People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research
Chadli Bendjedid University
Faculty of Science and Technology
IT department

الجمهورية الجزائرية الديمقراطية الشعبية
وزارة التعليم العالي والبحث العلمي
جامعة الشاذلي بن جديد
كلية العلوم و التكنولوجيا
قسم الاعلام الالي

Master Thesis in Computer Science

Presented by

KARROUCHE Haithem

Field: Computer Science

Specialty: Intelligent Computer Systems

Theme

Forest Fires Prevention Employing Artificial Intelligence

Sustained on: 25/06/2023 **In front of the Jury composed of:**

Mme. BOUGARNE Imene	**MCB**	**Univ of El Tarf**	**President**
Mme. AHMED MALEK Nada	**MAA**	**Univ of El Tarf**	**Supervisor**
Mr. CHEMAM Chaouki	**MAA**	**Univ of El Tarf**	**Examiner**

University Year: 2022/2023

Acknowledgment

First and foremost, we begin by expressing our profound gratitude to Allah the Almighty for His unwavering support and guidance throughout our journey. It is through His blessings and the courage He bestowed upon us that we were able to overcome challenges and accomplish our studies successfully.

We would like to extend our heartfelt appreciation and warm greetings to our esteemed supervisor Mme. AHMED MALEK Nada. Your wisdom, invaluable advice, and the time dedicated to us have been pivotal in shaping our scientific growth. Your constant guidance and mentorship have inspired us to strive for excellence in our work.

Our sincere thanks go to the members of the jury who graciously accepted the responsibility of reviewing and evaluating our work. Their expertise and insightful feedback have undoubtedly contributed to the refinement of our research.

We would also like to express our gratitude to all the teachers of the computer science department. Your dedication and commitment to our education through master's course, have been instrumental in shaping our knowledge and skills in the field. Your passion for teaching has left an indelible mark on our academic journey.

Furthermore, we would like to extend our thanks to our parents, family, friends and all those who have directly or indirectly contributed to the realization of these cherished memories. Your unwavering support, encouragement, and belief in our abilities have been a constant source of motivation.

Lastly, we acknowledge the success of this beautiful academic year, which would not have been possible without the collective efforts of all those mentioned above. We are truly grateful for the guidance, support and remarkable contributions of each and every individual who has played a part in our journey.

May Allah bless you all abundantly.

With deepest appreciation.

Dedications

To my beloved parents,

It is with profound joy, an open heart and an overwhelming sense of pleasure that I dedicate this work to you. Throughout my life, you have been my unwavering pillars of support, demonstrating immense respect and magnificence in your roles as my parents. Your merits and sacrifices for which words fall short, have left me in awe. I struggle to express the depth of gratitude, pride, and profound love I hold for you, as you have selflessly dedicated yourselves to my success. May this dedication serve as a testament to my unwavering attachment, gratitude, and respect, and may Allah bestow upon you both good health and a long life.

To my cherished grandmother, Fatima,

You hold a special place in my heart, and it is impossible to quantify the depth of my love for you and the significance you hold in my life.

To my dear brothers and sisters,

I sincerely hope that this work meets and exceeds your expectations. May it serve as a heartfelt expression of the profound affection I hold for each of you. I extend my deepest gratitude for the moral support and encouragement you have provided me throughout this journey. I wholeheartedly wish you all the happiness and success you deserve, and may your future be filled with brilliance.

To my friends, those who love me and those whom I hold dear,

Your friendship has been a constant source of inspiration, encouragement, and strength. I am deeply grateful for your unwavering support and for always being there when I needed you most.

To all those who have shown me kindness, offered assistance, and provided support throughout my academic journey,

With sincere gratitude, I dedicate this work to you. Your contributions have made an indelible impact on my success, and I am forever grateful for your presence in my life.

Karrouche Haithem.

Abstract

Forest fires are a global concern with severe implications for ecosystems and communities. This thesis addresses the pressing issue of forest fire prevention using artificial intelligence, specifically deep learning techniques. The research explores various aspects related to forest fires, including their patterns, triggers, effects, and existing prevention approaches. The challenges encountered during the research process, such as acquiring a comprehensive dataset and designing a lightweight yet accurate model, are discussed. The proposed system utilizes supervised machine learning, with a focus on the Light-Fire Model (LF-Model), to detect and classify forest fires. Results demonstrate the effectiveness of the system in accurately predicting fire occurrences in real-time. The limitations and trade-offs of the models are also discussed, highlighting the need for optimization and deployment considerations. By continuously refining and improving the system, this thesis contributes to the preservation of natural ecosystems and the mitigation of the devastating impact of forest fires.

Keywords: *Classification, Deep Learning, Forest Fires, Light-Fire Model, Prevention*

Résumé

Les incendies de forêt sont un problème mondial qui a de graves répercussions sur les écosystèmes et les communautés. Cette thèse aborde la question urgente de la prévention des incendies de forêt en utilisant l'intelligence artificielle, en particulier les techniques d'apprentissage profond. La recherche explore divers aspects liés aux incendies de forêt, y compris leurs modèles, leurs déclencheurs, leurs effets et les approches de prévention existantes. Les défis rencontrés au cours du processus de recherche, tels que l'acquisition d'un ensemble complet de données et la conception d'un modèle léger mais précis, sont discutés. Le système proposé utilise l'apprentissage automatique supervisé, en mettant l'accent sur le modèle d'incendie léger (modèle LF), pour détecter et classer les incendies de forêt. Les résultats démontrent l'efficacité du système pour classifier avec précision les incendies en temps réel. Les limites et les compromis des modèles sont également discutés, soulignant le besoin d'optimisation et les considérations de déploiement. En affinant et en améliorant continuellement le système, cette thèse contribue à la préservation des écosystèmes naturels et à l'atténuation de l'impact dévastateur des incendies de forêt.

Mots clés : *Classification, Apprentissage Profond, Incendies de Forêt, Modèle d'incendie léger, Prévention.*

ملخص

تعتبر حرائق الغابات مصدر قلق عالمي ولها آثار وخيمة على النظم البيئية والمجتمعات. تتناول هذه الأطروحة القضية الملحة المتعلقة بالوقاية من حرائق الغابات باستخدام الذكاء الاصطناعي، وخاصة تقنيات التعلم العميق. يستكشف البحث الجوانب المختلفة المتعلقة بحرائق الغابات، بما في ذلك أنماطها ومحفزاتها وآثارها وأساليب الوقاية القائمة. تمت مناقشة التحديات التي تمت مواجهتها أثناء عملية البحث، مثل الحصول على مجموعة بيانات شاملة وتصميم نموذج خفيف الوزن ولكنه دقيق. يستخدم النظام المقترح التعلم الآلي الخاضع للإشراف، للكشف عن حرائق الغابات وتصنيفها. تظهر النتائج فعالية النظام في التنبؤ الدقيق بحوادث الحريق في الوقت الفعلي. تتم أيضًا مناقشة القيود والمقايضات الخاصة بالنماذج، مما يبرز الحاجة إلى اعتبارات التحسين والتطبيق. من خلال صقل النظام وتحسينه باستمرار، تساهم هذه الأطروحة في الحفاظ على النظم البيئية الطبيعية والتخفيف من الآثار المدمرة لحرائق الغابات.

الكلمات الرئيسية: التصنيف، التعلم العميق، حرائق الغابات، نموذج الحرائق الخفيفة، الوقاية.

Table of Contents

List of Figures

List of Tables

Acronyms List

- **FF:** Forest Fires
- **FWI:** Fire Weather Index
- **FFMC:** Fine Fuel Moisture Code
- **ISI:** Initial Spread Index
- **DMC:** Duff Moisture Code
- **DC:** Drought Code
- **BUI:** Build Up Index
- **CNN**: Convolutional Neural Network
- **DT**: Decision Tree
- **IOT**: Internet of Thing
- **WSN**: Wireless Sensor Networks
- **SVM**: Support Vector Machine
- **ViT**: Vision Transformers
- **GPU**: Graphics Processing Units
- **TPU**: Tensor Processing Units
- **SGD**: Stochastic Gradient Descent
- **Adam**: Adaptive Moment Estimation
- **RMSprop**: Root Mean Square Propagation
- **Nadam**: Nesterov Adam

Natural calamities such as earthquakes, tornadoes, floods and forest fires not only degrade the land but also endanger human lives, either directly or indirectly. One of the most significant hazards confronting our society is forest fires, particularly in areas of dense population or where properties and homes are located close to woodland areas. Wildfires are a major threat to our environment and economy and their destructive potential is compounded by the fact that they can be difficult to contain once they start due to the sheer size of many wildfires and the unpredictable nature of their propagation, fueled by strong winds, dry conditions, and dead vegetation. In some parts of the world, such as Algeria, these risks are exacerbated by climate change, leading to increased temperatures, longer periods of drought and subsequent higher levels of tinder-dry vegetation that can act as fuel for fires. As a result, it is essential to be aware of the risks posed by forest fires and understand how they can be prevented and mitigated in order to protect our environment, our communities, and the economic interests that rely on a healthy landscape.

According to global statistics research, forest fires have become an increasingly severe concern all over the world, with climate change escalating the problem and causing extensive damage to forests and other ecosystems. These implications have attracted the attention of governments, organizations and non-government entities, which are developing innovative measures to address the situation.

In order to effectively combat this problem, comprehensive strategies are needed to identify areas prone to forest fires, develop early warning systems and create effective firefighting services that can quickly respond to the danger.

One of the most important factors that contribute to reducing the dangerous repercussions of fire is early wildfire recognition. For these reasons, researchers have been working to come up with effective approaches and policies that aid in the prediction of fires and the treatment of their spread in order to limit the damage and reduce the danger they pose to our societies. Due to their practical significance in the swift completion of many jobs and the saving of many human efforts, artificial intelligence and machine learning are among the ways that scientists use to anticipate forest fires.

In this thesis, we suggest a system for detecting forest fires that is based on supervised machine learning especially deep learning.

By utilizing neural networks with multiple layers, deep learning models can extract complex features from visual data, leading to remarkable accuracy in various domains. In the context of forest fire detection, the application of deep learning holds immense potential for enabling rapid and reliable fire identification.

One of the key challenges in implementing deep learning models for forest fire detection lies in balancing accuracy with computational efficiency. While many existing models achieve high accuracy, they often come with significant computational and memory requirements, limiting their practicality for real-time deployment on resource-constrained platforms. To overcome this challenge, our study focuses on developing a lightweight deep learning model that prioritizes efficiency without compromising accuracy. This lightweight model can serve as a foundation for integrating fire detection capabilities into drones, enabling enhanced surveillance and response mechanisms.

To achieve these objectives, the study will be structured as follows:

1. We start with the first chapter which is called "the state of the art" it's focused on overview concerning forest fires and more detailed description on used techniques to solve this matter also an explanation of various deep learning methods besides CNN application which is the chosen method of our project.
2. The second chapter is called "conceptual study", consist of the design of our system. This section will detail the proposed approach from dataset to applied augmentations. Thus, the light-weight model architecture.
3. Chapter three and the last one, "Implementation and obtained results", this section will present a comparison and discussion of the results found also the implementation of our system as well as the used programming tools.

The brief ends as usual with a conclusion. We also indicate some possible perspectives for our research activities.

1. Introduction

For decades, forest fires have been a naturally occurring element of many ecosystems, playing an important role in altering the landscape and restoring plants. However, as human activities have affected the ecology and increased the frequency and severity of fires. Hence, the implications of wildfires have shifted over time.

The objective of this chapter is firstly, to make a descriptive analysis of the fires, patterns and behaviour as well as its complications especially in landscape, secondly, we will make a qualitative analysis of the current works done by the scientific community for preventing forest fires in order to suggest ways of improvement. Finally, the chapter focuses on the presentation of Convolutional Neural Networks (CNNs) and their relevance to forest fire detection.

2. Overview of Forest Fires

Historically, Fire has been employed as a tool for hunting, agriculture, and land management by ancient civilizations, surface flames is used to clear land for farming, create hunting grounds, and promote the growth of useful vegetation. Any such ancient use of fire was well-suited to the local environment, with fires normally being small and simply managed.

However, as human populations have risen and extended into new areas, the impact of fires has escalated as a result of human-caused events such as harvesting, mining, and construction activities, which have transformed the forest habitat and made it more susceptible to fires. Furthermore, climate change has resulted in drier circumstances and an increase in the frequency of droughts, making fires more likely to start and spread. (1)

Forest fires now have a major effect on the world, people's life and the economic system. Bushfire smoke and dust particles often lead to respiratory issues and other health concerns, while the fires itself can damage residences and other facilities and harm people. Wildfires can also cause biodiversity destruction for species, soil erosion and water quality deterioration. The economic impact is visible in the loss of resources such as lumber, as well as the expenditures of confronting and recovering from fires. (1)

2.1 Wild Fires in Algeria

Forest fires have become a growing worry in Algeria in past years. The country has a Mediterranean climate, which means that summertime is sweltering and dry, while winters are mild. (2)

As this climate as well as human activities such as logging, agriculture and urban expansion, forests are more vulnerable to flames. Moreover, the country has seen a considerable increase in the frequency and intensity of droughts in recent years, increasing the risk of fires. (3)

Climate change is also thought to be a significant influence in the increased frequency and intensity of forest fires in Algeria. The country has seen an increase in temperature and a decrease in rainfall, resulting in drier conditions and greater evaporation. These modifications have made it simpler for flames to start and spread, as well as more difficult to suppress fires once they have begun. Climate change is also likely to exacerbate the situation in the future by increasing the frequency and intensity of droughts and heat waves. (3)

Bushfires have had a significant influence on Algeria's environment, economy, and human life, destroying homes and other structures and resulting in the loss of lives.

It has been reported that over 100 fires occurred across north-eastern Algeria on the night of Sunday, August 14, 2022, effecting 14 governorates (Wilayas): Bejaia, Jijel, Souk Ahras, El Taref, Setif, Skikda, tipaza, tizi-ouzou, Guelma, Batna, Mila, Annaba, Constantine, and Bordj BouArreridj (4). It has been reported that it damaged a large part of Al-Kala, which is among the most important national parks of Algeria. Furthermore, at least 44 individuals were killed, 250 were injured, and more than 500 households were displaced (2,000 people), the flames destroyed almost 6,000 hectares according to estimates provided to the Algeria Red Crescent (4). The expenditures of fighting and recovering from flames demonstrate the economic impact.

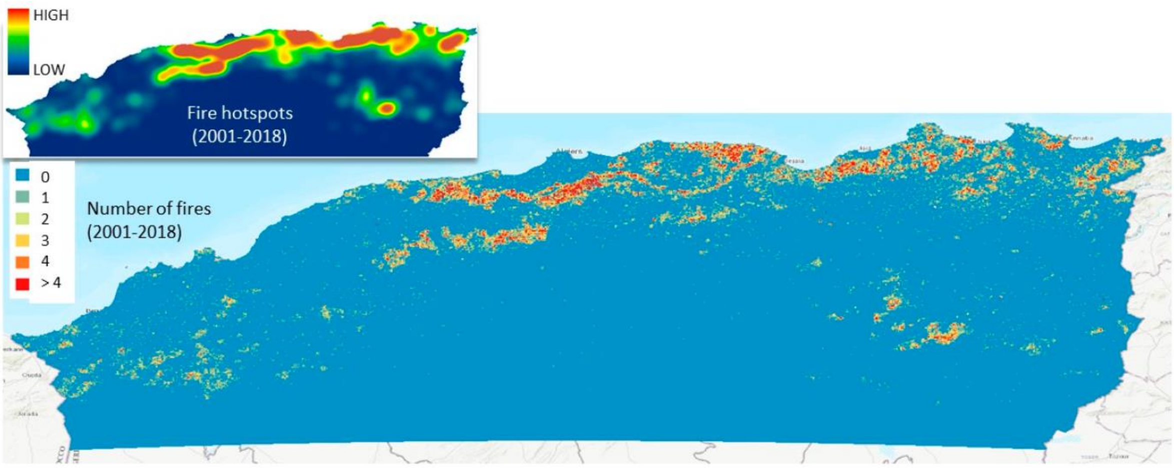

Figure 1: Maps of fire occurrence in Northern Algeria (number of fires, 2001–2019). The enclosed Figure (upper left) indicates the fire hotspots and their level of density. (5)

To address the problem of forest fires in Algeria, the government has initiated a variety of fire preventative measures, such as increasing patrols, enhancing firefighting equipment, mobilizing more than 1,200 firefighters (National Civil Protection) and seven fire helicopters (National Algerian Army) in response to this disaster (4), and raising public knowledge about the hazards of fires. However, as the frequency and severity of fires escalates, new and more effective solutions are required in order to prevent and manage these fires.

2.2Fire Triangle

Fire, as a broad term, refers to a common physical event caused by the rapid combustion of oxygen with a material and identified by heat, light and flame. Hence, the three elements that must be present for fire to start fuel, heat and oxygen are understood using the fire triangle model, as presented in figure 2.

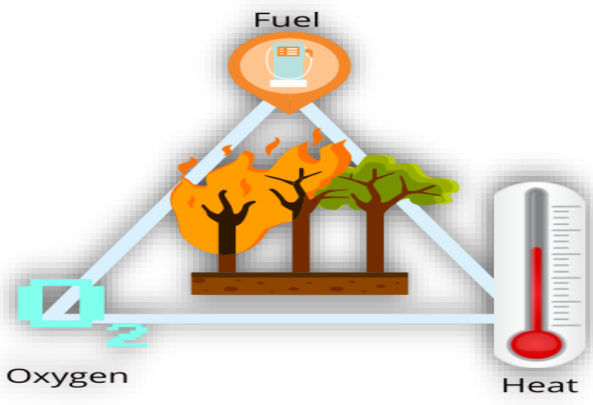

Figure 2 : Fire Triangle.

A removal of even one of these parts will eradicate the fire. The fire triangle is an essential tool for both fire prevention and suppression because it serves as a reminder of how important it is to get rid of any one of these elements in order to properly put out a fire.

2.3The Fire Behavior Triangle

A similar triangle known as the fire behavior triangle exists beside the fire triangle. Fuels, weather, and topography are the three apex points of this triangle. The parts that follow go into considerable detail about each of these and how they affect fire. (6)

2.3.1 Fuels

The flammability of a fuel is determined by its composition which includes the amount of moisture, chemical makeup and density.

- ***Moisture Level:*** The most crucial factor is the amount of moisture. The moisture content of dead logs is typically low compared to that of live trees. How quickly and how hot a fire can spread depends on the moisture content and distribution of these fuels. Because heat from the fire must first remove moisture, high moisture content will impede the burning process. (6)

- ***Chemical Makeup:*** How easily it will burn is what determines. Some plants, shrubs and trees contain oils or resins that help them burn more readily, quickly or intensely than other plants do. (6)
- ***Density:*** The quantity of fuel affects its flammability. Fuel will burn faster if the particles are near together because they will ignite each other. However, if the fuel particles are so close that air cannot pass easily or at a suitable distance, the fuel will not burn freely.

As a result of the impact that fire has both above and below the surface, soil types must also be taken into account. How much a fire will influence soil depends on its moisture content, its organic matter content, and how long it burns.

2.3.2 Weather

Weather factors also have an impact on fire behavior such as:

- ***Wind:*** One of the most essential aspects because it can provide a new supply of oxygen to the fire and drive it toward a new fuel source.
- ***Temperature of fuels:*** is determined by the air temperature since fuels obtain heat by absorbing solar radiation from the environment. A fuel's propensity to igniting is affected by its temperature. Fuels will often ignite more easily at high temperatures than at low temperatures. (6)
- ***Humidity:*** The amount of water vapor in the air influences the moisture content of a fuel. Fuels grow dry at low humidity levels allowing them to catch fire and burn more quickly than at high humidity levels.

