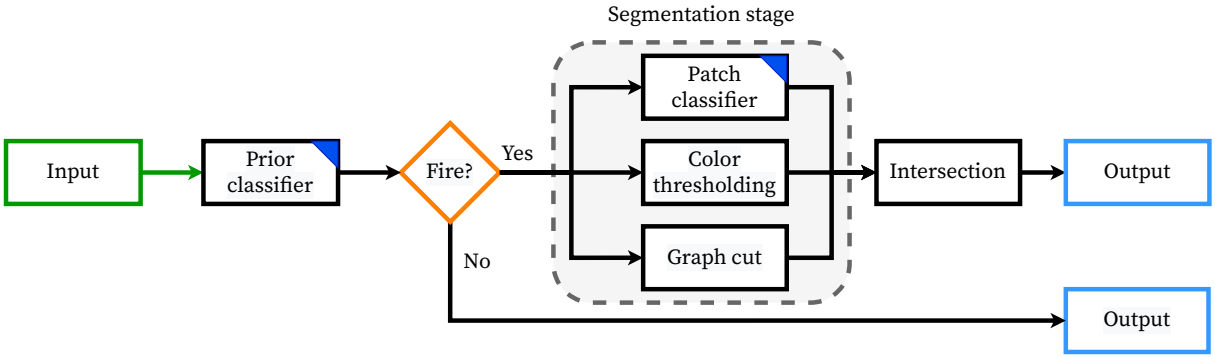


Figure 4.1: General architecture of the proposed model for fire detection and segmentation.

The prior classifier's role is to classify images as containing fire or not, thereby bypassing negative images in the segmentation stage. The performance of the prior classifier is crucial to the pipeline, as accurately identifying and excluding non-fire images significantly impacts overall results. If fire is detected in an image by this model, it proceeds to the segmentation stage. Here, a segmentation mask is produced by both the patch classifier and the graph cuts and color thresholding. The mask generated by the patch classifier is derived by classifying regions, or patches, of 50 pixels with a 25-pixel step increment, where the positive classification prevails in overlapping areas. Consequently, this mask provides a coarser segmentation. On the other hand, the color thresholding generates a more refined mask by classifying the color of each pixel in the image as fire or not based on a color range. Since fire colors are not limited to fire itself and are present in various other objects, the final step of the method narrows down the color-based mask to the regions identified as fire by the network through an intersection. The graph cuts technique further enhances the granularity of the masks by integrating both geometric

and color information, providing a more comprehensive segmentation.

The architectural framework for the networks aligns with our previous strategy (PEREIRA; VIEIRA; VILLELA, 2022), employing the CoAtNet-4 (DAI et al., 2021). Both the prior classifier and the patch classifier share the same CoAtNet-4 architecture but are trained on distinct datasets due to their specific purposes.

The patch classifier is primarily involved in region classification, producing an initial mask that may suffer from imprecision and a high FPR. To refine this mask and address these issues, it is combined with a mask generated through a process involving graph cuts and color thresholding. This process commences with the construction of a graph $\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$ for each image. In this graph, pixels are represented as nodes and are connected to their four neighboring nodes, along with a source node and a sink node. The source node symbolizes the background, while the sink node represents the foreground. The objective is to assign a label x_i to each node $i \in V$, indicating whether it belongs to the foreground ($x_i = 1$) or the background ($x_i = 0$). This binary labeling problem is tackled using graph cuts, minimizing energy functions that factor in pixel similarity with the terminal nodes (source and sink) and their neighborhood.

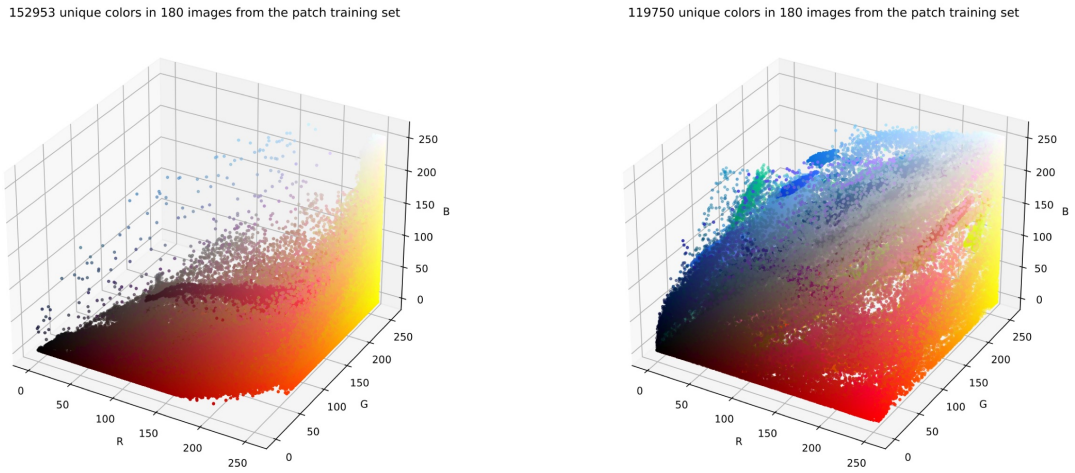
Many methods require manual selection of seed pixels for foreground and background in the graph cuts labeling step (LI et al., 2004; AGRAWALA et al., 2004), with these labels remaining fixed throughout the min-cut optimization. An illustration of such user input and the resulting outcome is depicted in Figure 4.2.

Figure 4.2: Example of graph cut result with user input. Source: Li et al. (2004).

