PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA

MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH

UNIVERSITY ECHAHID HAMMA LAKHDAR - EL OUED

FACULTY OF EXACT SCIENCES

Computer Science Department

End of Study Memory

Presented for The Diploma of

ACADEMIC MASTER

Domain: Mathematics and Computer Science

Specialty: Internet of Things and Cyber Security

Theme

Design and Development of an Integrated Smart System for Detecting and Fighting Forest Fires

Presented by:

Mohammed Elhadj Labadi

Amine Djaboub

Under Supervision of The Professor:

Mohammed Charaf Eddine Meftah

Academic Year: 2025-2026

Table of Contents

Table of Contents 2
List of Figures 5
Thanks 7
Dadicate 9
Abstract 11
Introduction General 13

1 Background on Forest Fires and Detection Fundamentals 14
1.1 Introduction 15
1.2 Concepts of Fire 15
1.2.1 Definition of Fire 15
1.2.2 Fire Traigale: 15
1.2.3 Types of Fire: 16
1.3 Forest Fires 16
1.3.1 Definition of Forest Fires: 16
1.3.2 Types of Forest Fires: 17
1.3.3 Characteristics of Forest Fires: 17
1.3.4 Causes of Forest Fires: 18
1.4 Traditional Fire Detection: 18
1.4.1 Definition of Fire Detection Systems: 18
1.4.2 Limitations of Traditional Systems 19
1.5 Fire Extinguishing Systems 19
1.5.1 Definition of Fire Extinguishing System: 19
1.5.2 Types of Extinguishing Agents: 19
1.6 Internet of Things (IoT) 20
1.6.1 Definition 20

1.6.2 Components 20
1.6.3 The Role of Internet of Things in Fire Detection 21
1.7 Artificial Intelligence (AI) 21
1.7.1 Definition 21
1.7.2 Components 21
1.7.3 The Role of Artificial Intelligence in Fire Detection 22
1.8 Conclusion 22

2 Related Works and Comparative Analysis 23
2.1 Introduction 24
2.2 Existing Fire Detection Systems 24
2.2.1 Silvanet (Dryad) 24
2.2.2 AlertWildfire 25
2.2.3 Smart Fire Detection and Alert System Drone: AI-IoT Integration 25
2.2.4 Design And Production of an Intelligent Forest Fire Detection System 26
2.2.5 Wireless Sensor Based Monitoring System for Forest Tree Poaching and Fire Using IoT 26
2.3 A Comparison of Current Forest Fire Control Systems 27
2.4 Conclusion 29

3 Design of The Proposed System 30
3.1 Introduction 31
3.2 Proposed Smart Firefighting System 31
3.3 System Overview 31
3.4 System Architecture 32
3.5 System Layers 33
3.5.1 Sensing Layer 34
3.5.2 Communication Layer 35
3.5.3 Processing Layer 36
3.5.4 Implementation Layer 36
3.5.5 User Interface Layer 37

3.6 Communication Protocols 37
3.6.1 Wi-Fi 37
3.6.2 MQTT Protocol 37
3.6.3 LoRa 38
3.7 Hardware Design 38
3.7.1 HZH XF100 Firefighting Drone 38
3.7.2 Water-based Fire Extinguishing Bomb 39
3.7.3 Ultra Long-Range Multi-Sensor PTZ Camera System 39
3.7.4 Bosch BME688 Gas Sensor Developer Kit 40
3.7.5 u-blox ZED-F9P GPS Module 41
3.7.6 SGP41 42
3.7.7 Raspberry Pi 5MP Camera OV5647 42
3.7.8 MQ-2 Smoke Detection Module 43
3.7.9 ESP32-CAM (OV2640) 44
3.7.10 Lithium Battery 3.7V (502525) 45
3.7.11 ESP32 46
3.7.12 Raspberry Pi 5 46
3.7.13 DHT22 Sensor 47
3.8 Software Tools 48
3.8.1 Arduino IDE 48
3.8.2 Roboflow 49
3.8.3 Visual Studio Code 49
3.8.4 Google Colab 50
3.9 UML Modeling 50
3.9.1 Use Case Diagram 50
3.9.2 Activity Diagrams 52
3.9.3 Sequence Diagram 54
3.10 Conclusion 55