2.3.3 Topography

A fire's behavior can be influenced by landform. It will normally move faster uphill than downhill or over level ground.

Land shape is described via topography. It can include elevation descriptions such as the height above sea level 'slope', the steepness of the land 'aspect' and the direction a slope faces 'features' such as canyons, valleys, and rivers.

These topographical features can either help or hinder fire spread. A rocky slope, for example, can serve as an excellent natural fire barrier due to a lack of fuel and a large gap of open space. Drainages can also serve as fire breaks if the fuels are damp or there is little vegetation. Aside from the land's shape, height, slope, and aspect must all be considered. Elevation and aspect can influence how hot and dry an area could be. (6)

The figure 3 provides a visual representation of the impact of topographic factors on vegetation and snow cover patterns.

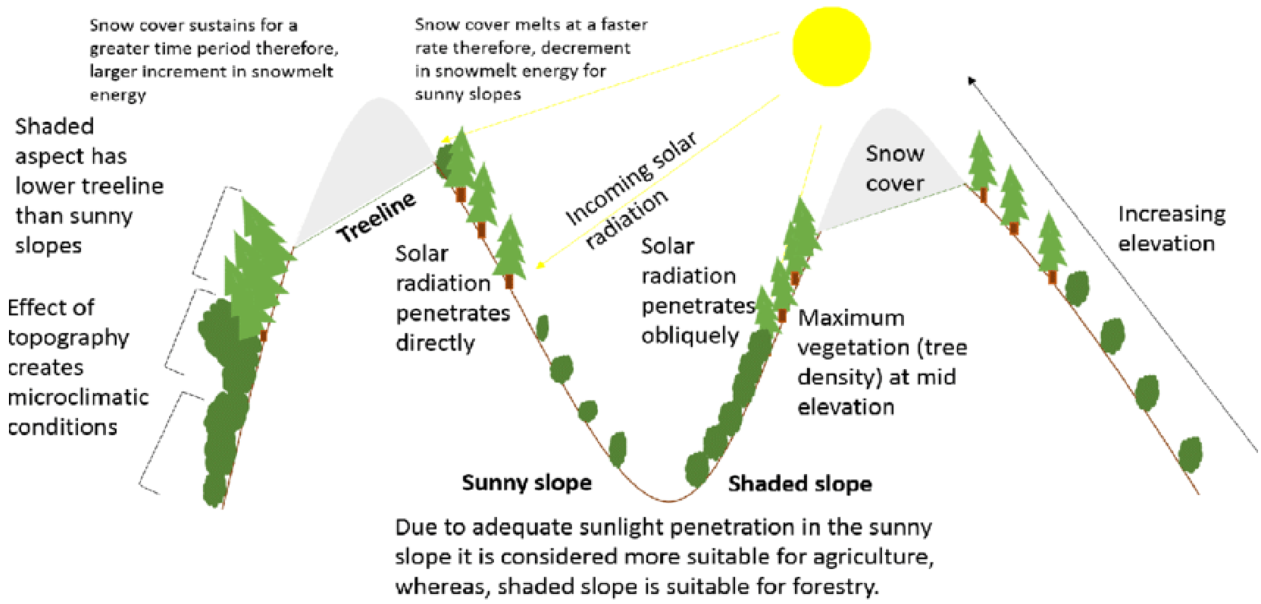

Figure 3 : Effect of topography (slope aspect, elevation and topographic position) on vegetation and snow cover. (7)

3.Forest Fires and their Patterns

3.1 Forest Fires

The term "forest fire" sometimes known as a " blaze" refers to any unrestrained and uncontrolled combustion or burning of plants in a natural setting or due to human activity, such as a forest or grassland, that burns organic fuels and spreads due to environmental variables (such as wind and topography).

3.2Forest Fire Patterns

Depending on the circumstances of the fire, one or more types of fires may occur at the same time. The sorts of forest fires are as follows:

- ***Ground Fire:*** As shown in next figure, is a type of wildfire that begins underground when fuels such as garbage ignite and burn. Depending on the surroundings of the fire, ground fires may continue to burn through the ground layer and become surface flames. These fires spread more slowly than surface fires and can last months.

Figure 4: Ground Fire.

- ***Surface Fire:*** See Figure 5, a surface fire can range in intensity from low to moderate level depending on the conditions. Small plants such as sticks or dried leaves are the primary fuel sources. Ground fires that have grown large enough to overtake the surface are to blame for those type of flames. They may blister a tree cover but they will not cause it to burn sufficiently to start a fire. Surface fires normally spread slowly, but they can spread quickly if they start on steeply inclined terrain or are pushed by the wind. On the other hand, most surface fires burn out before progressing to the next classification level: crown fires.

Figure 5: Surface Fire.

- ***Crown Fire:*** As illustrates in figure 6, crown flames are a distinct type of wildfire. These deadly fires grow from one treetop to the next, a phenomenon known as the "tree crown" or "canopy." As a result of their height and wind exposure, high-level fires often spread much faster than lower-level fires and burn quickly. As a natural outcome, they regularly evolve into extremely lethal flames.

Figure 6: Crown Fire.

4. Triggers of Forest Fires

Wild fires occur commonly due to two reasons: natural reasons and man-made reasons.

4.1 Man-made Causes

- ***Arson:*** This is the intentional setting of fires by individuals with malicious intent. Arson can be motivated by revenge, vandalism, or other criminal motives. (8)
- ***Campfires left unattended:*** A campfire that is not properly put out can ignite surrounding vegetation, leading to a wildfire.
- ***Discarding lit cigarettes:*** Throwing a lit cigarette out of a car window or carelessly discarding a lit cigarette can start a wildfire if the ground is dry and conditions are windy.
- ***Using equipment without spark arresters:*** Operating equipment such as chainsaws, tractors, and vehicles without spark arresters can create sparks that can ignite dry vegetation.

- ***Illegal outdoor burning:*** Outdoor burning of household waste, leaves, and other materials can quickly get out of control and start a wildfire. (8)
- ***Children playing with fire:*** Children who are playing with fire can start a wildfire if they are not supervised or do not understand the dangers associated with fire.

4.2 Natural Causes

- ***Lightning:*** Lightning strikes can ignite dry vegetation and start a wildfire, especially during dry and windy weather conditions.
- ***Dry and windy weather conditions:*** When the weather is dry and windy, fires can spread quickly and become difficult to control. (8)
- ***Volcanic eruptions:*** The intense heat generated during a volcanic eruption can ignite nearby vegetation and start a wildfire.
- ***Drought:*** Prolonged drought conditions can lead to a buildup of dry vegetation, making it easier for fires to start and spread.
- ***High temperatures:*** High temperatures can cause dry vegetation to ignite more easily, increasing the likelihood of a wildfire.

5. Effects of Forest Fires

The consequences of forest fires are numerous and varied. Some of these might be quick and obvious while others might be delayed and challenging to notice. Fire alters the forest in a number of ways through its physical, biological and chemical effects. A few types of fire effects include:

5.1 Effects on Trees

Trees can be significantly impacted by forest fires. Among the most frequent effects are:

- High-intensity flames burn established trees and seedlings, leaving behind dead trees that can become a threat.
- The extreme heat of a fire and crown scorch harm tree needles and branches, decreasing their capacity to re grow and store energy.
- The fire's intense heat might harm the bark of trees which exposes them to disease and insect infestations.
- Combustion can cause damage to tree roots, limiting their ability to absorb water and nutrients.

- The fire can change the chemical makeup of the soil, making re-establishment of trees more difficult. (8)

It is essential to mention that the effects of fire on trees might vary depending on the severity of the fire, the species and the surrounding environmental circumstances. Furthermore, some tree species may be more fire-resistant than others and recover more quickly after a fire.

5.2 Effects on Soil

- Physical changes to the soil structure making it less permeable and less able to retain water. (9)
- Deplete the soil's organic content leaving it more prone to erosion and runoff. (9)
- Flames cause flash floods by increasing the amount of water runoff of soil, Hence, lowering the amount of water that is available for growth. (9)
- Combustion can consume organic matter and release nutrients into the atmosphere reducing the fertility of the soil. (10)
- The pH and nutritional levels in the soil are altered due to fires making it more difficult for certain plants to re-establish. (10)
- Fires decrease the number of microorganisms and other soil biota which play an important role in nutrient cycling and soil health. (10)

5.3 Fire Effects on Animals (Wildlife)

- Destroys habitat and limit the availability of food, water and shelter for animals.
- wildfire might directly kill animals, particularly those who are unable to flee or reach safe zone in time
- Birds and other animals that depend on certain habitats for breeding or foraging may experience disruptions in their migration patterns as a result of fire.
- alter the distribution and abundance of predators and prey, which can have cascading effects on food webs and ecosystems

6. Fire Weather Index

The Fire Weather Index (FWI) is a metric that forecasts the likelihood and potential severity of wild-land fires. It is a composite indicator that assesses current and prospective fire danger by taking into account numerous meteorological elements such as temperature, relative humidity, wind speed and direction, and precipitation. While a high fire weather index score implies an increased danger of wildfire, it does not guarantee that a fire will occur. Other variables, such as human activity or lightning strikes, can also contribute to the initiation and spread of a fire. (11)

Fire management agencies utilize the FWI to prioritize fire suppression resources and make choices concerning fire restrictions, controlled burns, and evacuations. The FWI is utilized all around the world and has proven to be a useful tool for predicting and mitigating wildfires.

6.1Fire Weather Index Systems

There are several unique fire weather index systems used around the world to assess the risk of wildfires and to make decisions about fire management, taking into account different meteorological variables and using different calculation methods to predict the fire danger.

The most widely used fire weather index systems are:

- ***Canadian Fire Weather Index (FWI) System:*** This system provides a numerical rating of the fire danger based on atmospheric and fuel moisture conditions. It includes components such as the Fine Fuel Moisture Code (FFMC), Initial Spread Index (ISI), Duff Moisture Code (DMC), Drought Code (DC), Build Up Index (BUI), and Fire Weather Index (FWI). (12)
- ***Keetch-Byram Drought Index (KBDI):*** This system measures the moisture content of the soil and vegetation and is used primarily in the United States. (13)
- ***McArthur Forest Fire Danger Index (FFDI):*** This system is widely used in Australia and takes into account temperature, relative humidity, rainfall, and wind speed, among other factors. (14)
- ***Energy Release Component (ERC):*** used in the United States, provides a single number rating of the fire potential based on the available fuel and weather conditions. (15)
- ***Nesterov ignition Index:*** This system applied in Russia and other parts of Europe, is based on temperature, relative humidity, wind speed, and precipitation, and provides a numerical rating of the fire danger. (16)

These are some of the most widely used fire weather index systems, but there are many others in use around the world, each with its own strengths and limitations. It is important to use the appropriate system for the specific conditions in each region, and to always take into account other factors such as human activity and lightning strikes in making decisions about fire management.

6.2Fire Danger Levels

As for danger levels, The Canadian FWI System (12) as an example uses a numerical rating to assess the fire danger in a specific area. The FWI is calculated based on six components, as shown in next figure:

- ***FFMC*** (Fine Fuel Moisture Code)***:*** A measure of the moisture content of the forest litter and other fine fuels. It is calculated based on temperature, relative humidity, and recent precipitation.
- ***ISI*** (Initial Spread Index)***:*** A measure of the rate of fire spread, based on the wind speed and relative humidity.
- ***DMC*** (Duff Moisture Code)***:*** A measure of the moisture content of the forest floor, including the duff layer and larger fuels such as logs and branches.
- ***DC*** (Drought Code)***:*** A measure of the dryness of the forest floor and other fuels, based on the length of time since the last rainfall.
- ***BUI*** (Build Up Index)***:*** A measure of the total fuel available for combustion, based on the accumulated dead fuel and live fuel moisture.
- ***FWI*** (Fire Weather Index)***:*** An overall measure of the fire danger, based on the FFMC, ISI, DMC, DC, and BUI values.

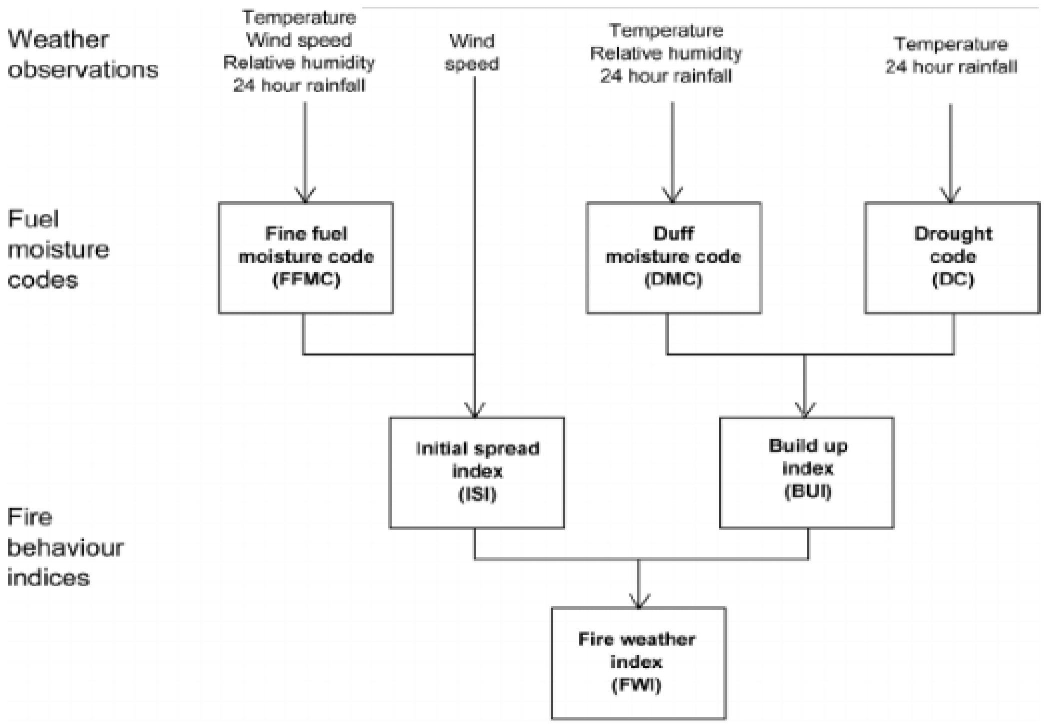

Figure 7: Fire-weather-index calculation scheme. (17)

The calculation of these components is based on a complex algorithm that considers the interplay of various weather and fuel moisture factors. The FWI provides a fire danger rating that ranges from low to extreme, which can be used to make decisions about fire management and resource allocation. The following are the levels of fire danger associated with the FWI system, as summarised in table 1:

- ***Low:*** A FWI of 0 to 8 indicates low fire danger. The fire potential is limited by the moisture content of the fuel and the weather conditions. (17)

- ***Moderate:*** A FWI of 8 to 16 indicates moderate fire danger. The fire potential is increasing, but fires are still unlikely to start and spread easily. (17)
- ***High:*** A FWI of 17 to 32 indicates high fire danger. Fires are more likely to start and spread rapidly, and the response time to control a fire is reduced. (17)
- ***Extreme:*** A FWI of 32 or higher indicates extreme fire danger. Fires start easily, spread rapidly, and can be difficult or impossible to control. (17)

FWI class	Value Range	Type of Fire	Potential Danger
Low	< 8	surface fire	Fire well be self-extinguishing
Moderate	8-16	surface fire	Easily suppressed with hand tools
High	17-32	surface fire	Power pumps and hoses are required
Extreme	>32	Developing crown fire	Immediate and strong action is critical

Table 1: Fire danger classification in the FWI system (17)

The FWI is just one aspect of a comprehensive fire management program and should be used in conjunction with other sources of information, such as satellite imagery, weather forecasts, and ground observations.

The following section will provide an overview of the current state of research on the context of this topic to gain a clearer understanding.

7. Related works

Over the recent years, researchers have focused on developing cutting-edge technologies and algorithms for forest fire detection (18). These studies are constantly contributing to the improvement of this field. Furthermore, with the advent of machine learning techniques, there has been a growing interest in using diverse strategies for appropriate counter-measures, after a review of various studies concerning the issue, it was concluded that the research could be broken down into three major areas (see figure 8):

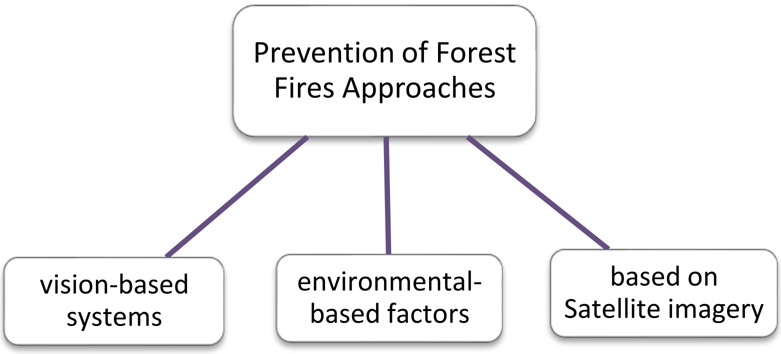

Figure 8: Principal categories of FF Prevention.

7.1 Fire detection vision-based systems

This subsection concentrates on vision-based systems using deep learning techniques an emerging field of machine learning that utilizes artificial neural networks to analyze the use of visual data to identify fires in real time and notify the appropriate authorities so that preventive measures can be taken. The visual sensor may be placed strategically throughout the forest as surveillance videos (19) (20)or used in conjunction with unmanned aerial vehicles (UAVs) (21), while deep learning models are able to classify the images and detect any features or patterns associated with fire.

Method	Dataset	Description	Performance	year
(19)	Foggia's video dataset (22), Chino's dataset (23)	The authors employed a pre-trained Alex-Net CNN model in their analysis. They fine-tuned it to detect fire in various indoor and outdoor conditions based on their problem domain, their model composed of five convolution layers, three pooling and fully connected layers, and receives color images of size 224 x 224 x 3 as input. The last layer outputs probabilities for the two classes (Fire, No Fire). All the experiments were	The performance of their method was investigated against other tests using a normal test image that had fire-like portions, and with small portions of real fire placed on different regions of the normal image. The method was able to detect fire at early stages and differentiate between fire and normal images, even with different rotations. The results suggest that this method can be useful for disaster management	2018

		performed using a multiple dataset of 68457 images divided onto 20% data for training and 80% for testing.	systems, attaining 94.39% accuracy.	
(20)	The authors created a dataset consisting of synthetic fire images with fog and real fire foggy images	The same authors suggested an efficient CNN-based approach for fire detection in an unclear IOT scenario on the collected video frame, also to save computing complexity and storage they used lightweight deep neural networks without applying dense fully connected layers.	The experimental results revealed that their model outperformed the state-of-the-art approach at the time with 95.86% accuracy.	2019

(21)	Dataset for Forest Fire Detection from Mendeley (24).	The proposed model for fire detection has an input size of 250x250 with three RGB color channels. The model includes standard convolutional operations, Relu activation, and Max Pooling to down-sample the feature maps. Depth-wise separable convolution operation is performed twice to reduce the feature map size. Then, two consecutive separable convolutional operations are conducted with 64 filters. Batch normalization is used to normalize the weight of the model, and Max Pooling is performed to get 64 feature maps sized 4x4. After flattening, three fully connected layers are used for classification, with the first two FC layers having a size of 128 and a 30% dropout, and the last FC layer's size is 64 with a 20% dropout. Two dense layers with a soft-max activation function are added to the proposed network to maintain the number of output classes, which are fire and non-fire.	The model achieved high accuracy, precision, recall, and F1-Score, with a test accuracy of 96.4% and a highest test accuracy of 97.6% at the 120th epoch. The precision, recall, and F1-Score curves are similar to the accuracy curves, indicating the unbiased model and reliability. The model's performance was compared with concurrent methods in the field (19) (20).	2022

Table 2:Overview of Existing Methods (1).