However, our approach automates this process by determining which pixels belong

to fire regions and which belong to the background region. Importantly, these labels are not static during the min-cut optimization but instead serve as initial conditions (Sá et al., 2005).

This approach relies on a reference to calculate pixel similarity with the terminal nodes, making use of the color distribution from the training set to define the seeds. This is achieved by determining the distance to the nearest fire and non-fire colors for each pixel. Given the extensive number of colors in the dataset, as depicted in Figure 4.3, the use of every possible color would be impractical. Interestingly, the fire color distribution in this dataset features a larger number of unique colors, encompassing numerous distinct hues reminiscent of fire.



(a) Fire color distribution.

(b) Non-fire color distribution.

Figure 4.3: Color distribution in the patch training set in the RGB space.

To streamline this process, the k-means algorithm is employed to cluster the colors, reducing the search space to 1024 colors. Resembling the approach used by Li (LI et al., 2004), we define energy functions, denoted as E_1 and E_2. E_1 is defined as follows:

$$E_1(x_i = 1) = \frac{d_i^{\mathcal{F}}}{d_i^{\mathcal{F}} + d_i^{\mathcal{B}}} \qquad E_1(x_i = 0) = \frac{d_i^{\mathcal{B}}}{d_i^{\mathcal{F}} + d_i^{\mathcal{B}}} \quad \forall i \in \mathcal{V}, \tag{4.1}$$

where $d_i^{\mathcal{F}} = |C(i) - K^{\mathcal{F}}(i)|$, which is the distance between the color of pixel i and its closest color from the foreground clusters. Following the same logic for the background distance, we have $d_i^{\mathcal{B}} = |C(i) - K^{\mathcal{B}}(i)|$.

Similarly, E_2 is defined as follows:

$$E_2(x_i, x_j) = \lambda |x_i - x_j|, \tag{4.2}$$

where the color difference between pixels x_i and x_j is multiplied by a λ value to balance the energy sum.

These energies are computed for each node in V. The term E_1 determines the inherent inclination of each pixel to belong to either fire or background, while E_2 establishes the suitability of neighboring pixels to have the same label. To illustrate the graph construction, Figure 4.4 showcases a few connections between pixels of a fire image and the terminal nodes.

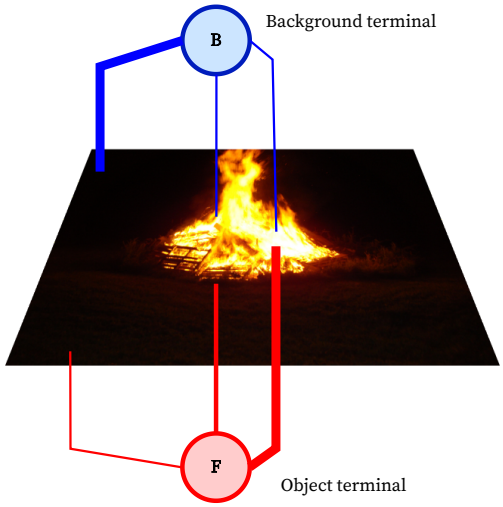


Figure 4.4: Illustration of initial graph connections between pixels of a fire image and the terminal nodes.

To minimize the energy functions, the Boykov-Kolmogorov algorithm (BOYKOV; KOLMOGOROV, 2004) is employed, a method that has demonstrated superior performance compared to previous methods in the context of CV problems, making it an ideal choice for this approach.

Finally, in constructing the color thresholding process to further enhance segmentation granularity, we leverage the color distribution of fire from the training set of patches. This approach facilitates the acquisition of a color range that optimally represents potential fire colors. The performance of color thresholding in fire segmentation also necessitates evaluation for potential refinements.

5 Experiments and Results

5.1 Datasets

As the proposed method involves a combination of neural networks and other CV techniques, working with different datasets is essential, as each network serves a specific purpose.

The creation of the initial classifier involves the compilation of a dataset from 5416 images originating from Li, Yan and Liu (2020), complemented by an additional 905 images sourced from Research (2017) and Saied (2020). This dataset maintains a distribution of 2605 images within the "fire" class and 3716 images within the "non-fire" class. Figure 5.1 shows a set of sample images used to train the prior classifier.

Figure 5.1: Sample images from the training set of the prior classifier.

In addition to neural network training, the dataset is also used for validation. Hence, this dataset was split into a training set and a validation set. The data division was performed in such a way that 85% of the images are used for training, and 15% are allocated for validation. The image distribution per class and dataset split is depicted in Figure 5.2.

The training of the patch classifier involved using datasets from two sources: the

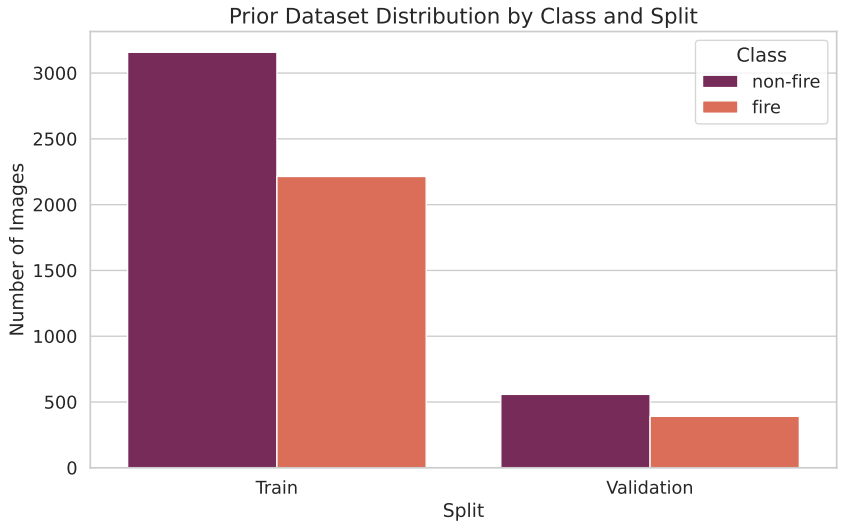


Figure 5.2: Prior classifier dataset distribution.