4 Implementation and Results **56**

4.1 Introduction 57
4.2 Real System Implementation 57
4.2.1 Sensing Layer 57

4.2.2 Communication Layer 58
4.2.3 Processing Layer 58
4.2.4 Actuation Layer 59
4.2.5 Simulation Workflow 60
4.3 Practical System Implementation 60
4.3.1 Sensing Layer 60
4.3.2 Communication Layer 62
4.3.3 Processing Layer 62
4.3.4 Actuation Layer 67
4.3.5 User Interface Layer 69
4.4 System Testing 71
4.4.1 Sensing Layer Testing 71
4.4.2 Communication Layer Testing 72
4.4.3 Processing Layer Testing 72
4.4.4 Full System Testing 72
4.5 Results and Discussion 73
4.5.1 Results 74
4.5.2 Discussion 74
4.6 Conclusion 75
General Conclusion 76

List of Figures

1.1 Elements responsible for the fire 16
1.2 Types of Forest Fires 17

2.1 Silvanet (Dryad) 24
2.2 AlertWildFire 25
2.3 Smart Fire Detection and Alert System Drone: AI-IoT Integration . . 26
2.4 Design of an Intelligent Forest Fire Detection System 26
2.5 Wireless Sensor Based Monitoring System For Forest Tree Poaching and Fire Using Iot 27

3.1 The system is in monitoring mode 32
3.2 The system is in the process of being suppressed 33
3.3 System Layers 34
3.4 Sensing Layer 35
3.5 Communication Layer 35
3.6 Processing Layer 36
3.7 Action Layer 37
3.8 HZH XF100 Firefighting Drone 38
3.9 Water-based Fire Extinguishing Bomb 39
3.10 Thermal Camera 40
3.11 Bosch BME688 Gas Sensor Developer Kit 41
3.12 u-blox ZED-F9P GPS Module 41
3.13 SGP41 42
3.14 Raspberry Pi 5MP Camera OV5647 43
3.15 MQ-2 Smoke Sensor 44
3.16 ESP32-CAM(OV2640) 45

3.17 Lithium Battery 3.7V (502525) 45
3.18 ESP32 46
3.19 Raspberry Pi 5 47
3.20 DHT22 Sensor 48
3.21 Arduino IDE 48
3.22 Roboflow 49
3.23 Visual Studio Code 49
3.24 google colab 50
3.25 Use Case Diagram Of The System 52
3.26 Activity diagram project 54
3.27 Sequence Diagram Project 55

4.1 High-altitude Thermal Camera Tower Overlooking a Dense Forest Canopy 58
4.2 An Industrial Hexacopter Drone Autonomously Releasing a Payload over a Localized Fire Spot 60
4.3 The Camera Tower 61
4.4 Node Sensors 62
4.5 Data Processing 64
4.6 Sorted Data 64
4.7 Model Training 65
4.8 Results Model Training 66
4.9 Performance curves and evaluation metrics for the training model . . 66
4.10 Testing YOLOv8 Model 67
4.11 Drone 68
4.12 Website 70
4.13 Image of GCS UI 71
4.14 Testing Node 72

Thanks

First of all, we thank Allah Almighty who gave us the strength and patience to accomplish this humble work.

It is a pleasure to extend our sincere thanks to our supervisor,

Meftah Mohammed Charaf Eddine

for his understanding, guidance, assistance, and valuable advice that greatly helped us throughout the realization of this project.

We also thank the jury members who kindly reviewed and evaluated this thesis.

Finally, we would like to express our gratitude to everyone who contributed, directly or indirectly, to the completion of this work.

Dadicate

To our dear parents, there are no words to express what we owe them for their benevolence, affection and support Treasures of kindness, generosity and tenderness, in testimony of our deep loves and our great gratitude
"May Allah keep you".

We would like to dedicate this humble work as proof of respect, gratitude and gratitude to our dear parents, our families and friends to all those we for their encouragement, support, patience and prayers.

Abstract

Annually, forest fires cause huge environmental, economic and social losses all over the world, especially in areas with large forest areas, where early detection and rapid response are essential to limit the spread of fires and reduce their impact. Traditional fire detection systems often suffer from limitations such as delayed response, limited coverage, and false alarms. In this note, we propose an intelligent fire detection and extinguishing system based on the integration of Internet of Things (IoT) and artificial intelligence (AI) technologies. The system collects environmental data through distributed sensors and a camera suspended in towers over the forests. This information is then sent to the processing platform for analysis and detection of potential fire events with higher accuracy. Then the automatic fire extinguishing process takes place. This proposal aims to enhance forest fire monitoring and reduce potential damage caused by forest fires.