In this table, various studies related to forest fire detection vision-based systems and classification are presented, along with the datasets used and their corresponding performance metrics.

7.2Predictive modeling based on environmental factors

This subsection focuses on applying machine learning algorithms to forecast the probability of a fire utilizing environmental indicators such as humidity, wind speed, and drought degree captured using wireless sensor networks (WSN) which is a device that collects and converts physical parameters into a signal that such an observer or machine can interpret. Sensors come in a variety of shapes and sizes, and they're employed in a wide range of tasks. The detection of forest fires is one of the most popular uses for sensors. Hence, the algorithms examine previous data and estimate the likelihood of upcoming fires.

Numerous fire detection methods and algorithms have been proposed in the recent literature. In this section, we review some works related to wildfire detection, as illustrates in next table.

Method	Dataset	Description	Performance	year
(25)	The authors collected dataset that regroups a data of two regions: the SidiBel-abbes region located in the northwest and the Bejaia region located in norths of Algeria with 244 instances, 122 instances each region, dedicated for target class of two possibilities, namely ‘fire’and ‘not fire’.	Faroudja A et al. (25) investigated various ensembles of different decision tree algorithms for fire prediction and found that the simple decision tree performed the best in terms of recall and accuracy. The boosting decision tree had the best overall performance but was deemed too resource-intensive for hardware implementation. The simple decision tree was chosen as it offered	Comparison of the performance between the proposed model to existing methods using recall and F-measure as metrics. The results show that the proposed model performs better than some existing models in terms of recall with value of 0.92 and has a comparable F-measure (0.85) to other models. However, the comparison is only	2020

		a good tradeoff between performance and resource utilization on hardware.	indicative due to the difference in datasets used in each study.	
(26)	UCI repository Forest Fires Data Set (27), and collected Images.	The authors investigated supplementary areas of forest fire detection such as the deployment of WSNs and the use of machine learning models to predict forest fires using IoT sensors. Eight machine learning algorithms were tested (Boosted Decision Trees, Decision Forest Classifier, Decision Jungle Classifier, Local Deep Support Vector Machine (SVM), Logistic Regression and Binary Neural Network model). The paper proposes a novel IoT based smart fire prediction system that incorporates both meteorological data and images for early fire prediction.	The Boosted Decision Trees model was found to be the most suitable with an AUC value of 0.78, for an objective of classification the accuracy obtained with value 72%.	2020

Table 3: Overview of Existing Methods (2).

7.3Satellite imagery

Most existing forest fire detection technologies based on satellite imagery, the first Moderate-resolution Imaging Spectroradiometer (MODIS) (28)instrument launched in December

1999, MODIS offers daily products that address global forest fires. Every one to two days MODIS scans the entire surface of the planet. Additionally, its scanners collect data at three different spatial resolutions: 2 bands at 250 meters, 5 bands at 500 meters, and 29 bands at 1,000 meters. It is a long way from being able to detect forest fires in real time. In (29) (30), the imagery data from MODIS and AVHRR in China and Canada, respectively, are utilized to assess the risk of wildfires and identify them. Authors of (31) explore the possible requirements for space-based observations in fire control, covering fuel mapping, risk assessment, detection and burned area recovery. As an outcome of the long scanning duration and low resolution of satellites, weather conditions will significantly reduce the precision of satellite-based forest fire detection systems.

8. Forest Fires Prevention based on Deep Learning

Deep Learning is a rapidly evolving field that has gained a lot of attention in recent years due to its remarkable success in solving complex problems. It is a subset of Machine Learning that is based on artificial neural networks with multiple layers also known as deep neural networks. These networks are designed to mimic the structure and function of the human brain and they have the ability to learn from large amounts of data in an unsupervised or supervised manner.

The use of deep neural networks in Deep Learning has led to significant improvements in various applications, including computer vision, natural language processing, and speech recognition. Deep Learning models have also shown great promise in the field of forest fire prevention (32) (33). By analyzing large amounts of data from various sources, such as satellite images, weather patterns, and topography, deep learning algorithms can be trained to detect early signs of forest fires and predict their spread.

8.1Deep feedforward networks

Deep feedforward networks, also known as feedforward neural networks or multilayer perceptron (MLPs) are a type of artificial neural network (ANN) that consists of multiple layers of interconnected nodes or neurons. In a feedforward network, information flows in one direction from the input layer to the output layer with no feedback loops. The structure of a feedforward network is defined by the number of layers, the number of neurons in each layer and the connections between the neurons. A typical feedforward network consists of an input layer, one or more hidden layers and an output layer. The input layer receives the raw input data, which is then processed by the hidden layers, and finally produces the output (as shown in figure 9).

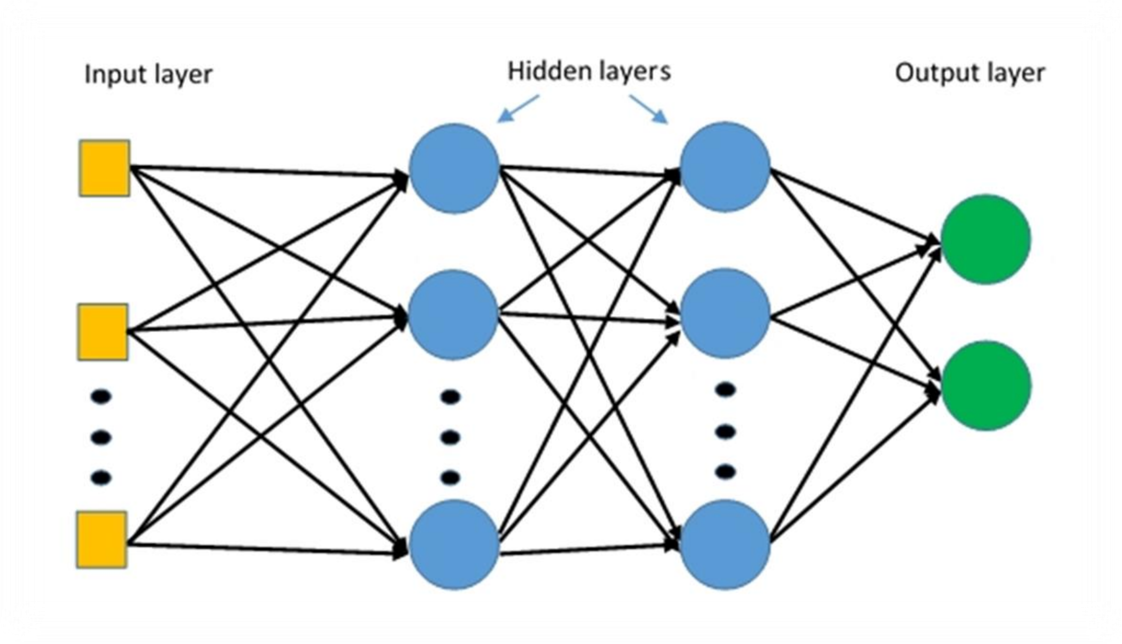

Figure 9: The Structure of Multi-layer Perceptron.

The basic building block of a feedforward network is a neuron, which receives input from other neurons or external sources, performs a computation on this input and produces an output. The output of one neuron is then passed as input to the next neuron in the network, until the final output is produced by the output layer.

Figure 10 depicts an artificial neuron's structure, represented as a circle with multiple input connections and a single output. Inputs are shown as arrows with associated weights, indicating their contribution to the neuron's output. Inputs are combined through summation, considering their respective weights. The resulting sum passes through an activation function, introducing non-linearity to model complex relationships. The output is then produced and serves as input for subsequent neurons or as the final output of the neural network.

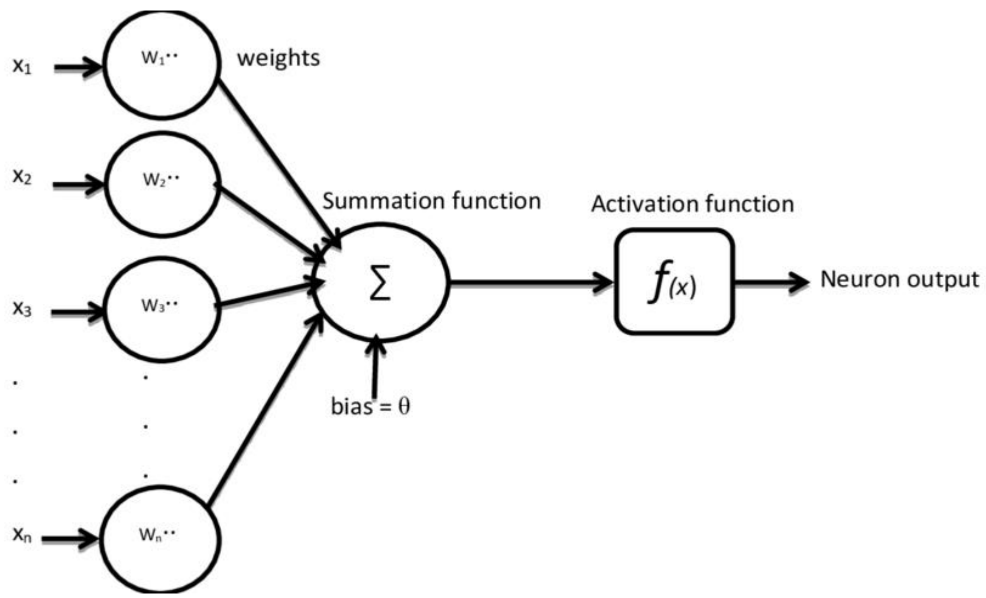

Figure 10: The structure of the artificial neuron. (34)

Each neuron in a feedforward network is typically characterized by an activation function, which maps the input to the output. Common activation functions include the sigmoid function, the rectified linear unit (ReLU) function, and the hyperbolic tangent (tanh) function. (See figure 11)

$$\boldsymbol{ReLu(x) = \max(0, x)} \quad \textbf{(1)} \qquad \textbf{sigmoid}(\mathrm{x}) = \frac{1}{(1+\mathrm{e}^{-\mathrm{x}})} \quad \textbf{(2)} \qquad \boldsymbol{tanh\, x = \frac{e^x - e^{-x}}{e^x + e^{-x}}} \quad \textbf{(3)}$$

Figure 11: Activation Functions ReLU (1), sigmoid and tanh (3).

8.2Convolutional Neural Network

Convolutional Neural Networks (CNNs) are a type of deep learning neural network that are specifically designed for processing grid-like data, such as images or videos. CNNs are widely used in computer vision tasks such as image classification, object detection, and image segmentation.

The key feature of CNNs is the convolutional layer, which performs a convolution operation on the input data. A convolution operation involves sliding a small window, called a filter or kernel, over the input data and computing a dot product between the filter and the input data at each position. The result of the convolution operation is a feature map, which highlights the presence of certain features in the input data.

The figure 12 below shows an example of a 2D convolution operation on a 5x5 input image using a 3x3 filter. The filter slides over the image one pixel at a time, computing the dot product between the filter weights and the corresponding pixels in the image.

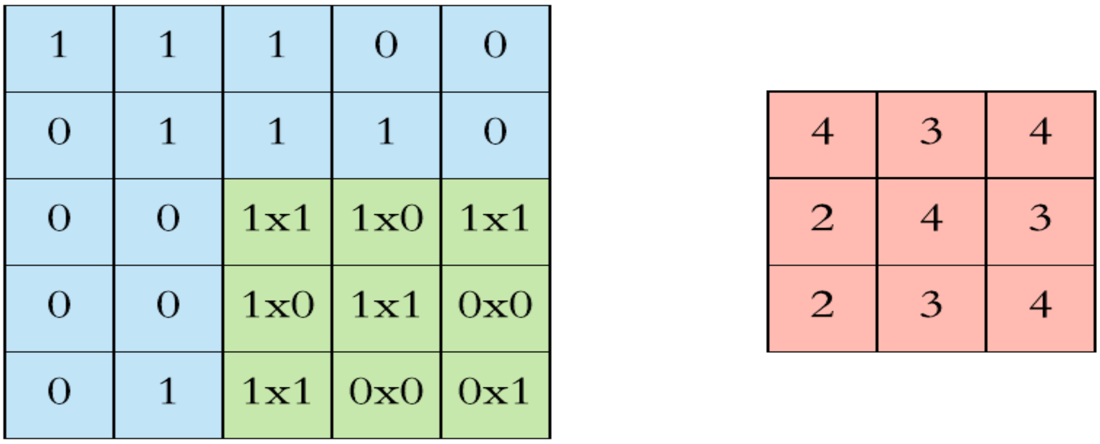

Figure 12: Convolution operation. (35)

In addition to the convolutional layer, CNNs also typically include other types of layers such as pooling layers, activation layers and fully connected layers. Pooling layers down sample the feature maps produced by the convolutional layer by computing a summary statistic such as the maximum or average value within a small window. This reduces the dimensionality of the feature maps and helps to make the network more robust to small variations in the input.

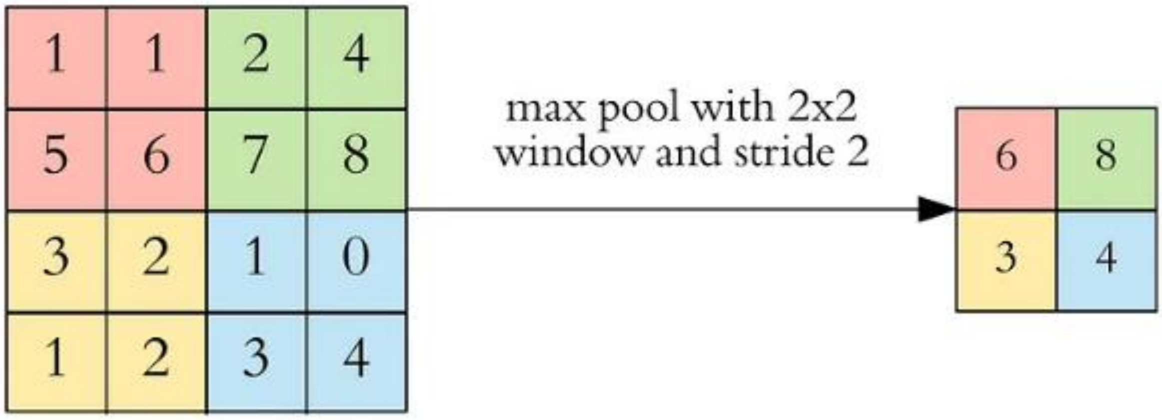

Figure 13: Pooling operation. (35)

Activation layers apply a non-linear function such as the ReLU or sigmoid function (as referred in figure 11) to the output of the previous layer. This introduces non-linearity into the network and allows it to learn more complex representations of the input data.

Fully connected layers, also known as dense layers, connect every neuron in one layer to every neuron in the next layer. These layers are typically used in the output layer of the network to produce the final classification or regression result.

The architecture of a CNN typically consists of a series of convolutional and pooling layers, followed by one or more fully connected layers. The figure below shows an example of a simple CNN architecture with four convolutional layers, four pooling layers followed by two fully connected layers.

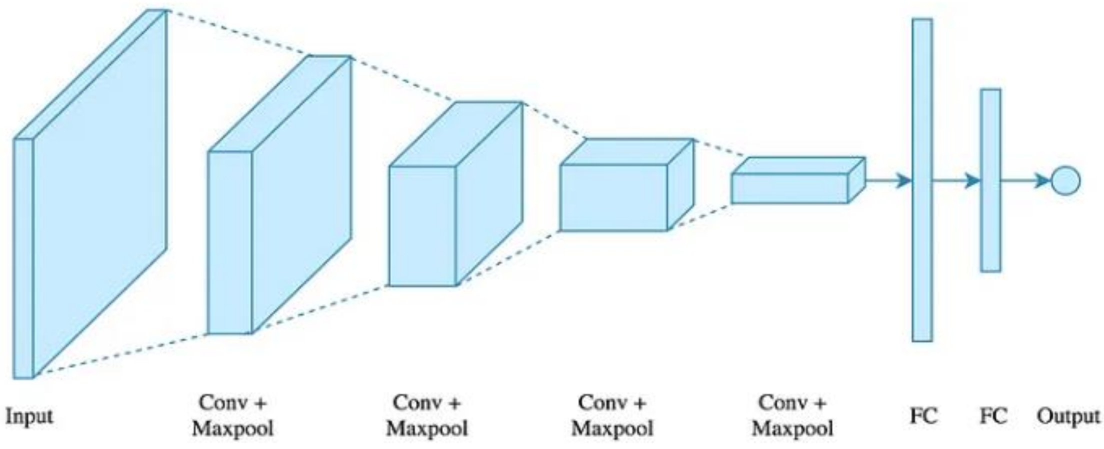

Figure 14: example of a CNN architecture. (35)

CNNs are trained using back-propagation, a gradient-based optimization algorithm that adjusts the weights of the network to minimize a loss function. The loss function measures the difference between the network's output and the desired output and is typically chosen based on the task being performed, this type of models has achieved state-of-the-art performance on a wide range of computer vision tasks including image classification, object detection and semantic segmentation. By using convolutional layers to automatically learn hierarchical representations of the input data, they are able to capture complex patterns and relationships in the data, making them a powerful tool for many applications.

8.3Vision Transformers (ViT)

Vision Transformers (ViT) are a type of deep learning model that apply the self-attention mechanism to image processing. ViT models have achieved state-of-the-art performance on a wide range of computer vision tasks, including image classification, object detection, and segmentation. Traditional convolutional neural networks (CNNs) process images using a sequence of convolutional and pooling layers to capture local spatial information. In contrast, ViT models use a self-attention mechanism that captures global spatial relationships between image regions.

The ViT model consists of a series of transformer blocks, similar to those used in natural language processing. Each transformer block contains two sub-layers: a multi-head self-attention mechanism and a feed-forward network. The self-attention mechanism is used to compute a weighted sum of the image features, taking into account the importance of each feature for the task at hand.

The figure below shows the architecture of a typical ViT model with transformer block.

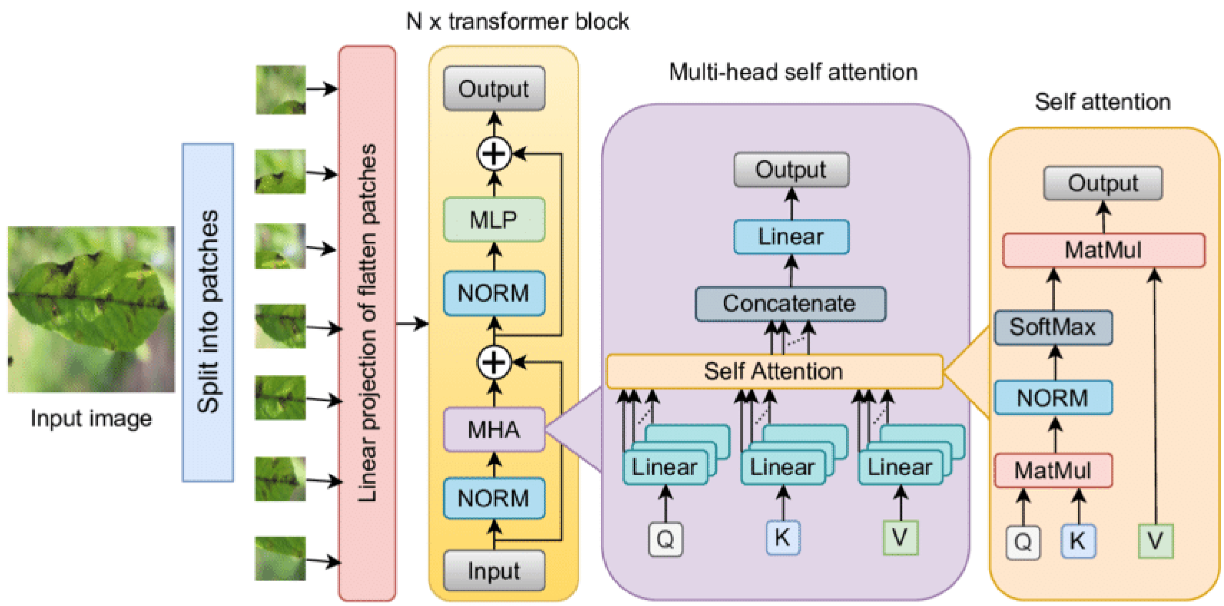

Figure 15: ViT Architecture. (36)

In a ViT model, the input image is first divided into a fixed-size grid of non-overlapping patches, each of which is treated as a separate token. Each patch is then projected to a lowerdimensional embedding space, and a positional encoding is added to capture spatial information. The self-attention mechanism in the transformer block computes a weighted sum of the embeddings where the weights are determined by a learned attention function that considers the relationships between all pairs of embeddings. This allows the model to capture global spatial relationships between image regions, rather than relying on local spatial information captured by convolutional filters. The feedforward network in the transformer block applies a non-linear transformation to the outputs of the self-attention mechanism, allowing the model to learn complex representations of the input data. After processing the image patches through multiple transformer blocks, the final output of the ViT model is passed through a fully connected layer to produce the final classification or regression result.