compilation by Chino et al. (2015) and the collection of images assembled by Cazzolato et al. (2017), which comprises emergency situation images for fire and smoke analysis. The dataset constructed by Chino et al. (2015) contains images of size 50×50 pixels (patches). By training the network on this dataset, the desired classifier for fire region identification can be obtained. Figure 5.3 shows images used to train the patch classifier, being 5.3a, 5.3b, 5.3e and 5.3f from the training set used by Chino et al. (2015), and 5.3c, 5.3d, 5.3g and 5.3h from Cazzolato et al. (2017).

Figure 5.3: Sample images from the training set of the patch classifier.

This dataset used for the patch classifier consists of 240 images, with 80 belonging to the “fire” class and 160 to the “non-fire” class. In order to expand and balance the dataset further, an additional 240 images were added from the dataset created by

Cazzolato et al. (2017). This results in a dataset of 240 labeled "fire" patches and 240 labeled "non-fire" patches. The image distribution per class and dataset split is depicted in Figure 5.4.

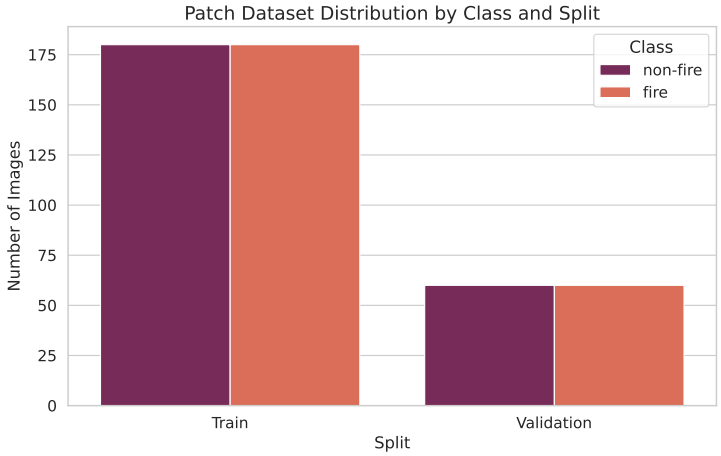


Figure 5.4: Patch classifier dataset distribution.

Ultimately, an additional dataset is employed to evaluate our methodology. This segmentation dataset, curated by Chino et al. (2015), encompasses a total of 226 images. Despite its size, the dataset presents a set of challenging images for a comprehensive assessment. Out of these, 119 images depict instances of fire, while the remaining 107 images do not feature any fire. The segmentation performance of our approach is benchmarked against other methodologies using this dataset. Figure 5.5 displays a selection of images from the test set assembled by Chino et al. (2015).

Figure 5.5: Sample images from the test set.

5.2 Experiment Setting

The complete architecture is implemented using the PyTorch framework (PASZKE et al., 2019) and is obtained from a publicly available repository[1]. In the experiments, the model is initialized with random weights. For the training of the patch classifier network, the Adaptive Moment Estimation (Adam) optimizer is employed, along with a cross-entropy loss function, a learning rate of 0.001, a batch size of 8, and training over 40 epochs. Similarly, the training of the prior classifier also uses the Adam optimizer and cross-entropy loss function, spanning 40 epochs. However, the prior classifier training incorporates a lower learning rate of 0.0001, a batch size of 12, and a learning rate decay by a factor of 0.1 at the 12th, 24th, and 36th epochs to facilitate learning.

In order to enhance the performance of the initial model and achieve a more balanced classification, adjustments have been made to the class threshold. Rather than classifying images with an output probability of 0.5 or greater as non-fire, the criterion is now set at 0.7. This modification ensures that an image is discarded only if the model is highly confident that it does not contain any fire. Consequently, even if some non-fire images proceed to the next step, the patch classifier can correct this error by not detecting fire regions within the image. This refinement significantly improves the overall accuracy and reliability of the classification process.

Similarly, the class threshold for the patch classifier undergoes modification to offer increased flexibility to the graph cuts in making the final determination regarding pixel classification as fire or non-fire. The non-fire threshold for the patch classifier is adjusted to 0.9, indicating that only predictions with a probability exceeding 90% are classified as non-fire. This adaptation empowers the graph cuts to exert their influence, given the high confidence of the patch classifier in identifying non-fire regions.

During the construction of graphs from images, instances may arise where the difference between $d_i^{\mathcal{F}}$ and $d_i^{\mathcal{B}}$ is relatively small, potentially causing confusion in the graph cut algorithm. To mitigate this, a criterion is introduced: if the difference in distance is less than a quarter of the maximum possible distance, the energy function $E_1(x_i = 0)$ is increased by a factor of 1.5. Through experimentation, it was determined that a value

[1] ⟨https://github.com/chinhsuanwu/coatnet-pytorch⟩

of 0.0005 for λ in Eq. (4.2) yields the best results. This adjustment contributes to a clearer distinction between fire and non-fire regions, thereby enhancing the accuracy of the segmentation process. The graph cut algorithm is implemented using the PyMaxflow library[2].