Keywords:forest fires, Internet of Things (IoT), artificial intelligence (AI), smart monitoring systems, forest fire management.

Summary

Chaque année, les incendies de forêt provoquent d'énormes pertes environnementales, économiques et sociales partout dans le monde, en particulier dans les zones abritant de vastes zones forestières, où une détection précoce et une réponse rapide sont essentielles pour limiter la propagation des incendies et réduire leur impact. Les systèmes de détection d'incendie traditionnels souffrent souvent de limitations telles qu'une réponse tardive, une couverture limitée et de fausses alarmes. Dans cette note, nous proposons un système intelligent de détection et d'extinction d'incendie basé sur l'intégration des technologies de l'Internet des objets (IoT) et de l'intelligence artificielle (IA). Le système collecte des données environnementales grâce à des capteurs distribués et une caméra suspendue dans des tours au-dessus des forêts. Ces informations sont ensuite envoyées à la plateforme de traitement pour analyser et détecter les incendies potentiels avec une plus grande précision. Ensuite, le processus d'extinction automatique d'incendie a lieu. Cette proposition vise à améliorer la surveillance des incendies de forêt et à réduire les dommages potentiels causés par les incendies de forêt.

Mots clés: incendies de forêt, Internet des objets (IoT), intelligence artificielle (IA), systèmes de surveillance intelligents, gestion des incendies de forêt.

ملخص

تتسبب حرائق الغابات سنويا في خسائر بيئية واقتصادية وإجتماعية فادحة في جميع أنحاء العالم، خاصة في المناطق ذات المساحات الحرجية الكبيرة, حيث يعد الاكتشاف المبكر والاستجابة السريعة أمرًا ضروريًا للحد من انتشار الحرائق وتقليل تأثيرها. وما ذلك غالبًا ما تعاني أنظمة الكشف عن الحرائق التقليدية من قيود مثل الاستجابة المتأخرة والتغطية المحدودة والإنذارات الكاذبة. حيث نقترح في هذه المذكرة نظامًا ذكيًا للكشف عن الحرائق وإطفاءها يعتمد على دمج تقنيات إنترنت الأشياء (IoT) والذكاء الاصطناعي .(AI) يقوم النظام بجمع البيانات البيئية من خلال أجهزة الاستشعار الموزعة والكاميرا المعلقة على الأبراج في الغابات ثم يتم إرسال هذه المعلومات إلى منصة المعالجة لتحليلها وإكتشاف أحداث الحرائق المحتملة بدقة أعلى ثم يتم عملية الإطفاء التلقائية للحريق, حيث يهدف هذا المقترح إلى تعزيز مراقبة حرائق الغابات وتقليل الأضرار المحتملة الناجمة عن حرائق الغابات.

كلمات مفتاحية:حرائق الغابات، إنترنت الأشياء ،(IoT) الذكاء الاصطناعي ،(AI) أنظمة المراقبة الذكية، إدارة حرائق الغابات.

General Introduction

Forest fires are among the most serious environmental disasters that threaten ecosystems around the world, as they annually destroy large areas of forests and cause serious damage to the environment, economy and society. Forest fires are characterized by their great ability to spread rapidly, especially in dry climate conditions and the presence of winds, which makes them more difficult to control and increases the amount of losses resulting from them.

Early detection of fires is one of the most important factors that help limit their spread and reduce the damage resulting from them. However, traditional detection systems often rely on limited methods based on discrete sensors or human observation, which makes them insufficient in some cases, especially in large and complex environments such as forests.

With the rapid development in the field of technology, smart solutions based on the Internet of Things and artificial intelligence have emerged. Through these technologies, it is possible to create smart monitoring systems capable of continuously collecting and analyzing environmental data, which helps to detect fires early and take appropriate measures quickly.

We will propose an intelligent system for detecting and extinguishing forest fires based on the Internet of Things and artificial intelligence technologies, with the aim of improving detection accuracy and reducing losses resulting from forest fires. We will present this note in four chapters:

The first chapter presents basic concepts related to fires and modern technologies.

The second chapter was devoted to studying current forest fire detection systems.

The third chapter focuses on the design of the proposed system.