ViT models have achieved state-of-the-art performance on a wide range of computer vision tasks, often surpassing the performance of traditional CNN models. By using the self-attention mechanism to capture global spatial relationships between image regions ViT models are able to learn more robust and flexible representations of the input data. However, they typically require more computational resources and training data than CNN making them more challenging to train and deploy.

9. Synthesis

In the preceding section on related works, we discussed various studies and investigations conducted to predict and detect fires. These works utilized numerous approaches, some of which demonstrated outstanding performance in preventing forest fires.

Thus far, our findings indicate that the most effective strategy involves the utilization of deep learning algorithms with particular emphasis on convolutional neural networks (CNN).

The best research area for us to develop an AI-based system for preventing and managing forest fires will depend on a number of factors, such as our specific interests and resources. However, some suggestions for research areas that are currently facing significant challenges in terms of forest fire management and that may be good candidates for AI-based solutions include:

- Wild-land-Urban Interface (WUI) areas: These are areas where human settlements and natural wilderness meet and are often at high risk of fires due to the presence of both flammable vegetation and human-caused ignition sources.
- Remote and inaccessible areas: large wilderness areas, such as those found in mountainous regions, can be difficult to monitor and access, making it challenging to detect and respond to fires.

10. Conclusion

Within this chapter, we conducted an investigation into artificial intelligence techniques for forest fire prevention. We examined the strengths and weaknesses of various approaches and identified the method that we will employ in the subsequent chapter. Our focus then shifted to the analysis and design of our application, which is geared towards the detection and prediction of fires. Ultimately, our goal is to leverage these capabilities in order to prevent forest fires.

Chapter 2: Conceptual Study

1. Introduction

As previously demonstrated in the preceding chapter, there existed numerous techniques for the detection of forest fires in an effort to streamline this identification process and render it globally accessible in a cost-effective and timely manner, automatic detection systems have been introduced. However, due to the limited amount of the used dataset and the fluctuating nature of fires in this particular prevention system, the real-time application of the model may be subject to variations in accuracy based on a multitude of factors such as lighting, weather conditions, and others.

Therefore, we put forth a CNN technique with the aim of mitigating mentioned concerns and enhancing the caliber of the outcomes.

2. System Design

In this study, we propose a deep learning-based classification model using Convolutional Neural Networks (CNN) for forest fire detection. Our aim is to achieve higher precision and accuracy than existing models. To demonstrate the effectiveness of our proposed solution we evaluated several classification algorithms on a curated dataset of forest fire images.

Figure 16 depicts the comprehensive overview of our proposed Forest Fire Prevention System. The system involves developing a model through training and testing on a carefully selected dataset that consists of Fire and Non-Fire scenes. Once the model has been trained, we will evaluate its reliability and effectiveness by analyzing the quantitative results of the test data. After satisfactory performance has been achieved the model will be deployed in high-tower cameras to enable real-time fire detection. These cameras will be deployed and maintained by the forest department through a robust communication network. The cameras capture real-time images from forest landscapes and classify Fire or Non-Fire outcomes using our proposed model. These outcomes are communicated to the data center through a reliable communication system. In the event of a Fire signal (i.e., alarm) the data center immediately alerts the disaster management system or the fire brigade with the camera ID which leads to its position in the region, facilitating necessary actions to control and prevent the Fire.

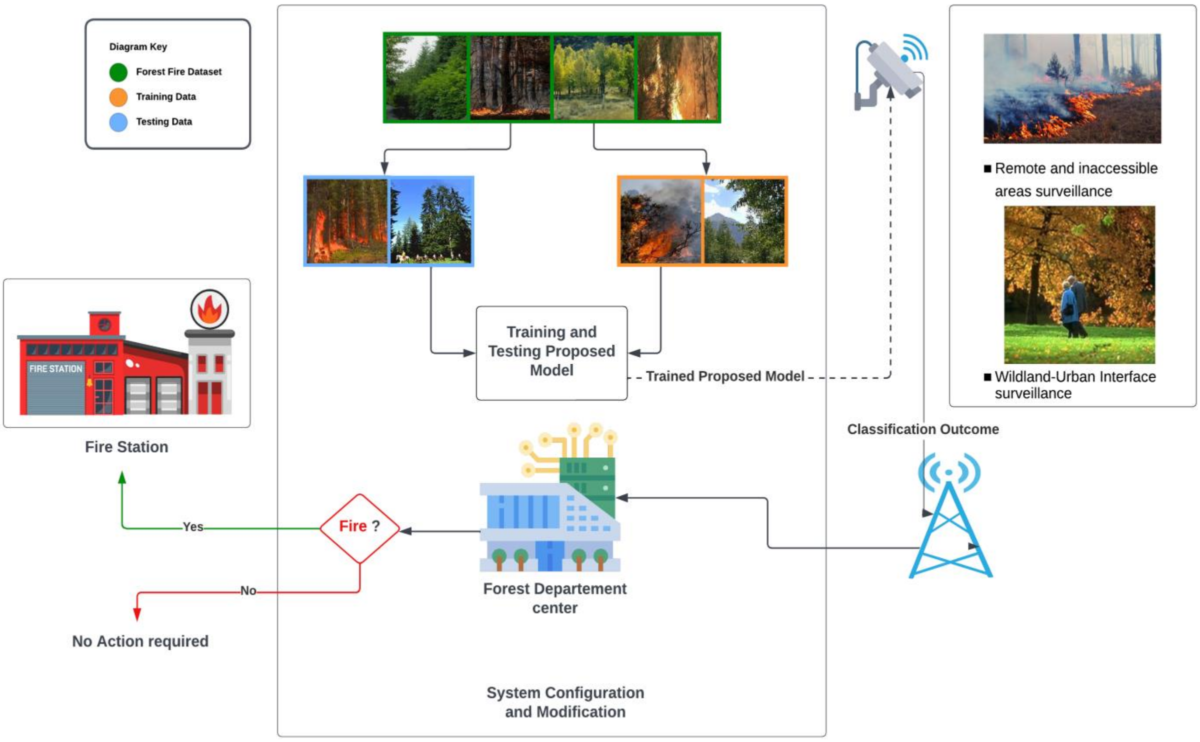

Figure 16: System Architecture

The proposed forest fire prevention system relies on a high-tower surveillance camera to capture forest scenes in real-time. Our main focus is to build a reliable Fire detection model, which can detect fires accurately and efficiently. Figure 17 depicts the workflow for creating the Fire detection Model, which involves training a customized CNN architecture on a set of Forest Fire images including both fire and non-fire scenes.

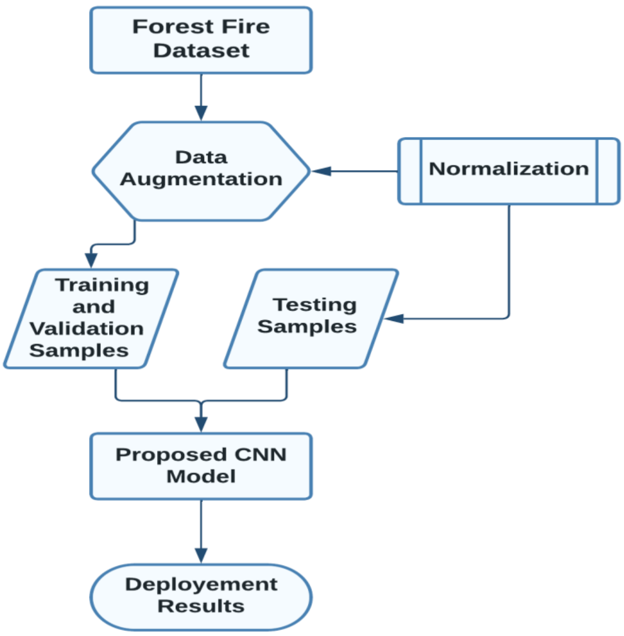

Figure 17: Workflow of the proposed method.

In case a fire is detected in the captured image, an alarm signal is transmitted to the fire department with camera id to take necessary actions. On the other hand, if there is no fire detected, the high-tower camera continues to feed the model with forest scenes for surveillance purposes. To further enhance the system's performance, it is recommended to periodically retrain the model with new and updated data to improve its accuracy and generalization ability and ensuring that the model is trained on a diverse and representative dataset. Additionally, considering the possibility of network failure or communication breakdown it would be beneficial to equip the high-tower cameras with local processing capabilities to enable autonomous fire detection even in the absence of a communication network.

Overall, our proposed system offers a promising solution for automated forest fire detection with improved accuracy and precision.

3. Presenting Dataset

In this section, we're presenting the dataset used to train the model of the proposed system.

3.1 Data Collection

A carefully curated dataset was developed to tackle the issue of forest fire detection. The dataset contains 3-channel images with a spatial resolution of 250 × 250 obtained through multiple search engines using various search terms. The images were then thoroughly examined to remove any irrelevant components such as people and fire-extinguishing machinery ensuring that only the pertinent fire region is included in each image. The figure below depicts some samples from the used dataset.

Figure 18: samples of fire and non-fire scenes from the used dataset.

3.2 Data Structure

The dataset is claimed to be balanced, and intended for binary fire or no-fire detection in the forest landscape comprising a total of 1900 images with 950 images for the training for each class. For training and testing purposes, the dataset is already divided into an 80:20 ratio.

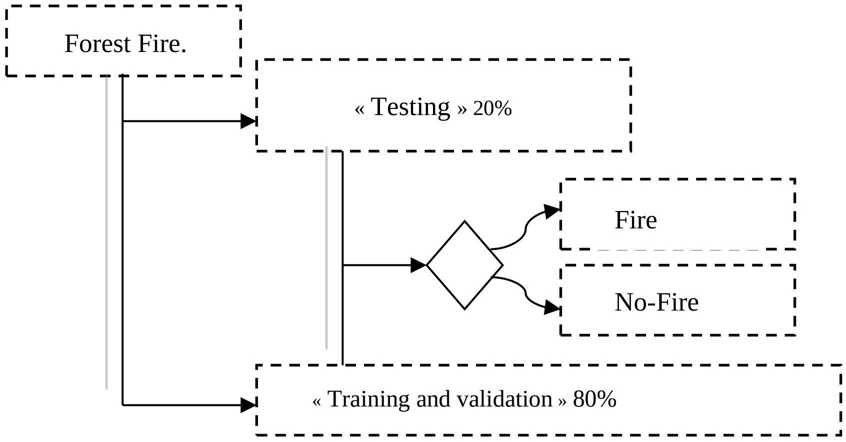

Figure 19: Structure of the dataset and ratios.

The figure 19 illustrates the hierarchical structure of the folders and their respective ratios.

3.3 Data Augmentation

Convolutional Neural Network (CNN)-based machine learning algorithms require a large amount of data to overcome the overfitting problem. Hence, Data augmentation is a promising technique to expand the dataset which assists the model to learn a function with relatively low variance. To address our problem domain, we have performed experiments with different augmentation techniques and evaluate their effectiveness in improving the model's performance, such as:

- *Horizontal flipping* on each image. The rationale behind this is that an image and its mirror version should be recognized in the same way, while vertical flipping is not always recommended and depends on the situation for example may biased by the sun's rays concluded to wrong outcomes.
- *Zooming*: This technique zooms in or out of the image. This can be useful in cases where the fire may be small or far away from the camera. By zooming in, the model can learn to recognize small fires and by zooming out it can learn to recognize larger fires.
- *Brightness/Contrast adjustment*: This technique adjusts the brightness and contrast of the image. This can be useful in cases where the lighting conditions may vary across different images. By adjusting the brightness and contrast, the model can learn to recognize fires in different lighting conditions

- *Width shift range and height shift range:* these augmentation techniques can be useful because they can simulate the possibility of a fire being present in different parts of the image. For example, a fire may appear on the left or right side of the image or at the top or bottom of the image. By randomly shifting the image horizontally and vertically the model can learn to recognize fires in different locations within the image and become more robust to variations in the location of the fire.

We have applied this augmentation process to the entire dataset, increasing the number of training samples.

4. Model Architecture

4.1 Light-Fire Model (LFM)

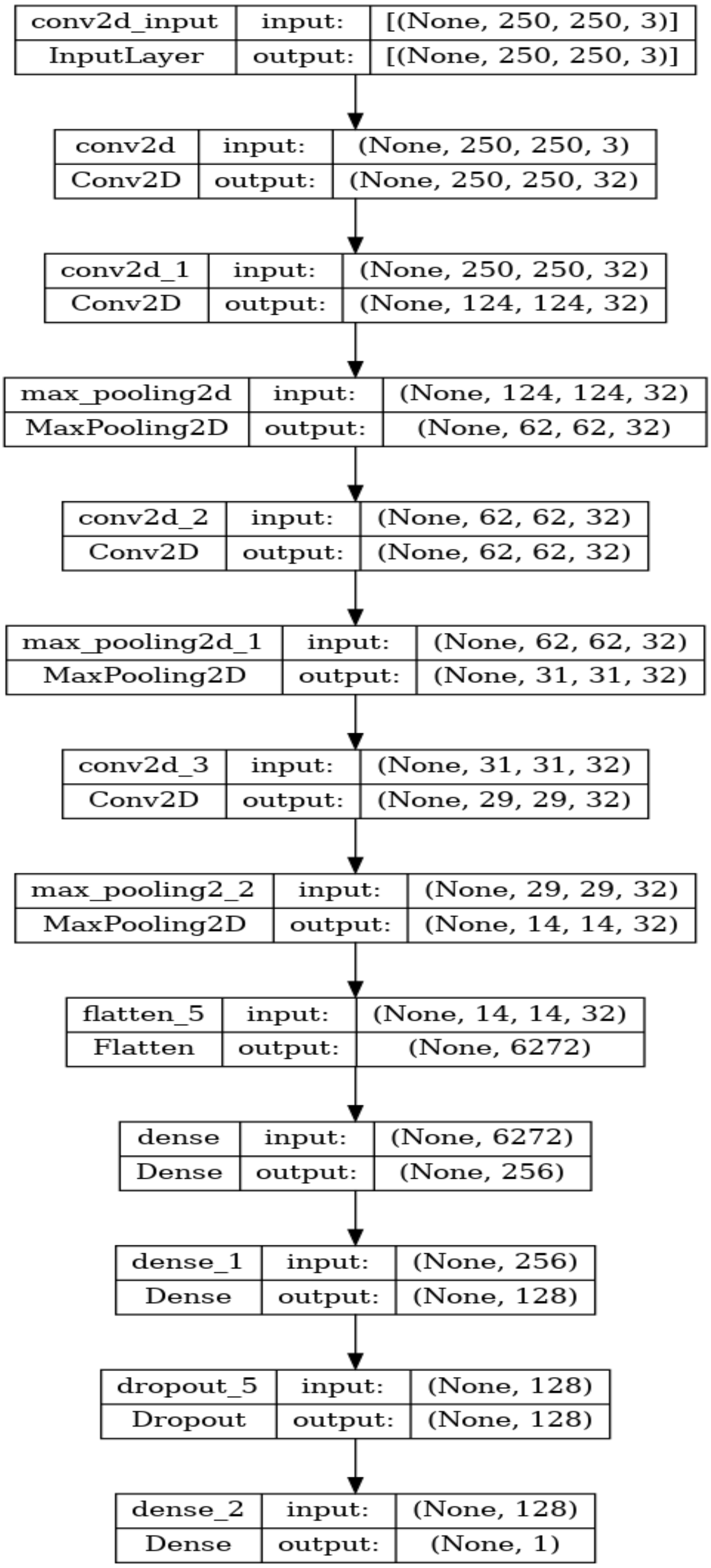

Figure 20: Light Fire Model Architecture.

In this subsection, we will explain the function of each layer type employed in the architecture of our CNN model, as illustrated in the figure 20.

Model architecture

- model = Sequential()
- model.add(Conv2D(32, (5, 5), padding='same', input_shape=(250, 250, 3), activation='relu'))
- model.add(Conv2D(32, (3, 3), strides=(2, 2), activation='relu'))
- model.add(MaxPooling2D(pool_size=(2, 2)))
- model.add(Conv2D(32, (3, 3), padding='same', activation='relu'))
- model.add(MaxPooling2D(pool_size=(2, 2)))
- model.add(Conv2D(32, (3, 3), activation='relu'))
- model.add(MaxPooling2D(pool_size=(2, 2)))
- model.add(Flatten())
- model.add(Dense(256, activation='relu'))
- model.add(Dense(128, activation='relu'))
- model.add(Dropout(0.2))
- model.add(Dense(1, activation='sigmoid'))

Figure 21: Model Creation.

- **'conv2d'**: The first convolutional layer is comprised of 32 filters with a kernel size of 5x5 and is activated using the Rectified Linear Unit (ReLU) activation function. The layer utilizes the same padding method which ensures that the output feature map has the same spatial dimensions as the input feature map. This layer acts as the input layer for the model and has an input shape of (250, 250, 3) which indicates that the input images are sized at 250 by 250 pixels with 3 channels corresponding to the RGB colors. The primary purpose of this layer is to extract relevant features from the input images by convolving each filter with the input image and applying the ReLU activation function to the resulting output feature map.
- **'conv2D_1_2_3'**: This is the second convolutional layer with 32 filters with kernel size of 3x3 and ReLU as an activation function. The 'strides' parameter is set to (2,2) which means the layer will move the filter with a step size of 2 in both height and width dimensions. The layer is applied directly to the output of the previous layer without any pooling which helps in preserving spatial information. The padding of the previous layer is set to 'same' to ensure that the input size of this layer is the same as the image size, thus avoiding any loss of spatial information. This layer plays a crucial role in extracting important features from the input images and reducing the output size of the feature maps to make the model more lightweight.

- **'Max_pooling2d_1_2'**: These MaxPooling2D layers perform max pooling with pool size of 2x2. These layers reduce the dimensionality of the output from the previous layer even more to let the next layer learn more features of the edges of the images by taking the maximum value of each 2x2 block of the output.
- **Flatten layer**: **'Flatten'** this layer flattens the output from the previous layer into a 1D vector, which can be used as input to a fully connected layer.
- **'Dense', 'Dense_1'**: Or a fully-connected layer with 256 and 128 neurons respectively, this takes the flattened output from the previous layer as input and applies the ReLU activation function to each neuron.
- **'Dropout'**: Dropout layers with a rate of 0.2 these layers randomly drop out 20% of the neurons in the previous layer to prevent over-fitting.
- **'Dense_2':** A layer with 1 neuron, this layer is the output layer of the model which uses the sigmoid activation function to produce a binary output indicating whether the input image is a fire or not.