5.3 Evaluation Criteria

This section introduces the main objective comparison criteria to evaluate fire detection methods. The most common are recall (also known as TPR), precision, accuracy, F_1-score and F_β-score. With C being the confusion matrix of a binary problem, $C_{i,j}$ refers to the number of samples known to be in class i and predicted to be in class j, with $i, j = 1, 2$. Hence, the overall accuracy is defined as the number of correctly predicted pixels divided by the total number of pixels:

$$A = \frac{\sum_{i=1}^{2} C_{i,i}}{\sum_{i=1}^{2} \sum_{j=1}^{2} C_{i,j}}. \tag{5.1}$$

The F_1-score is the harmonic mean of the precision (P) and recall (R) and is defined as follows:

$$F_1\text{-score} = 2 \times \frac{P \times R}{P + R}, \quad P = \frac{TP}{TP + FP}, \quad R = \frac{TP}{TP + FN}, \tag{5.2}$$

where TP is the number of true positive pixels correctly predicted as positive (fire), FP is the number of true negative pixels predicted as positive and FN is is the number of true positive pixels predicted as negative (non-fire).

The F_β-score serves as a versatile metric, extending the F_1-score to allow control over the balance between precision and recall through the β coefficient. This parameter plays a crucial role in adjusting the emphasis on either precision or recall. Specifically, a $\beta < 1$ accentuates the importance of precision and proves beneficial for scenarios where minimizing false positive predictions is of particular interest. Conversely, a $\beta > 1$ dimin-

[2] ⟨https://github.com/pmneila/PyMaxflow⟩

ishes the significance of precision, prioritizing the reduction of false negative predictions.

$$F_\beta\text{-score} = (1 + \beta^2) \times \frac{P \times R}{\beta^2 \times P + R}. \tag{5.3}$$

5.4 Results

An extensive series of experiments was conducted to find the best parameters for the graph cuts. Table 5.1 shows a sample of the combinations tested and their respective results. In this table, the column labeled k represents the number of clusters employed for k-means clustering during the calculation of color distances for the graph cut energies, as elucidated in Chapter 4. Importantly, no substantial improvements were observed for higher values of k, and such increments would likely introduce unnecessary complexity without proportional benefits.

Table 5.1: Sample of parameter combinations tested and their respective results.

k	λ	TPR	FPR
5	0.1	0.90	0.21
15	0.05	0.93	0.21
25	0.03	0.96	0.24
50	0.01	0.96	0.23
128	0.01	0.96	0.21
256	0.003	0.96	0.19
512	0.001	0.96	0.18
1024	0.0001	**0.97**	**0.17**

To evaluate the efficacy of the method, its performance is assessed on the dataset compiled by Chino et al. (2015), juxtaposing the results with those achieved by alternative approaches. A comprehensive summary of the experimental outcomes for each step of the proposed method, as well as their combinations, is presented in Table 5.2. Notably, the color classifier and graph cuts demonstrate an exceptionally high TPR. However, they also exhibit a notable tendency to overlook a substantial number of true negative pixels. As indicated in the ablation study, the prior classifier effectively reduces the FPR at a small cost to TPR.

Table 5.3 displays the best results for TPR and FPR values obtained from the experiments, comparing them with the results reported by state-of-the-art works. Our

Table 5.2: Results of each step and their combinations.

Phase		TPR	FPR
Without prior classifier	Color classifier	**0.99**	0.59
	Graph cut	**0.99**	0.36
	Patch classifier	0.97	0.17
	Intersection	0.96	0.11
With prior classifier	Color classifier	0.91	0.20
	Graph cut	0.97	0.23
	Patch classifier	0.94	0.07
	Intersection	0.92	**0.03**

method achieves the highest TPR among all the approaches, preserving a competitive FPR. It is important to note that, in the case of the method proposed by Mlích et al. (2020), while it achieves the lowest FPR for the problem, it comes with the cost of a 0.87 TPR.

Table 5.3: Comparison of TPR and FPR reported by various approaches.

Method	TPR	FPR
Color Classification (CHINO et al., 2015)	0.77	0.13
BoWFire et al. (CHINO et al., 2015)	0.65	0.03
CNNFire T=0.40 (MUHAMMAD et al., 2019)	0.82	0.02
CNNFire T=0.45 (MUHAMMAD et al., 2019)	0.85	0.04
CNNFire T=0.50 (MUHAMMAD et al., 2019)	0.89	0.07
Mlích et al. (MLÍCH et al., 2020)	0.87	**0.01**
Previous method (PEREIRA; VIEIRA; VILLELA, 2022)	0.91	0.04
Novel method (PEREIRA; VIEIRA; VILLELA, 2023)	**0.92**	0.03

Table 5.4 reveals a significant improvement in precision and F_1-score compared to the previous proposal. As noted by Pereira, Vieira and Villela (2022), it is crucial to highlight the considerable class imbalance within the dataset, contributing to the challenges with these criteria. Despite a similar number of fire and non-fire images, the dataset contains around 21 times more non-fire pixels than fire pixels. In response to this, the F_2-score was computed, resulting in a value of 0.80. Given the critical importance of not missing fire cases in fire detection, where false alarms are less harmful than undetected fires, the method demonstrates superior fire classification (TPR) compared to the cited methods. Furthermore, the latest method achieves an accuracy of 0.96, surpassing the 0.95 accuracy reported by Pereira, Vieira and Villela (2022). Although some works, like

those by Rudz et al. (2013), Chen, Wu and Chiou (2004), Muhammad et al. (2019), and Mlích et al. (2020), achieve good precision and F_1-score values, they still fall behind in terms of TPR.

Table 5.4: Comparison of Precision and F_1-score reported by different works.