Finally, the fourth chapter, where we will present an implementation of the proposed system



Chapter 1

Background on Forest Fires and Detection Fundamentals

1.1 Introduction

Fire accidents are very dangerous in many areas and often cause serious damage and human lives lost. Limitations of traditional fire detection techniques in early detection and response. With the rapid development of modern technologies, especially the Internet of Things (IoT) and artificial intelligence (AI), new smart to make better fire detection and extinguishing systems because they let us detect and make early choices.
In this chapter we will describe the basics of fire detection and extinguishing systems, Internet of Things and artificial intelligence technologies, and their role in the development of smart systems.

1.2 Concepts of Fire

1.2.1 Definition of Fire

Fire is the rapid oxidation of a substance in an exothermic chemical combustion process, releasing heat, light, and the products of various chemical reactions[1].

1.2.2 Fire Traigale:

The fire triangle is a scientific model that shows that the occurrence of a fire requires the presence of three basic elements, as shown in the figure(Figure 1.1):

- **Fuel:** Fuel is any combustible material and is characterized by the moisture, size, shape, quantity, and order in which it is spread over the landscape.
- **Heat:** The heat source is responsible for the initial ignition of the fire, and heat is necessary to sustain the fire and allow it to spread.
- **Oxygen:** The air around us contains approximately 21% oxygen. In order for combustion to occur, the air must contain at least 16% oxygen[2].

It prevents ignition in the absence of one of these three elements.

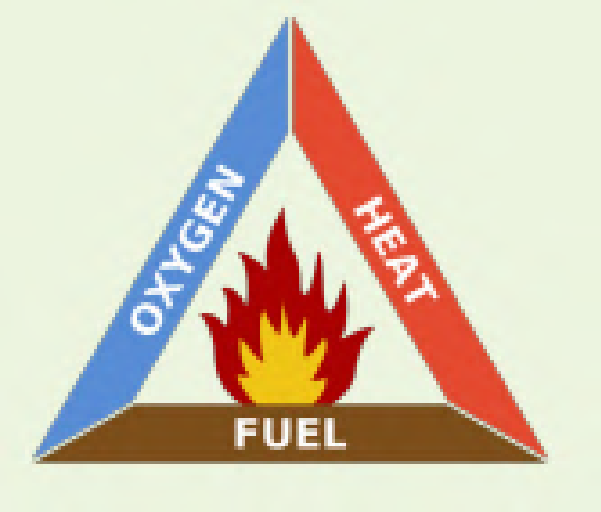


Figure 1.1: Elements responsible for the fire

1.2.3 Types of Fire:

Fires are classified into classes (A, B, C, D, and K) by the National Fire Protection Association (NFPA). This classification depends on the nature of the material used in the combustion process and the nature of the reaction during the fire, as different materials require different methods for fighting fires and different extinguishing agents, which we mention:

Class A: Fires of ordinary combustible materials, such as wood, cloth, paper, rubber and many plastics.

Class B: Fires of flammable liquids and flammable gases such as petroleum grease and oils.

Class C: Fires involving energized electrical equipment.

Class D: Fires of combustible metals, such as magnesium and potassium.

Class K: Fires of cooking oils and fats in the kitchen. [3].

1.3 Forest Fires

1.3.1 Definition of Forest Fires:

Forest fires are an uncontrolled burning process that occurs in natural environments such as forests or vegetated areas, where plants are used as fuel for the fire and the fire spreads rapidly under the influence of environmental factors such as wind, temperature,

terrain or human activities.

1.3.2 Types of Forest Fires:

Once started, a fire can take different forms, each of which is conditioned by the characteristics of the vegetation and the climatic conditions, We mention it:

Crown Fires: Crown fires burn trees up their entire length to the top. These are the most intense and dangerous wildland fires.

Surface Fires: Surface fires burn only surface litter and duff. These are the easiest fires to put out and cause the least damage to the forest.

Ground Fires: Ground fires occur in deep accumulations of humus, peat and similar dead vegetation that become dry enough to burn. These fires move very slowly but can become difficult to fully put out[4].



Figure 1.2: Types of Forest Fires

1.3.3 Characteristics of Forest Fires:

Propagation Speed:

According to the General Directorate of Forests, 811 fires have broken out in 32 states in Algeria since the beginning of 2025. These fires have destroyed 40,001 hectares of forests, shrublands and grasslands[5].