The ReLU activation function is used in the convolutional and dense layers to introduce non-linearity and improve the model's ability to learn complex patterns in the input data. The sigmoid activation function is used in the output layer to produce a binary output between 0 and 1, which can be interpreted as the probability that the input image contains a fire.

4.2 LF Model Configuration and parameterization

```
Model: "sequential_4"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_16 (Conv2D)          (None, 250, 250, 32)      2432

 conv2d_17 (Conv2D)          (None, 124, 124, 32)      9248

 max_pooling2d_12 (MaxPoolin  (None, 62, 62, 32)       0
 g2D)

 conv2d_18 (Conv2D)          (None, 62, 62, 32)        9248

 max_pooling2d_13 (MaxPoolin  (None, 31, 31, 32)       0
 g2D)

 conv2d_19 (Conv2D)          (None, 29, 29, 32)        9248

 max_pooling2d_14 (MaxPoolin  (None, 14, 14, 32)       0
 g2D)

 flatten_4 (Flatten)         (None, 6272)              0

 dense_12 (Dense)            (None, 256)               1605888

 dense_13 (Dense)            (None, 128)               32896

 dropout_4 (Dropout)         (None, 128)               0

 dense_14 (Dense)            (None, 1)                 129

=================================================================
Total params: 1,669,089
Trainable params: 1,669,089
Non-trainable params: 0
_________________________________________________________________
```

Figure 22: LF-Model Parameters.

In this subsection we discuss the configuration of our proposed model then a look throughout the parameters of the LF-Model,

Our proposed model, the LF-Model is a deep learning model with a total of 1,669,089 learnable parameters including both weights and biases. These parameters are adjustable by the optimizer during the training process, with the aim of minimizing the loss function and ultimately improving the model's accuracy. The Trainable parameters of the model, which is also 1,669,089 refers to the number of parameters that will be updated during the learning process. This is important because the trainable parameters play a critical role in the model's performance and its ability to generalize well to new data. It is worth noting that in some cases a model may contain non-trainable parameters such as those used in batch normalization or other normalization techniques. These non-trainable parameters do not get updated during training and are therefore not counted as trainable. In summary, our LF-Model has a reasonable number of learnable parameters that can be trained to improve its performance making it a powerful tool for deep learning tasks.

4.2.2 Loss Function

We implemented the Binary cross-entropy loss function, also known as binary log loss, is a popular loss function used in binary classification tasks. It measures the difference between the predicted probability distribution and the actual probability distribution.

The formula for binary cross-entropy loss function is:

$$\mathbf{L_{binary-crossentropy}} = -\frac{\mathbf{1}}{\boldsymbol{N}}\sum_{\boldsymbol{i}=\mathbf{1}}^{\boldsymbol{N}}[\boldsymbol{Y_i}\,\mathbf{log}(\boldsymbol{\hat{y}_i}) + (\mathbf{1}-\boldsymbol{y_i})\,\mathbf{log}(\mathbf{1}-\boldsymbol{\hat{y}_i})]$$

where $\boldsymbol{Y_i}$ is the actual label of the i^{th} sample (0 or 1), $(\boldsymbol{\hat{y}_i})$ is the predicted probability of the i^{th} sample being 1, and $\boldsymbol{N}$ is the total number of samples.

The first term in the equation $\boldsymbol{Y_i}\,\mathbf{log}(\boldsymbol{\hat{y}_i})$ penalizes the model when it predicts a low probability for a positive sample (actual label is 1) while the second term $(\mathbf{1}-\boldsymbol{y_i})\,\mathbf{log}(\mathbf{1}-\boldsymbol{\hat{y}_i})$ penalizes the model when it predicts a high probability for a negative sample (actual label = 0). By summing up the losses for all samples we get the overall binary cross-entropy loss for the model.

In other words, the binary cross-entropy loss function measures how well the predicted probabilities match the actual labels. It is commonly used in logistic regression models and neural networks with sigmoid activation function in the last layer for binary classification tasks. The goal of the training process is to minimize this loss function to improve the model's performance.

4.2.1 Optimizers

Optimizers are an important component of the training process for deep learning models. They help to adjust the model parameters during training in order to minimize the error between the predicted outputs and the actual outputs. In our case, we have used four different optimizers: Stochastic Gradient Descent (SGD), Adaptive Moment Estimation (Adam), Root Mean Square Propagation (RMSprop), and Nesterov Adam (Nadam).

- SGD is a popular optimizer that updates the weights of the model by taking the gradient of the loss function with respect to the weights, multiplied by a learning rate. Momentum is also added to SGD in order to speed up convergence and prevent oscillations during training.

$$\boldsymbol{w} = \mathbf{w} - \mathbf{learning\_rate} \times \mathbf{gradient}$$

Where **w** is the weight matrix of the model, gradient is the derivative of the loss function with respect to the weights, and learning rate is a hyper-parameter that controls the size of the update step.

- Adam is a more sophisticated optimizer that uses adaptive learning rates for each weight in the model. It also uses momentum to speed up convergence, and includes a bias correction term to account for the fact that the initial estimates of the first and second moments of the gradients may be biased.

$$\mathbf{m} = \mathbf{beta\_1} \times \mathbf{m} + (\mathbf{1} - \mathbf{beta\_1}) \times \mathbf{gradient}$$

$$\boldsymbol{v} = \mathbf{beta\_2} \times \mathbf{v} + (\mathbf{1} - \mathbf{beta\_2}) \times \mathbf{gradient}^{\mathbf{2}}$$

$$\mathbf{w} = \frac{\mathbf{w} - \mathbf{learning\_rate} \times \mathbf{m}}{\sqrt{\boldsymbol{v}} + \boldsymbol{eposilon}}$$

Where **m** and **v** are the first and second moment estimates of the gradient, **beta_1** and **beta_2** are decay rates for the first and second moment estimates, respectively, **epsilon** is a small constant to prevent division by zero and all other variables have the same meaning as in SGD.

- Nadam is a variant of Adam that includes the Nesterov momentum method, which allows the optimizer to take into account the future gradient at the current time step in order to improve convergence.

$$\boldsymbol{m} = \mathbf{beta_1} \times \mathbf{m} + (\mathbf{1} - \mathbf{beta_1}) \times \mathbf{gradient}$$

$$\boldsymbol{v} = \mathbf{beta\_2} \times \mathbf{v} + (\mathbf{1} - \mathbf{beta\_2}) \times \mathbf{gradient}^{\mathbf{2}}$$

$$\hat{\mathbf{m}} = \frac{(\mathbf{beta_1} \times \mathbf{m} + (\mathbf{1} - \mathbf{beta_1}) \times \mathbf{gradient})}{(\mathbf{1} - \mathbf{beta_1}^{\mathbf{t}})} \qquad \hat{\boldsymbol{v}} = \frac{(\mathbf{beta_2} * \mathbf{v} + (\mathbf{1} - \mathbf{beta_2}) * \mathbf{gradient}^{\mathbf{2}})}{(\mathbf{1} - \mathbf{beta_2}^{\mathbf{2}})}$$

$$\boldsymbol{w} = \frac{\mathbf{w} - \mathbf{learning_{rate}} \times \widehat{\mathbf{m}}}{\sqrt{\widehat{\boldsymbol{v}}} + \boldsymbol{epsilon}}$$

Where $\widehat{\mathbf{m}}$ and $\widehat{\boldsymbol{v}}$ are biased corrected first and second moment estimates of the gradient, t is the current iteration, and all other variables have the same meaning as in Adam.

5. Conclusion

In this chapter, we have outlined the design of our Forest Fire Prevention System showcasing the step-by-step process involved in its development. We presented a detailed view of our customized CNN architecture which was optimized to detect fire shapes in the early layers while maintaining a lightweight structure. To further enhance the accuracy of our model, we employed various data augmentation techniques which allowed us to increase the amount of data and improve the generalization of the model. Our goal was to create a robust and reliable system that can be integrated with low-end devices and deployed in forest surveillance cameras. Our experiments demonstrate that while CNNs can efficiently learn the task at hand their high hardware requirements and time-consuming training process must be taken into account.

Chapter 3: Implementation and obtained results

1. Introduction

In this chapter, we will start first with a list of the development tools used in our system accomplishment. Then we present the results of our experimentations conducted to investigate the effectiveness of our proposed method. We first investigate the dataset distribution and prepare functions for data flow used for training and evaluation. Importantly, we present the results of our experiments along with a discussion of their implications. Finally, we'll present the system interface and will display some screenshots of the real-time test phase.

2. Representation of the development tools

2.1 Physical environment

In order to implement our system, we utilized the following hardware resources. Also, for the testing phase of our system we conducted experiments on a laptop personal computer (PC) equipped with the following specifications:

Personal Computer	Dell Latitude 7490
Processor	Intel(R)Core (TM) i5-8350U CPU @1.70 GHz
RAM	8 GB DDR4
Hard Drive	512 GB SSD
Operating System	Windows 10 Professionnel

Table 4: Laptop hardware specifications.

For the training phase of our model, we employed the use of Google Colab a cloud-based platform for running Python scripts. By leveraging the computational resources of Colab we were able to train our deep learning model efficiently and effectively. Additionally, we utilized Google Drive to store and access our selected dataset, which we then imported and mounted in colab for use in the training process. This allowed for seamless integration of our data with the colab environment enabling us to train our model on the cloud without the need for local storage. However, it is worth noting that Google Colab has a limitation on the maximum running time for a session which can be problematic for longer training times or for large datasets.

Google Colaboratory (Colab) Is a cloud-based platform that provides a free Jupyter notebook environment for running and sharing machine learning code. It enables users to write, run and collaborate on Python code using a web browser. Colab comes with pre-installed libraries such as ***TensorFlow***, ***Keras*** and ***PyTorch*** making it ideal for building and training deep learning models. It also provides access to powerful hardware resources such as GPUs and TPUs which significantly speeds up the training process. it is widely used in the fields of machine learning, data science and artificial intelligence research as well as in education and online courses. It has become a popular choice for researchers and students who do not have access to high-end computing resources or who want to collaborate on projects in real-time. However, one of the limitations of using Colab is that it has a time limit for each session and the resources available to each user may be limited depending on the demand. This may impact the performance and training time of complex models that require large datasets and significant computation power. (37)

Google Drive Is also a cloud-based storage and collaboration platform developed by ***Google***. It allows users to store and share files, documents, and multimedia content with others. Google Drive offers a variety of features such as file syncing across devices, real-time collaboration, and version control. Users can access their files on Google Drive from any device with an internet connection and can easily share files and folders with others by providing a link or granting them access. In particular, researchers and data scientists often use Google Drive to store and share large datasets or code repositories, as well as to collaborate on research projects. By using Google Drive, researchers can easily share their work with colleagues from different locations and can work together in real-time, improving their productivity and efficiency. (38)

2.2 Software and libraries used in the implementation

We carefully selected the tools that align with our project requirements to achieve optimal performance. The following tools have been chosen for their high efficiency and suitability for the task at hand:

Visual Studio Code (VS Code) a free and open-source code editor developed by Microsoft. It provides developers with a variety of features such as syntax highlighting, code completion, debugging, version control integration and extension support making it suitable for various programming languages and development tasks. VS Code is available on multiple platforms including Windows, macOS and Linux. It has become a popular choice among developers due to its ease of use, flexibility and extensibility. (39)

Python: is a high-level, interpreted programming language that was first released in 1991. It was designed with an emphasis on code readability and ease of use, making it a popular language for beginners and experts alike. Python's syntax is clean and straightforward, making it easy to learn and use. It has a large and active community of developers who contribute to its growth and development. It has become the go-to language for many scientific, statistical, and data analysis applications, including deep learning. Its popularity in the field of artificial intelligence (AI) is due in large part to its many powerful libraries for machine learning such as TensorFlow, PyTorch and Keras. These libraries provide easy-to-use interfaces for building and training deep learning models, as well as tools for data preprocessing, model visualization and more. (40)

Tensorflow: is an open-source software library used for dataflow and differentiable programming across a range of tasks. It is a symbolic math library and is also used for machine learning applications such as neural networks for various fields such as natural language processing, image recognition, and predictive analytics. TensorFlow provides a variety of tools and APIs for building, testing, and deploying models at scale across multiple platforms and devices. It is widely used in the industry and academia for research and production purposes. (41)

Keras: is an open-source neural network library written in Python and It is designed to provide a user-friendly interface for building and training deep learning models also acts as an interface for the TensorFlow library making it easier to use for developers who are not familiar with TensorFlow's low-level API. It provides a high-level API that allows for the rapid prototyping of deep learning models that made it a popular choice among researchers and practitioners. Keras supports a wide range of neural network architectures including convolutional neural networks, recurrent neural networks and multi-layer perceptrons. It also includes a variety of built-in loss functions, optimizers and metrics for model evaluation. Additionally, Keras is known for its ease of use as it allows developers to quickly build and test different models without having to write a lot of boilerplate code. (42)

NumPy: is a powerful Python library that provides a high-performance multidimensional array object and tools for working with these arrays. It is widely used in scientific computing and data analysis and provides a flexible interface to efficiently store and manipulate large datasets. NumPy also includes a number of mathematical functions to perform operations on these arrays including linear algebra, Fourier transform and random number generation. Its performance and ease of use make it a must choice for numerical computations in many fields including machine learning, data science and physics. (43)

Matplotlib: is a Python library for creating visual representations of data that can be either static or dynamic. It is a 2D plot library that generates figures suitable for publication. It has many features for plot style, formatting and layout customization. Also provides a vast array of plot types including line, scatter, bar, histogram, heatmap, 3D and subplots. Matplotlib is extensively used for data analysis in scientific and engineering fields and is a crucial tool in the data science part of Python. (44)

OpenCV (Open-Source Computer Vision): is an open-source computer vision and machine learning software library designed to help developers create computer vision applications. It was originally developed by Intel in 1999 and is now maintained by the OpenCV community. OpenCV provides a vast array of computer vision algorithms, image and video processing tools, and machine learning models that can be used for various tasks, such as object detection, facial recognition, image segmentation, and more. (45)

PyQt 5 (Python Qt): a Python binding for the Qt application framework which is a popular cross-platform toolkit for developing desktop applications with graphical user interfaces (GUIs)and allows Python developers to create desktop applications with rich UI (User Interface) features using the extensive functionality provided by the framework such as windowing, event handling, layout management and various UI controls. Also provides a comprehensive set of Python modules that enable seamless integration with Qt's C++ libraries allowing developers to create powerful and interactive desktop applications. It is widely used for building cross-platform desktop applications in Python and offers a wide range of capabilities for creating modern and professional-looking GUI applications.

3. Experimental Setup

3.1 Preparations and workflow of the data

The data preparation for image classification model training involves several steps. First, we set the image dimensions and number of classes for our model. We then load the test and training data from their respective directories. For each class directory and image file in it, we use the OpenCV library to read the image file and resize it to the specified dimensions. We then append the image to the appropriate data array, along with its corresponding label.

After loading the data, and to increase the diversity of our training data we also perform data augmentation by applying various transformations to the images in our training data. This helps the model to better generalize to new images that it has not seen before. We preprocess it by converting the pixel values of the images in both the train and test data arrays to floating point numbers between 0 and 1 by dividing each pixel value by 255 which helps the model speed up the process of learning and converging better.

We use the keras Image Data Generator to apply random transformations, such as horizontal and vertical flipping, rotation, and zooming, to our training images. This generates new images on-the-fly during training, so our model is trained on a larger and more diverse dataset.

3.2 Evaluation Metrics

We implemented our proposed method in Python, using the keras deep learning framework. We conducted experiments using the training set and evaluated the performance of our method on the testing set. We used the following evaluation metrics to measure the performance of our method

- *Accuracy:* The accuracy is the proportion of correctly classified samples out of all the samples in the dataset. The formula for accuracy is:

  Accuracy = (number of correctly classified samples) / (total number of samples)

- *Precision:* The precision is the proportion of correctly classified positive samples out of all samples classified as positive. The formula for precision is:

  Precision = (number of true positives) / (number of true positives + number of false positives)

- *Recall:* The recall is the proportion of correctly classified positive samples out of all positive samples in the dataset. The formula for recall is:

  Recall = (number of true positives) / (number of true positives + number of false negatives)

- *F1 score:* The F1 score is the harmonic mean of precision and recall combines both precision and recall into a single metric that balances between them. The formula for F1 score is:

  F1 score = 2 * (precision * recall) / (precision + recall)

- *Confusion matrix:* is a table used to evaluate the performance of a classification model. It shows the number of true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN) for each class. In a binary classification problem, such as the one in our example, the confusion matrix is a 2x2 table. The columns of the matrix represent the predicted class labels, while the rows represent the true class labels. The TP and TN cells represent correct predictions, while the FP and FN cells represent incorrect predictions.

  Where in the above formulas the true positives (TP) represent the number of correctly classified positive samples, false positives (FP) represent the number of negative samples classified as positive (predicted as fire but the true label's no fire), and false negatives (FN) represent the number of positive samples classified as negative (predicted as no fire but the true label's fire).

- *Receiver Operating Characteristic (ROC) curve* is a plot that shows the performance of a binary classification model at different classification thresholds where the curve is created by plotting the True Positive Rate (recall) against the False Positive Rate (FPR) at various threshold settings. The TPR is the proportion of true positives that are correctly identified as such by the model. The FPR is the proportion of false positives that are incorrectly identified as positive by the model. A perfect classifier would have a TPR of 1 and an FPR of 0, resulting in a curve that hugs the top left corner of the plot. A classifier with random performance would have a diagonal line from the bottom left to the top right of the plot.

- *The area under the ROC curve (AUC)* is a commonly used metric for evaluating the performance of a binary classification model. A model with an AUC of 0.5 performs no better than random guessing, while a model with an AUC of 1.0 has perfect performance.

In summary, all of these metrics are important tools for evaluating the performance of a binary classification model, and can provide valuable insights into its strengths and weaknesses.

5. Discussion and comparison of the obtained results

In this subsection, we will discuss the different models and techniques that we have employed in our experimental setup. We began by training a baseline model using a light-weight convolutional neural network (CNN) architecture through variety of testing techniques and parameters. We then experimented with different CNN architectures such as VGG16, InceptionV3, and ResNet50 to see how they perform on the used dataset. Finally, we evaluated ensemble learning techniques such as a CNN combined with an SVM or a Vision Transformer.

5.1 Baseline LF-Model Results and Evaluation

The LF-model which serves as the baseline model was trained with the Adam optimizer and various hyper-parameters. The optimizer's learning rate was set to 0.0001 to ensure the model converged to an optimal solution. The model was trained with a batch size of 32 over a total of 80 epochs.

To prevent over-fitting the Early Stopping function was implemented with a patience of 7 epochs monitoring the validation accuracy. Additionally, the Reduce Learning Rate on Plateau (ReduceLROnPlateau) function was also implemented to further improve the model's performance. This function monitors the validation loss and reduces the learning rate if the validation loss does not improve after a certain number of epochs.

The learning rate is a hyper-parameter that determines the step size at each iteration of the optimization process. By reducing the learning rate, we allow the optimization process to take smaller steps towards the minimum of the loss function, which can help improve the accuracy of the model.

The model's performance was evaluated using multiple metrics including accuracy score, precision, recall, confusion matrix and F1 score. These metrics provide a comprehensive analysis of the model's ability to predict accurately and reliably.

Next, we'll present the evaluation results of our trained model using two different approaches. The first approach involved applying data augmentation techniques to the training dataset, while the second approach used the same model architecture without data augmentation. In both cases we split the training data into a validation set with a ratio of 60:20 while the left 20% for testing.