Method	Precision	F_1-score
Chino et al. (CHINO et al., 2015)	0.50	0.57
Rudz et al. (RUDZ et al., 2013)	0.63	0.52
Rossi et al. (ROSSI; AKHLOUFI; TISON, 2011)	0.39	0.28
Celik et al. (CELIK; DEMIREL, 2009)	0.55	0.54
Chen et al. (CHEN; WU; CHIOU, 2004)	0.75	0.25
Muhammad et al. (MUHAMMAD et al., 2019)	**0.86**	**0.91**
Mlích et al. (MLÍCH et al., 2020)	0.73	0.79
Previous method (PEREIRA; VIEIRA; VILLELA, 2022)	0.50	0.65
Novel method (PEREIRA; VIEIRA; VILLELA, 2023)	0.54	0.68

In Figure 5.6, the superiority our novel method is evident as it outperforms state-of-the-art results in terms of TPR, while maintaining a very low FPR comparable to the best-reported results in the literature. It is noticeable how more recent methods, harnessing the advent of deep learning, improved the state of the art. Despite consistently low FPR results, some methods struggle to achieve a balance between TPR and FPR.

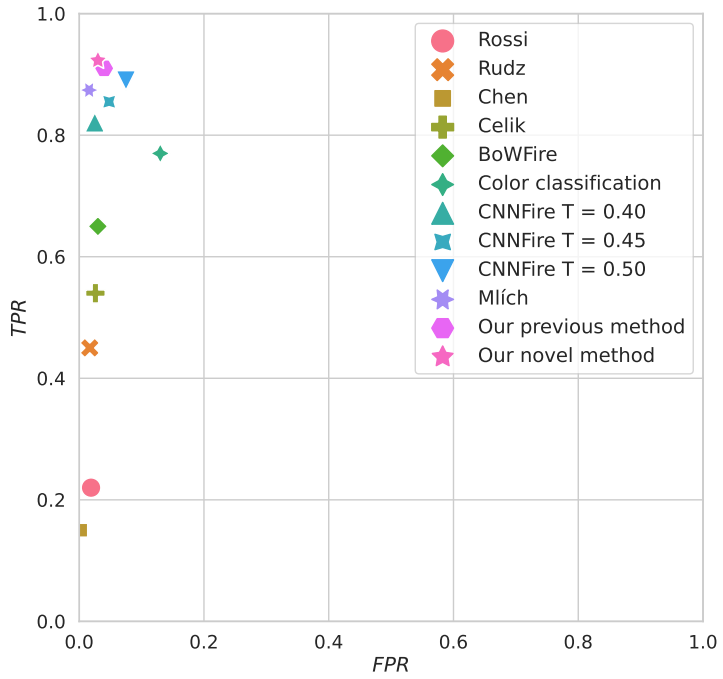
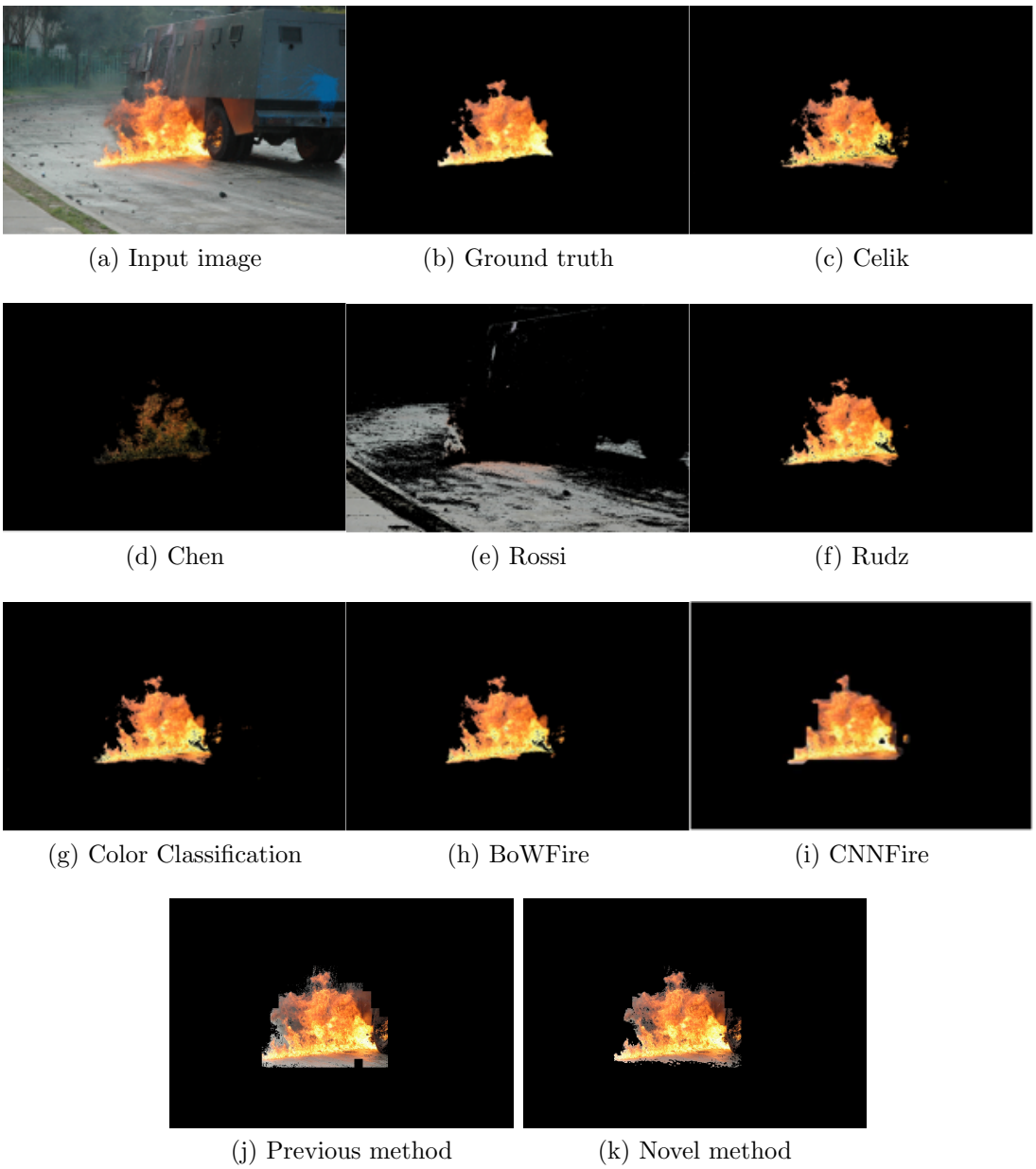