Large Spaces:

Algerian forest land is estimated at 4,115,908 hectares, of which 2,413,090 (58%) hectares are bush or maki. Research has shown that forest fires are the main factor

explaining the degradation of forest lands in Algeria at an annual rate of 45,000 to 50,000 hectares. From 1985 to 2010 in 40 states of northern Algeria, 42,555 forest fires burned a total of 910,640 hectares. On average, 1,636 fires burn 35,024 hectares of forest land annually[6].

Difficulty of Extinguishing:

Forest fires spread fastest in hard-to-reach places and in the right conditions. Large parts of Algeria mark these boxes, and there is usually very dry and flammable undergrowth. With limited access to the forest and steep slopes, effective initial detection and intervention by firefighters is very difficult[7].

1.3.4 Causes of Forest Fires:

Natural Causes:

Extreme weather conditions, including drought, strong winds and high temperatures, contribute to the spread of fires[5].

Human Causes:

Forest areas in Algeria are considered more vulnerable to multiple human pressures, and among the most common causes of fire outbreaks are criminal acts, private and agricultural forest works, irresponsible behavior, throwing waste, re-ignition of fires, and so on. This creates suitable conditions for fires to spread [7].

1.4 Traditional Fire Detection:

1.4.1 Definition of Fire Detection Systems:

They are systems that detect fires using standard sensors (smoke, heat, flame, and gas) by monitoring the physical indicators of the fire and raising an alarm when preset threshold values are exceeded. These sensors operate separately and in a limited manner, especially in forests, based on the detection method on which these sensors rely, as they issue alarms when one of the materials emanating from the fire is detected without confirming it, by processing images or previous data.

1.4.2 Limitations of Traditional Systems

Traditional fire detection systems rely on sensors such as smoke, heat and flame detectors that operate under pre-set thresholds to detect physical changes associated with a fire. Despite their widespread use in residential and industrial buildings, these systems suffer from several limitations. The most important of these limitations is the high rate of false alarms resulting from environmental factors such as dust, steam, or cooking activities, in addition to the limited sensitivity in detecting fires in their early stages, especially in large or complex environments. Also, these systems often rely on a fixed network of sensors, which leads to limited coverage and the presence of areas where the fire may not be detected quickly. Furthermore, most traditional systems lack intelligent data analysis and real-time decision-making capabilities, which can result in delayed response and increased fire spread risk.

1.5 Fire Extinguishing Systems

1.5.1 Definition of Fire Extinguishing System:

Fire suppression systems always consist of at least one component designed to suppress a fire using an external substance. Many of these systems also include fire detection systems, in addition to alarm mechanisms to alert workers to the presence of a fire and prompt officials to take additional suppression measures. Detecting a fire often automatically activates the suppression element of the system; However, some fire suppression systems require manual activation.

1.5.2 Types of Extinguishing Agents:

Water: Extinguishes the fire by cooling and may be used in the form of a jet or finely pulverized. Water in a jet should only be used in Class A fires. Pulverised water can be use in Class A fires and in Class B fires when these involve so-called “heavy” combustible liquids, such as fuel oil, gas oil, etc.

CO2: This is an inert gas and hence is used as an element to suffocate fires. It is effective for fires produced by inflammable liquids and in electrical fires because it is not a conductor and it does not leave any residue.

Chemical Powder: This is generally a chemical compound made up of sodium bicarbonate and a water-resistant agent. It acts by suffocating and paralysing the chain reaction. Two types of dry chemical powder are chiefly employed nowadays: normal and polyvalent (or anti-re-ignition) dry chemical powder. The latter cools the fuel much more and so is more effective than normal dry chemical powder for Class A fires. There also exist a series of special dry chemical formulations for Class D fuels. Normal dry chemical powder is effective against Class B and C fires and against fires in the presence of electric voltage. It can be used in Class A fires, but water will then have to be used to avoid re-ignition.

Chemical Foam: This is formed by mixing an acid solution in another, basic solution. When closely mixed together, the two solutions react producing carbon dioxide (CO2), with the subsequent increase in pressure that discharges the extinguishing foam. This type of foam presents the drawbacks of attacking metals, being a conductor of electricity and dissolving alcohols and so is not presently used[8].

1.6 Internet of Things (IoT)

1.6.1 Definition

The Internet of Things (IoT) refers to a network of physical devices, vehicles, appliances, and other physical objects embedded with sensors and software and connected to a network, allowing them to collect and share data. It is used in all fields[9].