- **With Augmentation techniques**

  For the first approach, we used the same architecture as the baseline model but applied data augmentation techniques such as rotation, zooming, width and height shifting and horizontal flipping to the training and validation images, we evaluated the model's performance using various metrics such as accuracy score, precision, recall and F1 score.

	No Fire	**Fire**	**Accuracy**	**Macro-avg**
Precision	0.95	0.89	-	0.92
Recall	0.94	0.90	-	0.92
F1-score	0.92	0.92	**0.92**	0.92

Table 5: LF-Model Evaluation using Data Augmentation.

Based on the classification report, the model achieved an overall accuracy of 0.92 on the test data, which is a reasonable performance. Looking at the precision, recall, and F1-score for each class, we can see that the model performs better on the "no fire" class compared to the "fire" class.

- The precision for "no fire" is very high at 0.95 meaning that out of all the predictions the model made for "no fire", 95% were correct. However, the precision for "fire" is lower at 0.89 meaning that out of all the predictions the model made for "fire", only 89% were correct.
- The recall for "no fire" is also reasonably high at 0.94, meaning that out of all the actual "no fire" instances in the test data, the model correctly identified 94% of them. However,

the recall for "fire" is much less at 0.90, meaning that out of all the actual "fire" instances in the test data, the model correctly identified 90% of them.

- The F1-score is a harmonic mean of precision and recall, and is a good overall measure of the model's performance. The F1-score for "no fire" is 0.92, while the F1-score for "fire" is equal at 0.92.
- The macro-average of precision, recall, and F1-score is 0.92, which is calculated by averaging the precision, recall, and F1-score across both classes. This indicates that the model has a reasonable overall performance, but there is still room for improvement, particularly for the "fire" class.

Based on the augmentation parameters used, the data was augmented with rotation, horizontal and vertical shift, shear and zoom. These augmentations can help to improve the model's ability to generalize to new data by increasing the variability in the training data. However, it's also possible that some of these augmentations could be detrimental to performance if they introduce too much noise or distortion.

Overall, the results suggest that the model could benefit from further tuning of the hyper-parameters and exploration of different augmentation strategies to improve its performance on the "fire" class.

- **Without Augmentation techniques**

For the second approach, we used the same architecture and the same parameters as the baseline model, but we did not apply any data augmentation techniques to the training data.

	No Fire	**Fire**	**Accuracy**	**Macro-avg**
Precision	0.98	0.96	-	0.97
Recall	0.95	0.99	-	0.97
F1-score	0.97	0.97	**0.97**	0.97

Table 6: LF-Model Evaluation without Data Augmentation.

The classification report provides a comprehensive evaluation of our model's performance. It includes metrics such as precision, recall, and F1-score for each class (no fire and fire) as well as the overall accuracy of the model.

In our results, the classification report shows that our model achieved an overall accuracy of 97.21%. This means that our model correctly predicted the fire or no fire status of 97.21% of the samples in our test set.

Looking at the precision and recall for each class, we can see that the model performs well for both classes. For the no fire class precision is 0.98 and recall is 0.95, indicating that the model correctly identifies a high proportion of no fire instances and has a low false positive rate. For the fire class, precision is 0.96 and recall is 0.99, indicating that the model correctly identifies a high proportion of fire instances and has a low false negative rate.

The F1-score is also high for both classes with values of 0.97 for no fire and 0.97 for fire. This means that the model performs well in terms of both precision and recall for both classes.

In summary, the classification report shows that our model achieved high accuracy and performs well in identifying both fire and no fire instances.

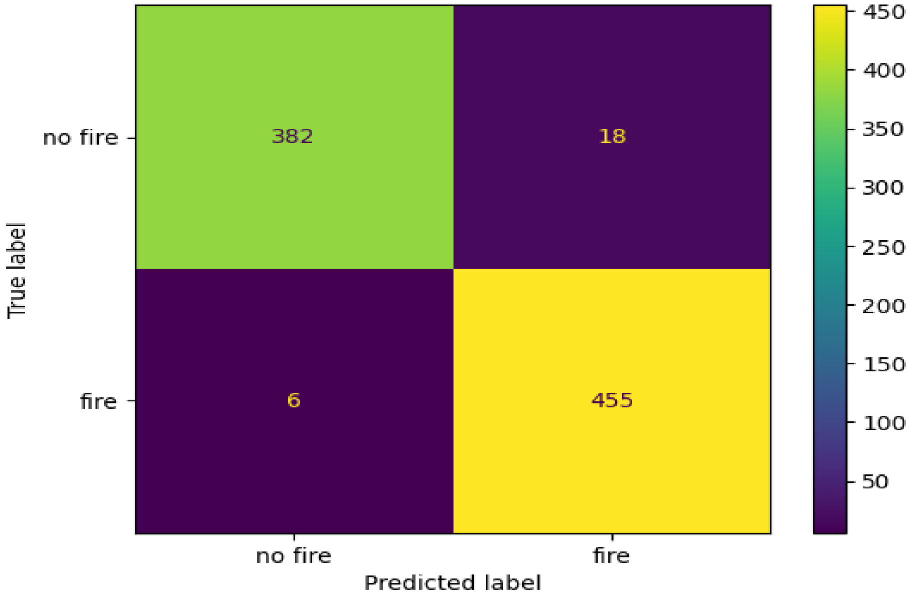

Figure 23: LF-Model Confusion Matrix.

The confusion matrix shown above has two classes: "no fire" and "fire". The rows of the matrix represent the true labels of the data, while the columns represent the predicted labels of the data. Therefore, in this case, there were 400 instances of "no fire" in the true labels, out of which 382 were correctly classified as "no fire" (true negatives), while 18 were misclassified as "fire" (false positives). Similarly, there were 461 instances of "fire" in the true labels, out of which 455 were correctly classified as "fire" (true positives), while 6 were misclassified as "no fire" (false negatives).

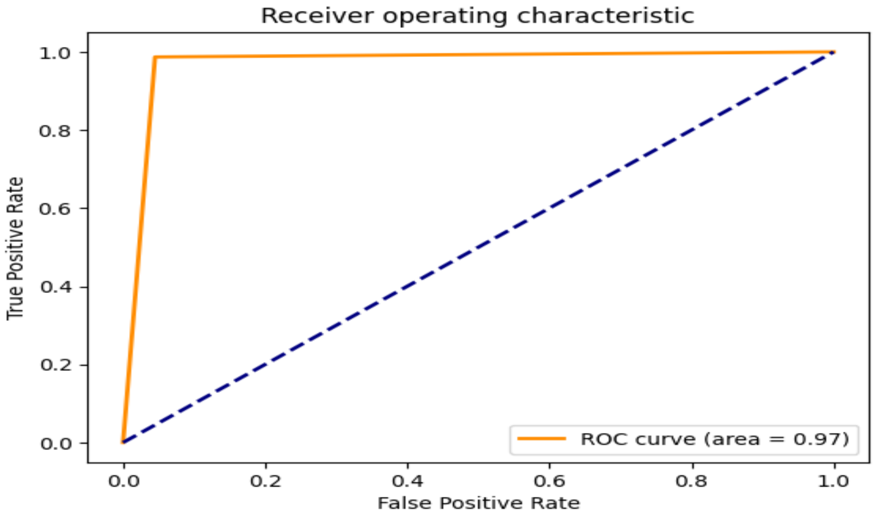

Figure 24: LF-Model ROC Curve.

The ROC (Receiver Operating Characteristic) curve is a graphical representation of the performance of a binary classifier as the discrimination threshold is varied. An ROC curve is created by plotting the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings. In this case, the area under the ROC curve (AUC) is 0.97, which indicates a very good performance of the classifier.

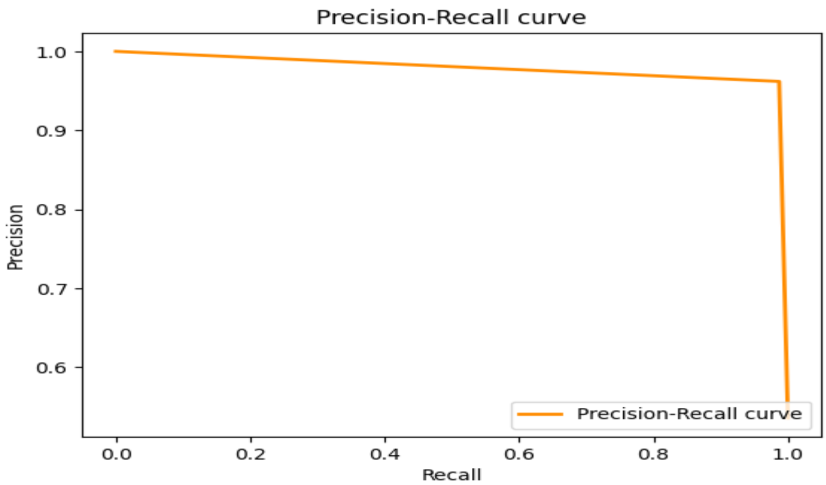

Figure 25: LF-Model Precision-Recall Curve.

The precision-recall curve shows the trade-off between precision and recall at different classification thresholds. In this case, the precision-recall curve shows that as the threshold is increased, the precision and recall values also increase, indicating that the classifier is performing well at correctly classifying both "no fire" and "fire" instances. The precision and recall values both approach 1.0 at the upper right corner of the curve, indicating a high level of performance.

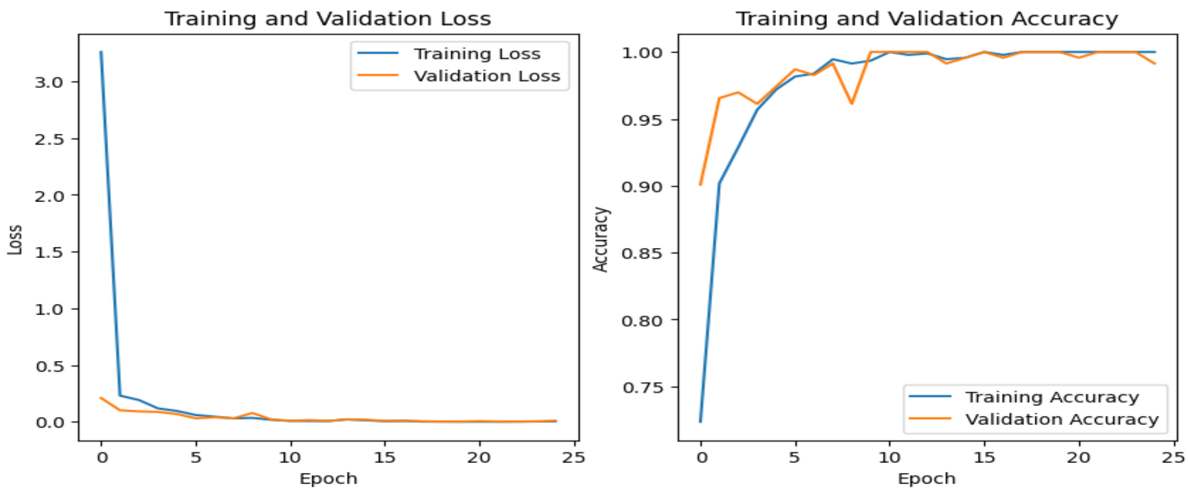

Figure 26: LF-Model Training and Validation Plots.

The training process of the model started with an initial loss of 3.2594 and an accuracy of 0.7235 on the training set. The validation loss was 0.5098 with a validation accuracy of 0.9009. As the training progressed, the model's loss decreased and accuracy increased with each epoch. The final loss was 0.0035 with an accuracy of 1.0 on the training set, and the validation loss was 0.0101 with a validation accuracy of 0.9914. These results indicate that the model is performing very well on both the training and validation sets.

The loss plot shows the decrease in loss over each epoch while the accuracy plot shows the increase in accuracy. In this case, both plots show a steady decrease/increase in their respective values which indicates that the model is learning and improving with each epoch. This is a healthy process of training without any sign of over-fitting.

The early stopping function was applied with a patience of 7 epochs. This means that if the validation loss does not improve for 15 consecutive epochs, the training process will stop. This is done to prevent over-fitting and ensure that the model generalizes well on unseen data.

Overall, the plots and results show that the model is performing well and the training process is healthy, with no signs of over-fitting.

5.2 Comparison of Different CNN Architectures

The performance of a Convolutional Neural Network (CNN) model highly depends on its architecture and the number of layers it comprises. In this subsection, we present a comparison of the performance of three popular CNN architectures namely VGG16 and InceptionV3 for our image classification task. We employed the same hyper-parameters and training settings for each architecture to ensure a fair comparison. We evaluated the models using several metrics such as accuracy, precision, recall, and F1 score. We also provide a table and graph summarizing the performance to facilitate easy comparison.

5.2.1 Training and Evaluation of VGG16 Model

VGG16 is a convolutional neural network architecture known for its simplicity and its strong performance in image classification tasks, introduced in 2014 by researchers at the Visual Geometry Group at the University of Oxford. The architecture consists of 16 layers with 13 convolutional layers and 3 fully connected layers. The convolutional layers are all 3x3 filters with the number of filters increasing from 64 to 512 as the depth of the network increases. Also includes max pooling layers and dropout layers to reduce over-fitting. (46)

After loading the pre-trained VGG16 model with the weights of the ImageNet dataset and freezing the feature extraction layers, we added our own MLP head for classification with the same hyper-parameters. The results showed that the model achieved a high accuracy score on the test set, indicating that VGG16 is a powerful architecture for image classification tasks.

In order to comprehensively evaluate the performance of the model, we calculated a variety of metrics.

	No Fire	Fire	Accuracy	Macro-avg
Precision	0.99	0.98	-	0.99
Recall	0.98	1.00	-	0.99
F1-score	0.99	0.99	**0.98**	0.99

Table 7: VGG 16 Model Evaluation Metrics.

- This evaluation report shows the performance metrics of the VGG16 model. The accuracy score is 0.9884, indicating that the model correctly classified 98.84% of the test samples. The precision for the "no fire" class is 0.99, meaning that 99% of the samples classified as "no fire" are actually "no fire". The recall for the "fire" class is 1.00, indicating that the model correctly identified all the "fire" samples. The F1-score for both classes is 0.99, which is a harmonic mean of precision and recall. Overall, the weighted average F1-score is 0.99, demonstrating the high performance of the VGG16 model in classifying fire and non-fire images.

Figure 27: VGG 16 Model Training and Validation Plots.

- The training process of VGG16 began with an initial epoch where the loss and accuracy values were recorded as 0.3786 and 0.9178, respectively, while the validation loss and validation accuracy were 0.0605 and 0.9862, respectively. The training process was stopped early with a patience of 5 since the VGG16 model's base had already been trained. The final epoch of the model resulted in a loss value of 0.0014 and an accuracy of 0.9993. The validation loss and validation accuracy were 0.0495 and 0.9890, respectively.

5.2.2 Training and Evaluation of InceptionVersion3 Model

InceptionV3 is a convolutional neural network architecture developed by Google. It is an improvement over the original Inception model and is designed to be more computationally efficient while still achieving high accuracy. The architecture is based on the idea of using "inception modules" which allow the network to learn multiple levels of abstraction in parallel. These modules consist of several convolutional layers with different filter sizes as well as pooling and concatenation layers. It’s also including other features such as batch normalization and residual connections to improve training stability and performance. (47)

With the same steps done, we computed a range of metrics to assess the performance of inception V3:

	No Fire	Fire	Accuracy	Macro-avg
Precision	0.75	0.94	-	0.85
Recall	0.95	0.73	-	0.84
F1-score	0.84	0.82	**0.83**	0.83

Table 8 :Inception V3 Model Evaluation Metrics.

- These results for Inception V3 show an accuracy score of 0.829 indicating that the model is able to correctly classify fire and no fire instances with an accuracy of 83%. The precision and recall scores for both classes are also above 0.7 which suggests that the model is performing well in terms of correctly identifying positive instances (fires) and negative instances (no fires). However, it's important to note that the performance of Inception V3 is not as good as that of VGG16 as the accuracy score and other evaluation metrics are lower. Overall, while the training of Inception V3 appears to be healthy, there may be room for improvement in terms of its performance.

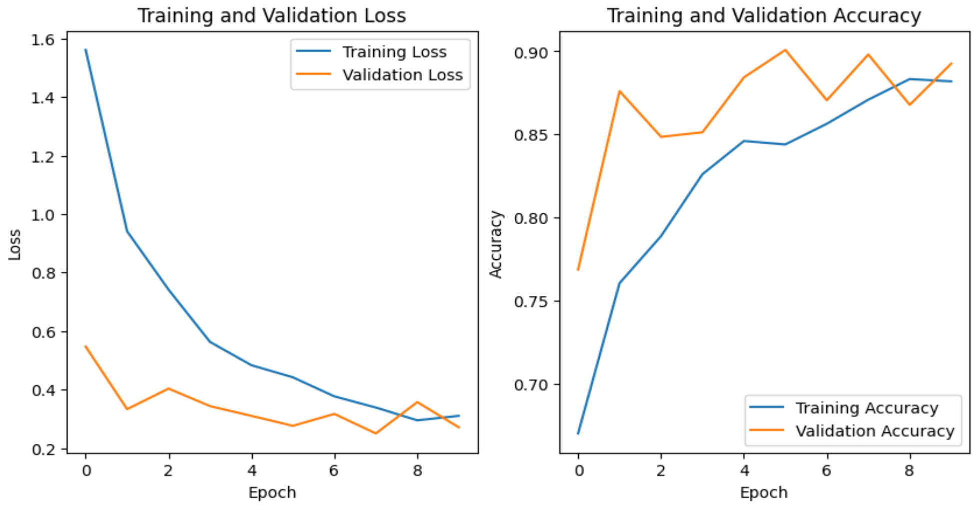

Figure 28: Inception V3 Model Training and Validation Plots.

- The training of Inception V3 started with a loss of 1.5605 and accuracy of 0.6699 in the first epoch, early stopping was applied with a patience of 5 as the base of Inception V3 was already trained. The final epoch had a loss of 0.3101 and accuracy of 0.8819. Overall, this training result seems to be decent, with a final validation accuracy of 0.8926. However, the training time per step is relatively long compared to VGG16, which may not be ideal in certain scenarios.

5.3 Evaluation of Ensemble Learning Techniques

In this section, we explore the evaluation of ensemble learning techniques, specifically the combination of a CNN (Convolutional Neural Network) with an SVM (Support Vector Machine) and a Vision Transformer.

5.3.1 Training and Evaluation of Vision Transformers Model

To train the Vision Transformer, we used ensembles of ordered functions with specific hyper-parameters. The learning rate was set to 1e-3, weight decay to 0.0001, batch size to 128 and the total number of epochs to 80. The input images of the utilized dataset were resized to a dimension of 75x75 pixels and patches of size 15x15 were extracted from these images. The projection dimension was set to 64 and there were 4 attention heads in the transformer. The transformer consisted of 4 layers and the dense layers of the final classifier had sizes of (256, 128).

The training process involved augmenting the data, creating patches from the input images, and encoding these patches using the Vision Transformer. Multiple layers of the Transformer block were used, which included layer normalization, multi-head attention, skip connections, and MLP (Multi-Layer Perceptron) layers. The final representation was obtained by flattening the encoded patches, applying dropout, and passing it through an MLP. The model was compiled using the Adam optimizer with a specific loss function (Binary Cross entropy) and accuracy as the metric. Early stopping was implemented with a patience of 20 epochs.