Figure 5.6: Comparison of results in the ROC space.

Figure 5.7 visually presents the results of the method applied to a fire image,

displaying the input image (5.7a), the corresponding ground truth (5.7b), and the resulting output achieved by several approaches. The masks obtained by Chen, Wu and Chiou (2004) and Rossi, Akhloufi and Tison (2011) are evidently unable to identify the fire in this particular sample. In contrast, while other approaches achieve fine-grained masks, our method successfully identifies the entire extent of the fire by prioritizing the positive class.



(a) Input image (b) Ground truth (c) Celik

(d) Chen (e) Rossi (f) Rudz

(g) Color Classification (h) BoWFire (i) CNNFire

(j) Previous method (k) Novel method

Figure 5.7: Result of various approaches for a fire image.

Figure 5.8 visually highlights the enhancements achieved through the integration of graph cuts into our methodology. Upon combining graph cuts with the other steps, a notable reduction in the FPR is observed. This reduction is attributed to the focused

predictions on regions identified as actual fire by the patch classifier, showcasing the efficacy of our refined approach.

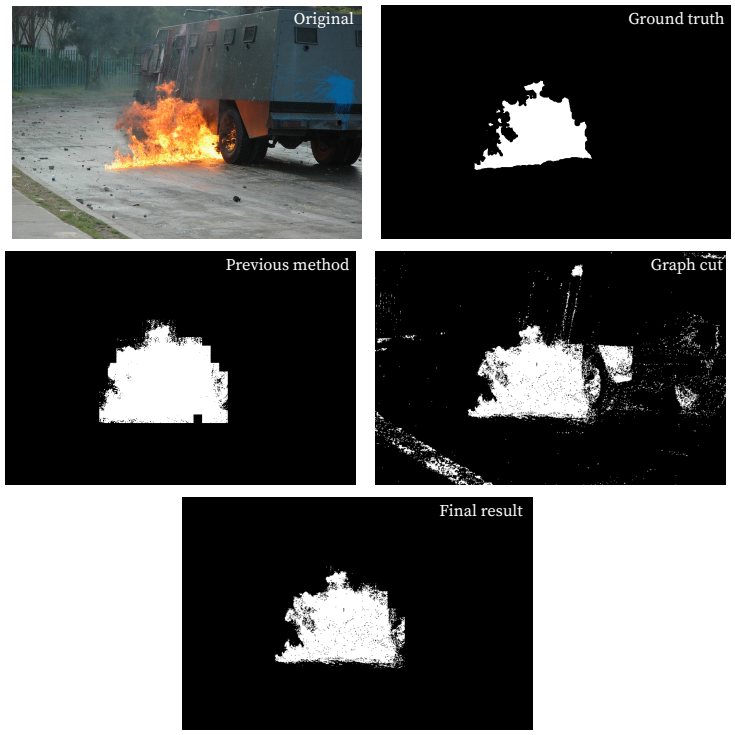


Figure 5.8: Visual demonstration of improvements over our previous method.

6 Conclusion

In today's world, characterized by the exponential growth of data and the increasing demand for time efficiency and accuracy, the role of AI is becoming increasingly crucial. AI algorithms have the potential to automate complex processes, enabling faster and more precise analysis of large volumes of data.

In the context of fire detection, the ability to segment fire regions in images is of great significance. Fire segmentation not only allows for the identification and localization of fire incidents but also provides a comprehensive understanding of the affected area, aiding in the planning of effective responses and mitigation strategies. A promising approach to addressing the fire detection problem is the combination of neural networks with graph cuts. By harnessing the power of DL and extracting meaningful features from input data, neural networks can learn to distinguish fire patterns from normal background elements.

The integration of graph cuts and color thresholding further enhances the accuracy of fire detection, capturing distinct colors associated with fire. This combined approach shows significant potential in detecting fires with greater efficiency and reliability, contributing to data-driven decision-making to prevent future incidents and minimizing the devastating consequences of fires.

In summary, this approach excels in advancing TPR while simultaneously reducing the FPR, a pivotal achievement given the criticality of accurately identifying fire instances. The application of graph cuts and the color thresholding algorithm following the patch classification step ensures that the resultant segmentation accurately outlines fire-related regions within an image. This synergistic combination of methods allows for a fine-grained understanding of the spatial extent and boundaries of fire instances, thereby contributing to a more precise analysis of fire occurrences.

While our approach excels in critical aspects, there is still room for improvement in refining the generated masks to adequately capture fire shapes. The inherent difficulty of the problem is illustrated in Figure 4.3, as the majority of fire colors are also present

in non-fire images, posing a significant challenge. Consequently, future research efforts will concentrate on conducting extensive experiments with the proposed combination of methods and refining associated parameters.

Achieving a remarkable TPR of 0.96 and a low FPR of 0.04 in the final segmentation results is possible with a perfect prior classifier. Therefore, enhancements to the models should be considered, experimenting with transfer learning methods and pre-trained models. Moreover, strategies such as augmenting the training dataset with synthetic data and applying data augmentation techniques are viable options to enhance the robustness and effectiveness of the models.

Bibliography

AGGARWAL, C. *Neural Networks and Deep Learning: A Textbook*. Springer International Publishing, 2018. ISBN 9783319944647.

AGRAWALA, A., DONTCHEVA, M., AGRAWALA, M., DRUCKER, S., COLBURN, A., CURLESS, B., SALESIN, D., COHEN, M. Interactive digital photomontage. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2004)*, ACM, 2004.

AHUJA, R. K., MAGNANTI, T. L., ORLIN, J. B. *Network Flows: Theory, Algorithms, and Applications*. 1. ed.. Prentice Hall, 1993. Hardcover. ISBN 9780136175490.

BOYKOV, Y., JOLLY, M.-P. Interactive graph cuts for optimal boundary & region segmentation of objects in n-d images. In: *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, 2001. v. 1, p. 105–112 vol.1.

BOYKOV, Y., KOLMOGOROV, V. An experimental comparison of min-cut/max- flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 26, n. 9, p. 1124–1137, 2004.

CAZZOLATO, M. T., AVALHAIS, L. P. S., CHINO, D. Y. T., RAMOS, J. S., SOUZA, J. A. d., JUNIOR, J. F. R., TRAINA, A. J. M. Fismo: A compilation of datasets from emergency situations for fire and smoke analysis. In: *Brazilian Symposium on Databases*. SBC, 2017.

CELIK, T., DEMIREL, H. Fire detection in video sequences using a generic color model. *Fire Safety Journal*, Elsevier, v. 44, n. 2, p. 147–158, 2009.

CHEN, L.-C., PAPANDREOU, G., SCHROFF, F., ADAM, H. *Rethinking Atrous Convolution for Semantic Image Segmentation*. 2017.

CHEN, T.-H., WU, P.-H., CHIOU, Y.-C. An early fire-detection method based on image processing. In: IEEE. *2004 International Conference on Image Processing, 2004. ICIP'04.*, 2004. v. 3, p. 1707–1710.

CHINO, D. Y., AVALHAIS, L. P., RODRIGUES, J. F., TRAINA, A. J. Bowfire: detection of fire in still images by integrating pixel color and texture analysis. In: IEEE. *2015 28th SIBGRAPI Conference on Graphics, Patterns and Images*, 2015. p. 95–102.

CORDTS, M., OMRAN, M., RAMOS, S., REHFELD, T., ENZWEILER, M., BENENSON, R., FRANKE, U., ROTH, S., SCHIELE, B. The cityscapes dataset for semantic urban scene understanding. In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

DAI, Z., LIU, H., LE, Q. V., TAN, M. *CoAtNet: Marrying Convolution and Attention for All Data Sizes*. arXiv, 2021.

DOSOVITSKIY, A., BEYER, L., KOLESNIKOV, A., WEISSENBORN, D., ZHAI, X., UNTERTHINER, T., DEHGHANI, M., MINDERER, M., HEIGOLD, G., GELLY, S., USZKOREIT, J., HOULSBY, N. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *CoRR*, abs/2010.11929, 2020.

FESTAG, S. False alarm ratio of fire detection and fire alarm systems in Germany – A meta analysis. *Fire Safety Journal*, v. 79, p. 119–126, 2016. ISSN 0379-7112.

GOODFELLOW, I., BENGIO, Y., COURVILLE, A. *Deep learning*. MIT press, 2016.

HE, K., ZHANG, X., REN, S., SUN, J. *Deep Residual Learning for Image Recognition*. 2015.

IANDOLA, F. N., HAN, S., MOSKEWICZ, M. W., ASHRAF, K., DALLY, W. J., KEUTZER, K. *SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and ¡0.5MB model size*. 2016.

KIM, Y.-H., KIM, A., JEONG, H.-Y. RGB Color Model Based the Fire Detection Algorithm in Video Sequences on Wireless Sensor Network. *International Journal of Distributed Sensor Networks*, v. 10, n. 4, p. 923609, 2014.

KRIZHEVSKY, A., SUTSKEVER, I., HINTON, G. E. Imagenet classification with deep convolutional neural networks. In: PEREIRA, F., BURGES, C., BOTTOU, L., WEINBERGER, K. (Ed.). *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2012. v. 25.

LECUN, Y., BENGIO, Y., HINTON, G. Deep learning. *nature*, Nature Publishing Group, v. 521, n. 7553, p. 436, 2015.

LECUN, Y., BOTTOU, L., BENGIO, Y., HAFFNER, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, v. 86, n. 11, p. 2278–2324, 1998.

LI, S., YAN, Q., LIU, P. An efficient fire detection method based on multiscale feature extraction, implicit deep supervision and channel attention mechanism. *IEEE Transactions on Image Processing*, v. 29, p. 8467–8475, 2020.

LI, Y., SUN, J., TANG, C.-K., SHUM, H.-Y. Lazy snapping. *ACM Trans. Graph.*, Association for Computing Machinery, New York, NY, USA, v. 23, n. 3, p. 303–308, 2004.