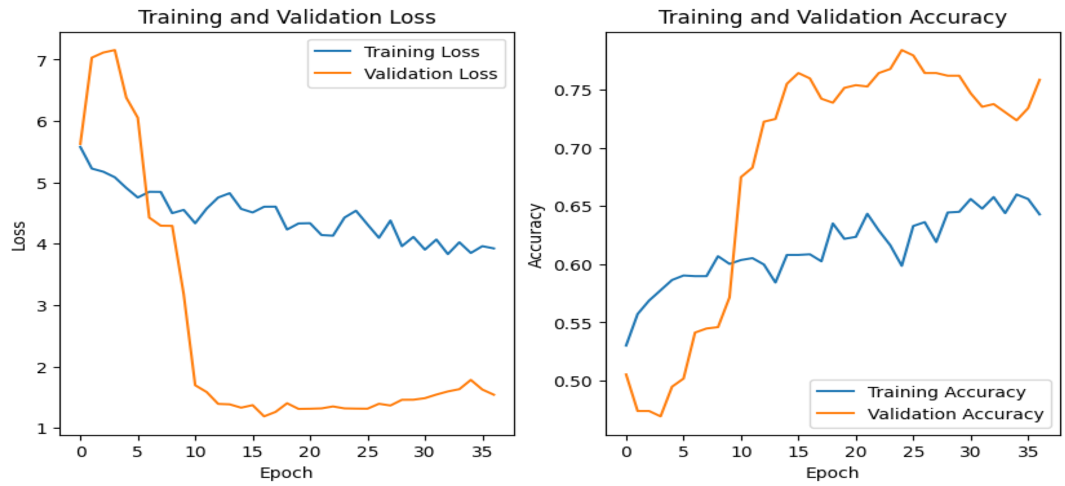

Figure 29: ViT Model Training and Validation Plots.

Based on the provided logs, we can see that the initial accuracy of the vision transformer was 0.50 and the final accuracy after 80 epochs with early stopping was 0.7584. The training loss decreased from 5.57 in the first epoch to 3.92 in the last epoch before early stopping. However, the validation loss decreased from 5.60 in the first epoch to 1.54 in the last epoch before early stopping.

Metric	Value
Accuracy	75.84%
Precision	81.38%
Recall	71.14%
F1 score	75.92%

Table 9: ViT Model Evaluation Metrics.

The advantages of Vision Transformers are that they can handle images of arbitrary sizes and scales. They are also computationally efficient as they do not require as much computation as Convolutional Neural Networks (CNNs). Additionally, the attention mechanism of Vision Transformers allows them to capture the global dependencies between image patches, making them suitable for tasks that require long-range dependencies. However, one disadvantage of Vision Transformers is that they require a large amount of training data and also most importantly require a longer training time than CNNs which can make them impractical for our applications. Additionally, Vision Transformers are more sensitive to hyperparameters so it can be difficult to find the optimal set of hyperparameters for a given task.

5.3.2 Training and Evaluation of LF-Model (CNN) + SVM Model

We performed training and evaluation on the LF-Model (our CNN model) with an SVM classifier. The next steps explain the process:

- We created a new model called feature extractor using the Keras functional API. This model takes the same input as the original model but outputs the activations from the second-to-last layer of the original model. By doing this we obtained a model that extracts features from the convolutional layers of the LF-Model.
- We used the feature extractor to extract features from the training and validation data. This was done by calling the predict method of the feature extractor model on the respective data sets. The extracted features were stored in training features and validation features.
- We initialized an SVM classifier (SVM) using the SVC class from scikit-learn.
- We trained the SVM classifier (SVM) on the extracted features (training features) and corresponding labels (training labels) using the fit method.
- Finally, we evaluated the trained SVM classifier on the validation features (validation features) and computed the accuracy using the score method. The obtained SVM accuracy was printed as SVM Accuracy: 98,62 %.

In summary, we leveraged the trained LF-Model to extract meaningful features from the convolutional layers and then utilized these features to train an SVM classifier. By evaluating the SVM classifier on the validation data, we obtained a high accuracy score, indicating the effectiveness of the combined LF-Model and SVM approach for the given task.

5.4 Comparing our obtained results with other models

5.4.1 Comparison of LF-Model with implemented architectures

We compare the LF-Model with different architectures that were implemented in our experiments. These architectures include VGG16, Inception V3, Vision Transformers, and our model combined with SVM. We evaluate and compare the performance of each model based on various metrics such as accuracy, precision, recall, and F1-score. We analyze the results to identify which architecture achieved the best performance in terms of accurately predicting and preventing forest fires. We consider factors such as model complexity, training time, and the ability to handle different types of data.

The evaluation results of different models and techniques are summarized in the table below:

Method	Accuracy	Precision	Recall	F1 score
LF-Model	97.21%	97%	97%	97%
VGG16	**98.84%**	**99%**	**99%**	**99%**
Inception V3	82.9%	85%	84%	83%
Vision Transformers	75.84%	81.38	71.14	75.92
LF-Model + SVM	98,62 %	98.0%	98.1%	98%

Table 10: Comparison of LF-Model with implemented Models.

Based on the evaluation metrics, it is evident that the LF-Model achieved an accuracy of 97.21% while VGG16 performed even better with an accuracy of 98.84%. However, VGG16 is computationally complex and may not be the ideal choice due to its high computational requirements.

In comparison, Inception V3 achieved an accuracy of 82.9%, which is lower than LF-Model and VGG16. While Vision Transformers achieved an accuracy of 75.84%, indicating lower performance compared to the other models. Vision Transformers may require further optimization to improve its performance and a reasonable amount of data.

Furthermore, LF-Model combined with SVM achieved an accuracy of 98.62%, which is comparable to VGG16 but with the added advantage of simpler model architecture. However, it is important to note that LF-Model + SVM has the disadvantage of increased computational time in real-time implementations, as it involves passing frames through the LF-Model to extract features and then feeding them into a separate SVM model.

Considering the trade-offs between accuracy, computational complexity, and real-time implementation, it is recommended for us to choose the LF-Model as it achieves a good balance between performance and efficiency.

5.4.2 Comparison of LF-Model with Existing works

In the second subsection, we conduct a comprehensive comparison between our LF-Model and existing works in the field of preventing forest fires. Through an extensive review and analysis of relevant literature, we have gained insights into the various approaches employed in previous research. By comparing the performance of our model with the reported results in these studies, we carefully evaluate its efficacy using key evaluation metrics such as accuracy, precision, recall, and F1-score. Furthermore, we meticulously assess the strengths and weaknesses of our model in relation to the existing works, emphasizing any noteworthy contributions or enhancements that we have introduced. This thorough comparison allows us to ascertain the effectiveness of our model in addressing the crucial challenge of preventing forest fires and its potential implications in the field.

Method	Accuracy	Precision	Recall	F1 score
LF-Model	97.21%	97%	97%	97%
(19)	94.39%	82%	98%	89%
(20)	95.86%	100%	99%	99%
(21)	97.6%	97%	98%	98%

Table 11: Comparison of LF-Model with Existing Models.

In comparing our LF-Model to existing models, we have achieved notable results and made significant contributions in addressing the challenge of preventing forest fires.

The LF-Model demonstrated an accuracy of 97.21%, with precision and recall both at 97%, and an F1 score of 97%. These metrics indicate the model's ability to accurately classify fire and non-fire instances.

- Comparing our model to Model 19, we outperformed it in terms of accuracy (97.21% vs. 94.39%). Additionally, our model showed higher precision (97% vs. 82%) and comparable recall (97% vs. 98%), resulting in a higher F1 score (97% vs. 89%). This suggests that our LF-Model is more reliable and effective in detecting and preventing forest fires.
- In comparison to Model 20, our LF-Model achieved a slightly higher accuracy (97.21% vs. 95.86%). Furthermore, our precision (97% vs. 100%), recall (97% vs. 99%), and F1 score (97% vs. 99%) were comparable or slightly lower. These results indicate that our LF-Model performs well and is on par with Model 20, demonstrating its effectiveness in forest fire prevention.
- When comparing our LF-Model to Model 21, we achieved a similar accuracy (97.21% vs. 97.6%). The precision (97% vs. 97%) and recall (97% vs. 98%) were also similar, resulting in a comparable F1 score (97% vs. 98%). This suggests that our LF-Model performs at a similar level to Model 21 in detecting and preventing forest fires.

Overall, our LF-Model exhibits strong performance and compares favorably to existing models in terms of accuracy, precision, recall, and F1 score. The meticulous assessment of our model in relation to other existing models highlights its effectiveness and potential implications in the field of forest fire prevention.

6. System Interface

In the System Interface section, we present the user interface of our application showcasing its key functionalities. The application consists of a single page with an intuitive design that allows users to easily interact with the classification model.

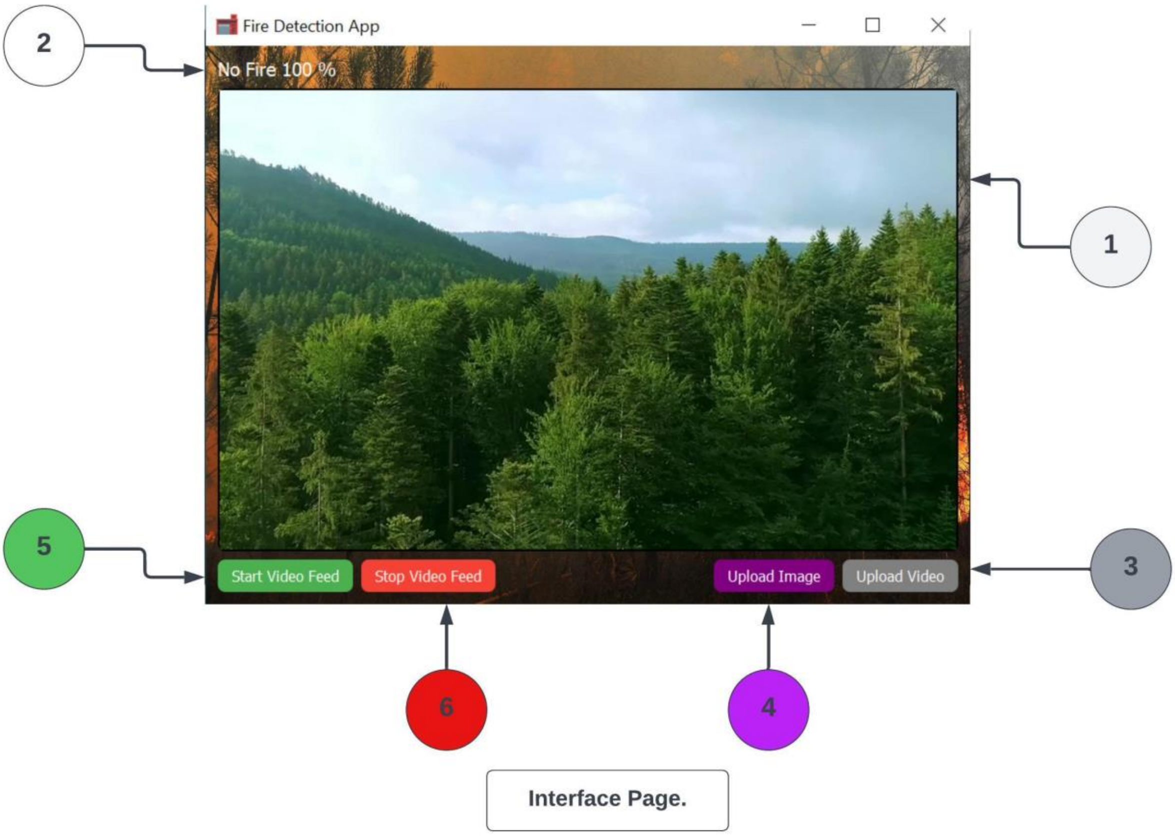

Figure 30: Home Page

Figure (30) illustrates the main interface of our application. The interface includes a central image display box (labeled as 1) where users can view the images that have been classified by the model. Above the image display box, there is a label text (labeled as 2) that dynamically shows the class label assigned to the classified image. This provides users with immediate feedback on the predicted class.

To facilitate user interaction, the application offers three primary functions through dedicated buttons.

- Button (3) allows users to upload a video for classification. By clicking this button, users can select a video file from their local system, and the application will process the video frames one by one, classifying them in real-time. This functionality is particularly useful for analyzing and categorizing video footage related to forest scenes or fire incidents.
- Button (4) enables users to upload an image for classification. Upon clicking this button, users can select an image file from their device. The application will then promptly classify the image and display the result in the image display box (1) along with the corresponding class label in the label text (2). This functionality allows users to obtain instant predictions for individual images.

- For real-time forest classification, users can utilize Button (5). By clicking this button, the application will activate the camera feed and begin real-time classification of the forest scenes captured by the device's camera. The classified images will be displayed in the image display box (1), while the corresponding class labels will be shown in the label text (2). To stop the real-time classification, users can simply click the "Stop Video Feed" button (6).

The system interface of our application provides users with a seamless and user-friendly experience, allowing them to effortlessly upload videos or images for classification and obtain real-time predictions for forest scenes. The intuitive design and clear display of results make it accessible for both experts and non-experts in the field of forest fire prevention and monitoring.

7. Testing

In the Testing section, we discuss the process of evaluating our classification model through various testing scenarios. We present three figures that demonstrate different aspects of the testing process.

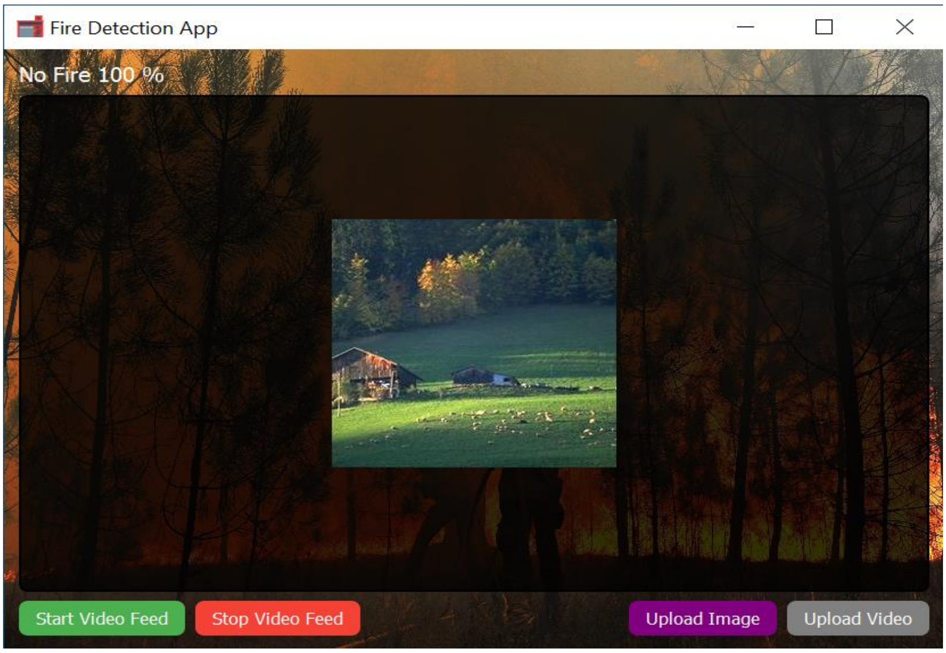

Figure 31: Image Classification Functionality.

Figure 31 showcases the image classification functionality of our application. Users can upload an image by selecting it from their local system using the provided button. Upon uploading the image, the application processes it through the classification model, and the label text dynamically updates to display the predicted class of the image. This figure visually represents the seamless integration of the model into our application, enabling users to obtain accurate and real-time classifications for individual images.

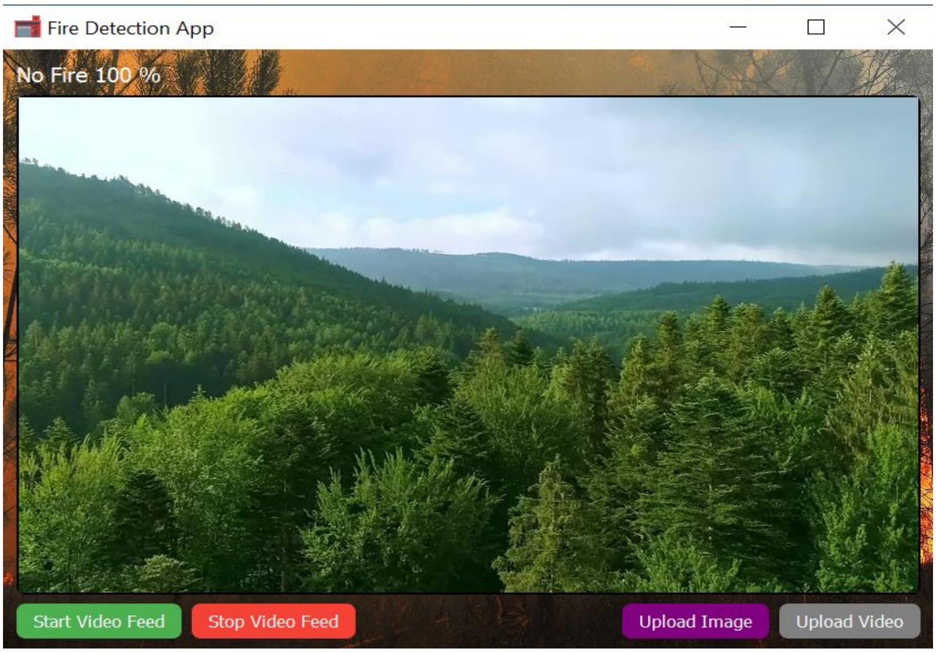

Figure 32: video classification.

Figure 32 focuses on the testing of video classification. Users can upload a video file by clicking the dedicated button. The application then processes each frame of the video sequentially performing real-time classification for each frame. However, to overcome the potential flickering of the class label due to frame-by-frame classification, we introduce a technique where the average prediction and label of 32 consecutive frames are computed. This technique helps provide a more stable and reliable class label for the entire video. Figure 2 illustrates the preprocessing and averaging process, ensuring a smoother and more accurate classification experience for video inputs.

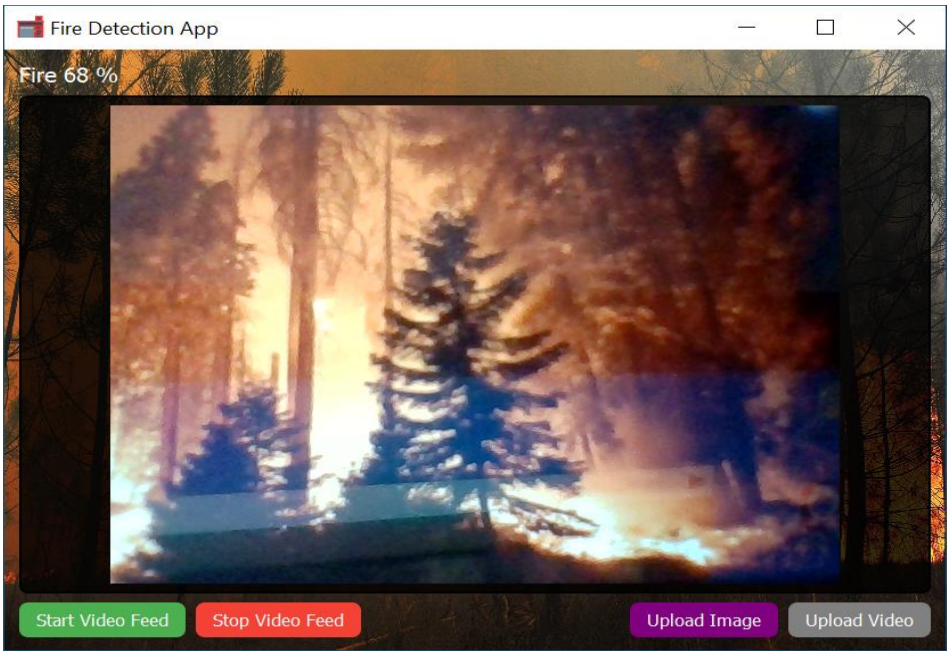

Figure 33: Real-time classification functionality.

Figure 33 demonstrates the real-time classification functionality of our application. By clicking the real-time classification button, users can activate the device's camera feed, allowing the application to continuously process and classify the live video stream. The classified frames are displayed in real-time, along with the corresponding class labels. This figure exemplifies the practical application of our model for monitoring and classifying forest scenes in real-time, providing immediate insights and alerts in critical situations.

Through these testing scenarios depicted in the figures, we validate the robustness and effectiveness of our classification model. The image classification, video classification with frame averaging and real-time classification functionalities demonstrate the versatility and adaptability of our system in different usage scenarios.

8. Conclusion

In conclusion, our research focused on developing an effective model for preventing forest fires. We began by implementing the LF-Model, which achieved a commendable accuracy of 97.21%, along with balanced precision, recall, and F1 score, all at 97%. This demonstrated the reliability of the LF-Model in accurately classifying fire and non-fire instances.

To further enhance our model, we explored different CNN architectures, including VGG16, Inception V3, and Vision Transformers. The VGG16 architecture exhibited impressive performance, with an accuracy of 98.84%, precision and recall both at 99%, and an F1 score of 99%. However, its computational complexity and resource requirements made it less suitable for our application.

The Inception V3 architecture, although less accurate with an accuracy of 82.9%, still demonstrated satisfactory precision, recall, and F1 score, indicating its potential effectiveness in fire detection. However, compared to the LF-Model and VGG16, its performance was relatively lower.

We also explored the use of Vision Transformers, which achieved an accuracy of 75.84%. Although the accuracy was comparatively lower, it showcased an acceptable precision and recall. However, the F1 score indicated room for improvement. The advantage of Vision Transformers lies in their ability to capture global dependencies in the input data, but further refinement is needed to enhance their performance in fire detection.

Additionally, we evaluated the performance of the LF-Model combined with SVM as an ensemble learning technique. This approach achieved an accuracy of 98.62% and demonstrated excellent precision, recall, and F1 score, all above 98%. However, it should be noted that the LF-Model + SVM has the disadvantage of increased computational time, which may limit its real-time implementation, particularly in scenarios where quick decisions are required.

Comparing all the architectures and experiments conducted, the LF-Model showcased strong performance, providing a balance between accuracy and computational efficiency. The VGG16 architecture demonstrated excellent accuracy but was computationally complex, making it less suitable for real-time implementation. Inception V3 showed satisfactory performance but had room for improvement. Vision Transformers exhibited potential but required further optimization.

Overall, our LF-Model, in combination with the SVM ensemble technique, offers a compelling solution for forest fire prevention. Its accuracy, precision, recall, and F1 score highlight its effectiveness in accurately detecting and classifying fire instances. Further research and refinements can be undertaken to improve the performance of Vision Transformers and explore other ensemble learning techniques to enhance the LF-Model further. Our study contributes valuable insights into the field of forest fire prevention, paving the way for future advancements and applications in this critical area.

General Conclusion and Perspectives

In conclusion, we reflect on the journey of our work in preventing forest fires using artificial intelligence which has addressed the challenge of developing an effective solution to this pressing issue. Throughout our research, we have explored various aspects related to forest fires, including their patterns, triggers, effects, and existing prevention approaches.

During the course of our work, we encountered several difficulties that needed to be overcome. One of the challenges was acquiring a comprehensive dataset for training our models. To address this, we utilized an existing dataset rather than creating a new one. Additionally, we focused on designing a lightweight and accurate architecture, the Light-Fire Model (LF-Model), which aimed to achieve high accuracy while maintaining computational efficiency. We faced the challenge of striking a balance between accuracy and efficiency in our model, which required careful parameter optimization and dealing with issues such as class imbalance and overfitting.

Another hurdle we encountered was the development of a user-friendly and intuitive application interface. Designing an interface that seamlessly integrated the classification models while providing easy accessibility to users required a careful balance of aesthetics and functionality. Ensuring smooth and efficient video processing, real-time classification and reliable class label display were additional complexities that needed to be addressed.

Despite these challenges, we successfully developed an application that showcased promising results in preventing forest fires. Our application reasonably classified images and videos providing accurate and real-time predictions of fire occurrences. The integration of different models, such as LF-Model, VGG16, Inception V3, and Vision Transformers, allowed us to compare their performances and identify the most effective approaches.

While our work yielded significant strengths, such as accurate classification, real-time processing, and user-friendly interface, it also had certain limitations. For example, the computational complexity and resource requirements of the VGG16 model made it less suitable for real-time deployment. Additionally, the LF-Model combined with SVM exhibited time-consuming processing, particularly in real-time applications. These limitations highlighted the need for optimization and trade-offs in terms of model selection and deployment considerations.

Looking ahead, there are several directions for future research and improvement. First, incorporating additional data sources, such as satellite imagery, could enhance the comprehensiveness of our system. Exploring ensemble learning techniques and integrating real-time data streams, such as weather updates and remote sensing data, could further enhance the accuracy and proactive nature of our fire prevention measures.

In summary, our thesis has contributed to the field of forest fire prevention using artificial intelligence. By continuously refining and enhancing our approach we can make significant strides in preserving natural ecosystems and mitigating the devastating impact of forest fires.

Citations

1. **Leonard, D. G. Neary and J. M.** Physical Vulnerabilities from Wildfires: Flames, Floods, and Debris Flows. *Natural Resources Management and Biological Sciences.* s.l. : intechopen, 2021, p. 386.

2. **Country, Report Algeria.** *Support to Climate Change Mitigation and Adaptation in the ENPI South Region.* s.l. : ENPI CLIMA-SOUTH, 2013.

3. *Natural Disasters in Algeria can be correlated to Global Warming?.* . **Amara, Sofiane & Nordell, Bo.** 2017.

4. **International Federation of Red Cross.** *Emergency Plan of Action (EPoA), Algeria: Forest Wildfires.* 05 September 2022. p. 14. MDRDZ008.

5. *Fire Activity in Mediterranean Forests (The Algerian Case).* **T. Curt, A. Aini, and S. Dupire.** 2020, Fire, Vol. 3, p. 58.

6. **Mark A. Finney, Sara S. McAllister, Jason M. Forthofer, Torben P. Grumstrup.** *Wildland Fire Behavior : Dynamics, Principles and processes.* Melbourne : CSIRO Publishing, 2021. p. 349.

7. *Understanding the role of slope aspect in shaping the vegetation attributes and soil properties in Montane ecosystems.* **Singh, Shipra.** 2018, Tropical Ecology, Vol. 59, pp. 417-430.

8. **ABLAZE, FORESTS.** Causes and effects of global forest fires. WWW study. [Online] 2017. [Cited: 02 13, 2023.] https://www.wwf.de/fileadmin/fm-wwf/Publikationen-PDF/WWF-Study-Forests-Ablaze.pdf.

9. *"A review of the effects of forest fire on soil properties.".* **Agbeshie, Alex Amerh, et al.** 2022, Journal of Forestry Research 33(5), pp. 1419-1441.

10. *Effects of wildfire on soil nutrients in Mediterranean ecosystems.* **Lucrezia Caon, V. Ramón Vallejo, Coen J. Ritsema, Violette Geissen.** ISSN 0012-8252, 2014, Earth-Science Reviews, Vol. 139, pp. 47-58.

11. **Copernicus Services.** Climate Change Service. [Online] ECMWF. [Cited: 02 13, 2023.] https://climate.copernicus.eu/fire-weather-index.

12. *"Development and structure of the Canadian Forest Fire Weather Index System".* **C.E, Van Wagner.** 1987, Canadian Forestry Service, Ottawa, ON. Forestry Technical Report, pp. 35-37 .

13. *Development of a Keetch and Byram—Based drought index sensitive to forest management in Mediterranean conditions.* **A. Garcia-Prats, Del Campo Antonio, Fernandes J.G. Tarcísio, Molina J. Antonio.** 0168-1923, 2015, Agricultural and Forest Meteorology, Vol. 205, pp. 40-50.

14. *Spatial interpolation of McArthur's Forest Fire Danger Index across Australia: Observational study.* **L.A. Sanabria, X. Qin, J. Li, R.P. Cechet, C. Lucas.** 1364-8152, 2013, Environmental Modelling & Software, Vol. 50, pp. 37-50.

15. **Service, U.S Forest.** U.S. DEPARTMENT OF AGRICULTURE. [Online] [Cited: 02 11, 2023.] https://www.fs.usda.gov/Internet/FSE_DOCUMENTS/stelprdb5339121.pdf.

16. *Deriving forest fire ignition risk with biogeochemical process modelling.* **Eastaugh, Chris & Hasenauer, Hubert.** 55, 2014, Environmental Modelling & Software, pp. 132–142.

17. *The projected 21st century forest-fire risk in Finland under different greenhouse gas scenarios.* **Lehtonen, Ilari & Ruosteenoja, Kimmo & Venäläinen, Ari & Gregow, Hilppa.** 2014, Boreal Environment Research, Vol. 19, pp. 127-139.

18. *A survey on technologies for automatic forest fire monitoring, detection, and fighting using unmanned aerial vehicles and remote sensing techniques.* **C. Yuan, Y. Zhang and Z. Liu.** 7, 2015, Canadian Journal of ForestResearch, Vol. 45, pp. 783–792.

19. *Early fire detection using convolutional neural networks during surveillance for effective disaster management.* **K. Muhammad, J. Ahmad and S. W. Baik.** 2018, Neurocomputing, Vol. 288, pp. 30–42.

20. *Efficient Fire Detection for Uncertain Surveillance Environment.* **Muhammad, Khan and Khan, Salman and Elhoseny, Mohamed and Ahmed, Syed and Baik, Sung.** 2019, EEE Transactions on Industrial Informatics, Vol. PP, pp. 1-1.

21. *Unmanned Aerial Vehicle Assisted Forest Fire Detection Using Deep Convolutional Neural Network.* **Rahman, A. K. Z and Sakif, S and Sikder, Niloy and Masud, Mehedi and Aljuaid, Hanan and Bairagi, Anupam.** 2022, Intelligent Automation and Soft Computing, Vol. 35, pp. 3259-3277.

22. *Real-Time Fire Detection for Video-Surveillance Applications Using a Combination of Experts Based on Color, Shape, and Motion.* **P. Foggia, A. Saggese, and M. Vento.** 2015, IEEE TRANSACTIONS on circuits and systems for video technology, Vol. 25, pp. 1545-1556.

23. *BoWFire: detection of fire in still images by integrating pixel color and texture analysis.* **D. Y. Chino, L. P. Avalhais, J. F. Rodrigues, and A. J. Traina.** 2015 , SIBGRAPI Conference on Graphics, Patterns and Images, pp. 95-102.

24. *Dataset for forest fire detection.* **Hassan, A. Khan and B.** 2020, Mendeley Data, Vol. 1. doi: 10.17632/gjmr63rz2r.1.

25. *Predicting Forest Fire in Algeria Using Data Mining Techniques: Case Study of the Decision Tree Algorithm.* **Abid, Faroudja & Izeboudjen, Nouma.** doi: 10.1007/978-3-030-36674-2_37, 2020, pp. 363-370. 978-3-030-36673-5.

26. *A smart approach for fire prediction under uncertain conditions using machine learning.* **R. Sharma, S. Rani, and I. Memon.** 37–38, 2020, Multimed. Tools Appl, Vol. 79, pp. 28155–28168.

27. *A Data Mining Approach to Predict Forest Fires using Meteorological Data.* **Morais, P. Cortez and A.** 2007, J. Neves, M. F. Santos and J. Machado Eds., New Trends in Artificial Intelligence, Proceedings of the 13th EPIA, pp. 512-523.

28. *The Moderate Resolution Imaging Spectroradiometer (MODIS): land remote sensing for global change research.* **Justice, C.O. and Vermote, E. and Townshend, J.R.G. and Defries, R. and Roy, D.P. and Hall, D.K. and Salomonson, V.V. and Privette, J.L. and Riggs, G. and Strahler, A. and Lucht, W. and Myneni, R.B. and Knyazikhin, Y. and Running, S.W. and Nemani, R.R.** 4, 1998, IEEE Transactions on Geoscience and Remote Sensing, Vol. 36, pp. 1228-1249.

29. *Using MODIS Land Surface Temperature to Evaluate Forest Fire Risk of Northeast China.* **Zhou, G. Guo and M.** 2, 2004, IEEE Geoscience and Remote Sensing Letters, Vol. 1.

30. *Satellite detection of Canadian boreal forest fires: development and application of the algorithm.* **Z. Li, S. Nadon, J. Cihlar.** 16, 2000, " International Journal of Remote Sensing, Vol. 21, pp. 3057-3069.

31. *Requirements for space-based observations in fire management: a report by the Wildland Fire Hazard Team, Committee on Earth Observation Satellites (CEOS) Disaster Management Support Group (DMSG).* **T. J. Lynham, C. W. Dull, and A. Singh.** 2002, n IEEE International Geoscience and Remote Sensing Symposium, Vol. 2, pp. 762-764.

32. *Forest fire detection based on deep learning and multi-source remote sensing data.* **Li, L., Wu, J., Liu, J., Yu, M., & Liu, C.** 4, 2020, Journal of Applied Remote Sensing, Vol. 14, p. 044512.

33. *Forest fire detection based on deep learning with high-resolution satellite images.* **Kang, X., Lin, Q., Wang, Y., & Zhang, Y.** 6, 2020, IEEE Transactions on Geoscience and Remote Sensing, Vol. 58, pp. 4026-4038.

34. *Impact of Artificial Neural Networks Training Algorithms on Accurate Prediction of Property Values.* **Yacim, Joseph & Boshoff, Douw.** 2018, Journal of Real Estate Research, Vol. 40, pp. 375-418.

35. **Dertat, Arden.** Applied Deep Learning - Part 4: Convolutional Neural Networks. [Online] Medium, 11 08, 2017. [Cited: 03 06, 2023.] https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2.

36. *Explainable vision transformer enabled convolutional neural network for plant disease identification: PlantXViT.* **Thakur, Poornima and Khanna, Pritee and Sheorey, Tanuja and Ojha, Aparajita.** 2022.

37. **Welcome to colaboratory. [Online] google.co. [Cited: 04 01, 2023.] https://colab.research.google.com/notebooks/intro.ipynb.**

38. Google Drive help center. [Online] google.co. [Cited: 04 01, 2023.] https://support.google.com/drive/?hl=en#topic=14940.

39. Microsoft. Documentations , Getting Started. [Online] [Cited: 04 01, 2023.] https://code.visualstudio.com/docs/.

40. Foundation, Python Software. What is Python? Executive Summary. [Online] [Cited: 04 04, 2023.] https://www.python.org/doc/essays/blurb/.

41. TensorFlow. Why TensorFlow ? [Online] [Cited: 04 04, 2023.] https://www.tensorflow.org/about?hl=fr.

42. Keras. About Keras. [Online] [Cited: 04 04, 2023.] https://keras.io/about/.

43. What is NumPy ? [Online] numpy org. [Cited: 04 04, 2023.] https://numpy.org/doc/stable/user/whatisnumpy.html.

44. Useres Guide. [Online] [Cited: 04 04, 2023.] https://matplotlib.org/stable/users/index.html#about-matplotlib.

45. OpenCV. About. *Open source computer vision.* [Online] [Cited: 05 11, 2023.] https://opencv.org/about/.

46. *Very deep convolutional networks for large-scale image recognition.* SIMONYAN, Karen et ZISSERMAN, Andrew. 1409.1556, s.l. : arXiv preprint , 2014.

47. Shlens, Jon. Train your own image classifier with Inception in TensorFlow. *google AI Blog.* [Online] Google, 03 09, 2016. [Cited: 05 13, 2023.] https://ai.googleblog.com/2016/03/train-your-own-image-classifier-with.html.

48. *An outdoor fire recognition algorithm for small unbalanced samples.* X. Song, S. Gao, X. Liu and C. Chen. 3, 2021, Alexandria Engineering Journal, Vol. 60, pp. 2801–2809.

49. *Aerial imagery pile burn detection using deeplearning: The FLAME dataset.* A. Shamsoshoara, F. Afghah, A. Razi, L. Zheng, P. Z. Fulé et al. 2020, Computer Networks, Vol. 193, p. 108001.

Résumé

Les incendies de forêt sont un problème mondial qui a de graves répercussions sur les écosystèmes et les communautés. Cette thèse aborde la question urgente de la prévention des incendies de forêt en utilisant l'intelligence artificielle, en particulier les techniques d'apprentissage profond. La recherche explore divers aspects liés aux incendies de forêt, y compris leurs modèles, leurs déclencheurs, leurs effets et les approches de prévention existantes. Les défis rencontrés au cours du processus de recherche, tels que l'acquisition d'un ensemble complet de données et la conception d'un modèle léger mais précis, sont discutés. Le système proposé utilise l'apprentissage automatique supervisé, en mettant l'accent sur le modèle d'incendie léger (modèle LF), pour détecter et classer les incendies de forêt. Les résultats démontrent l'efficacité du système pour classifier avec précision les incendies en temps réel. Les limites et les compromis des modèles sont également discutés, soulignant le besoin d'optimisation et les considérations de déploiement. En affinant et en améliorant continuellement le système, cette thèse contribue à la préservation des écosystèmes naturels et à l'atténuation de l'impact dévastateur des incendies de forêt.

Mots clés : Classification, Apprentissage Profond, Incendies de Forêt, Modèle d'incendie léger, Prévention.

Abstract

Forest fires are a global concern with severe implications for ecosystems and communities. This thesis addresses the pressing issue of forest fire prevention using artificial intelligence, specifically deep learning techniques. The research explores various aspects related to forest fires, including their patterns, triggers, effects, and existing prevention approaches. The challenges encountered during the research process, such as acquiring a comprehensive dataset and designing a lightweight yet accurate model, are discussed. The proposed system utilizes supervised machine learning, with a focus on the Light-Fire Model (LF-Model), to detect and classify forest fires. Results demonstrate the effectiveness of the system in accurately predicting fire occurrences in real-time. The limitations and trade-offs of the models are also discussed, highlighting the need for optimization and deployment considerations. By continuously refining and improving the system, this thesis contributes to the preservation of natural ecosystems and the mitigation of the devastating impact of forest fires.

Keywords: Classification, Deep Learning, Forest Fires, Light-Fire Model, Prevention

ملخّص

تعتبر حرائق الغابات مصدر قلق عالمي ولها آثار وخيمة على النظم البيئية والمجتمعات. تتناول هذه الأطروحة القضية الملحة المتعلقة بالوقاية من حرائق الغابات باستخدام الذكاء الاصطناعي، وخاصة تقنيات التعلم العميق. يستكشف البحث الجوانب المختلفة المتعلقة بحرائق الغابات، بما في ذلك أنماطها ومحفزاتها وآثارها وأساليب الوقاية القائمة. تمت مناقشة التحديات التي تمت مواجهتها أثناء عملية البحث، مثل الحصول على مجموعة بيانات شاملة وتصميم نموذج خفيف الوزن ولكنه دقيق. يستخدم النظام المقترح التعلم الآلي الخاضع للإشراف، للكشف عن حرائق الغابات وتصنيفها. تظهر النتائج فعالية النظام في التنبؤ الدقيق بحوادث الحريق في الوقت الفعلي. تتم أيضًا مناقشة القيود والمقايضات الخاصة بالنماذج، مما يبرز الحاجة إلى اعتبارات التحسين والتطبيق. من خلال صقل النظام وتحسينه باستمرار، تساهم هذه الأطروحة في الحفاظ على النظم البيئية الطبيعية والتخفيف من الآثار المدمرة لحرائق الغابات.

الكلمات المفتاحية : التصنيف، التعلم العميق، حرائق الغابات، نموذج الحرائق الخفيفة، الوقاية.