MA, N., ZHANG, X., ZHENG, H.-T., SUN, J. *ShuffleNet V2: Practical Guidelines for Efficient CNN Architecture Design*. 2018.

MLÍCH, J., KOPLÍK, K., HRADIŠ, M., ZEMČÍK, P. Fire segmentation in still images. In: BLANC-TALON, J., DELMAS, P., PHILIPS, W., POPESCU, D., SCHEUNDERS, P. (Ed.). *Advanced Concepts for Intelligent Vision Systems*. Cham: Springer International Publishing, 2020. p. 27–37.

MUHAMMAD, K., AHMAD, J., LV, Z., BELLAVISTA, P., YANG, P., BAIK, S. W. Efficient deep cnn-based fire detection and localization in video surveillance applications. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, v. 49, n. 7, p. 1419–1434, 2019.

NIELSEN, M. *Neural Networks and Deep Learning*. Determination Press, 2015.

OPENCV. *K-Means Clustering in OpenCV*. n.d. ⟨https://docs.opencv.org/3.4/d1/d5c/tutorial_py_kmeans_opencv.html⟩. Accessed December 1, 2023.

PASZKE, A., GROSS, S., MASSA, F., LERER, A., BRADBURY, J., CHANAN, G., KILLEEN, T., LIN, Z., GIMELSHEIN, N., ANTIGA, L., DESMAISON, A., KOPF, A., YANG, E., DEVITO, Z., RAISON, M., TEJANI, A., CHILAMKURTHY, S., STEINER, B., FANG, L., BAI, J., CHINTALA, S. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In: WALLACH, H., LAROCHELLE, H., BEYGELZIMER, A., ALCHé-BUC, F. d', FOX, E., GARNETT, R. (Ed.). *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2019. v. 32.

PEREIRA, D. M., VIEIRA, M. B., VILLELA, S. M. Combining neural networks and a color classifier for fire detection. In: *Intelligent Systems: 11th Brazilian Conference, BRACIS 2022*. Campinas: Springer, 2022. p. 139–153.

PEREIRA, D. M., VIEIRA, M. B., VILLELA, S. M. Graph cuts and deep neural networks for fire detection. In: *2023 36th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*. Rio Grande: IEEE, 2023.

RELIEFWEB. *2021 Disasters in numbers*. 2021. ⟨https://reliefweb.int/report/world/2021-disasters-numbers⟩. Accessed December 1, 2023.

RESEARCH, C. for A. I. *Fire-Detection-Image-Dataset*. 2017. ⟨https://github.com/cair/Fire-Detection-Image-Dataset⟩. Accessed December 1, 2023.

ROSSI, L., AKHLOUFI, M., TISON, Y. On the use of stereovision to develop a novel instrumentation system to extract geometric fire fronts characteristics. *Fire Safety Journal*, Elsevier, v. 46, n. 1-2, p. 9–20, 2011.

RUDZ, S., CHETEHOUNA, K., HAFIANE, A., LAURENT, H., SÉRO-GUILLAUME, O. Investigation of a novel image segmentation method dedicated to forest fire applications. *Measurement Science and Technology*, IOP Publishing, v. 24, n. 7, p. 075403, 2013.

RUMELHART, D. E., HINTON, G. E., WILLIAMS, R. J. Learning representations by back-propagating errors. *nature*, Nature Publishing Group UK London, v. 323, n. 6088, p. 533–536, 1986.

SAIED, A. *FIRE Dataset*. 2020. ⟨https://www.kaggle.com/phylake1337/fire-dataset⟩. Accessed December 1, 2023.

SANDLER, M., HOWARD, A. G., ZHU, M., ZHMOGINOV, A., CHEN, L. Inverted residuals and linear bottlenecks: Mobile networks for classification, detection and segmentation. *CoRR*, abs/1801.04381, 2018. ⟨http://arxiv.org/abs/1801.04381⟩.

SRIVASTAVA, N., HINTON, G., KRIZHEVSKY, A., SUTSKEVER, I., SALAKHUTDINOV, R. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, v. 15, n. 56, p. 1929–1958, 2014. ⟨http://jmlr.org/papers/v15/srivastava14a.html⟩.

SUN, C., SHRIVASTAVA, A., SINGH, S., GUPTA, A. Revisiting Unreasonable Effectiveness of Data in Deep Learning Era. *CoRR*, abs/1707.02968, 2017.

SZELISKI, R. *Computer vision algorithms and applications*. London; New York: Springer, 2011.

Sá, A. M. E., VIEIRA, M. B., CARVALHO, P. C. P., VELHO, L. Range-enhanced active foreground extraction. In: *International Conference on Image Processing*. Genova, Italy: IEEE, 2005. p. 81–84.

THOMSON, W., BHOWMIK, N., BRECKON, T. P. Efficient and Compact Convolutional Neural Network Architectures for Non-temporal Real-time Fire Detection. *CoRR*, abs/2010.08833, 2020.

U.S. FIRE ADMINISTRATION. *U.S. Fire Statistics*. 2022. ⟨https://www.usfa.fema.gov/statistics/⟩. Accessed December 1, 2023.

VASWANI, A., SHAZEER, N., PARMAR, N., USZKOREIT, J., JONES, L., GOMEZ, A. N., KAISER, L., POLOSUKHIN, I. Attention is all you need. *CoRR*, abs/1706.03762, 2017. ⟨http://arxiv.org/abs/1706.03762⟩.

ZOPH, B., VASUDEVAN, V., SHLENS, J., LE, Q. V. *Learning Transferable Architectures for Scalable Image Recognition*. 2018.