1.6.2 Components

The Internet Protocol (IP) and the Transmission Control Protocol (TCP) are considered the basis of the Internet of Things. The Internet of Things consists of several elements that work in an integrated manner. These elements include sensors that collect information from the required environment and wireless communication devices such as Wi-Fi and Bluetooth to transfer the data to the data processing unit and make decisions based on the extracted results[10].

1.6.3 The Role of Internet of Things in Fire Detection

The role of the internet of things in forest fires is through the development of fire detection and extinguishing systems, which is done by using smart sensors to monitor environmental indicators such as gases and temperature, where the data collected from the devices is sent to the processing units in order to analyze it and make appropriate decisions in real time, which allows for reducing material and human losses.

1.7 Artificial Intelligence (AI)

1.7.1 Definition

Artificial intelligence (AI), the ability of a digital computer or computer-controlled robot to perform tasks commonly associated with intelligent beings. The term is frequently applied to the project of developing systems endowed with the intellectual processes characteristic of humans, such as the ability to reason, discover meaning, generalize, or learn from past experience. Since their development in the 1940s, digital computers have been programmed to carry out very complex tasks—such as discovering proofs for mathematical theorems or playing chess—with great proficiency. Despite continuing advances in computer processing speed and memory capacity, there are as yet no programs that can match full human flexibility over wider domains or in tasks requiring much everyday knowledge. On the other hand, some programs have attained the performance levels of human experts and professionals in executing certain specific tasks, so that artificial intelligence in this limited sense is found in applications as diverse as medical diagnosis, computer search engines, voice or handwriting recognition, and chatbots[11].

1.7.2 Components

Artificial intelligence is built on basic components that help it achieve intelligent behavior of systems, including machine learning, which makes the system learn from data individually, and data processing, which organizes and analyzes information, as well as decision-making algorithms, which help choose the most appropriate solution according to the available data. Databases and processing units play a major role in supporting all these operations and improving performance[11].

1.7.3 The Role of Artificial Intelligence in Fire Detection

Artificial intelligence is crucial in fighting forest fires, as it helps in detecting them early. These systems rely on analyzing a huge amount of information, such as temperatures and smoke in the air, images taken from satellites, and video clips from cameras. It uses computer programs capable of detecting patterns that indicate the outbreak of a fire, which allows it to see smoke or flames even in difficult circumstances. These systems are constantly improving in detecting fires thanks to learning from previous experiences, which makes them more accurate and reduces false alarms. Artificial intelligence can also work with aircraft to monitor vast areas and reach places that are difficult to reach, which makes it possible to issue warnings to residents or make quick decisions to extinguish the fire, thus limiting its spread and reducing environmental and human damage [11].

1.8 Conclusion

In this chapter we discussed the basic concepts related to fires and traditional detection and extinguishing systems, with a focus on forest fires and their associated challenges. The most important limitations of traditional systems that hinder their effectiveness in large environments were also presented.
In the next chapter, we will conduct an analytical study of some systems and present our proposed system.



Chapter 2

Related Works and Comparative Analysis

2.1 Introduction

Forest fires pose major threats to natural and human resources, and despite the great technological development in the field of fire detection, many current detection and extinguishing systems still suffer from limited performance, both in terms of accuracy and speed of response. In this chapter, we will study some of the systems that were previously designed in the field of fire detection and extinguishing, mentioning the strengths and weaknesses of each system. We will make a comparison between these systems in terms of features and limitations and extract the most important limitations that these systems face.

2.2 Existing Fire Detection Systems

2.2.1 Silvanet (Dryad)

Dryad's Silvanet is a forest fire monitoring system that detects wildfires in their early stages. This device was completed by the German company Drayd on 2021. The Silvanet suite consists of four components: a forest fire sensor, a network gateway, a boundary gateway, and a cloud computing platform. Together, these components provide a unique LoRaWAN (Local Fire Network) fire detection device technology, comprised of solar-powered sensors and gateways distributed throughout the forest. Integrated artificial intelligence also helps provide accurate wildfire alerts[12].



Figure 2.1: Silvanet (Dryad)

2.2.2 AlertWildfire

AlertWildfire is a monitoring system for wildfires that utilizes cameras. Completed at the University of Nevada in 2013 in the United States of America. It links high-resolution cameras with artificial intelligence to detect smoke and monitor fires in real time. this system is designed for extensive monitoring and provides visual evidence of fire incidents[13].

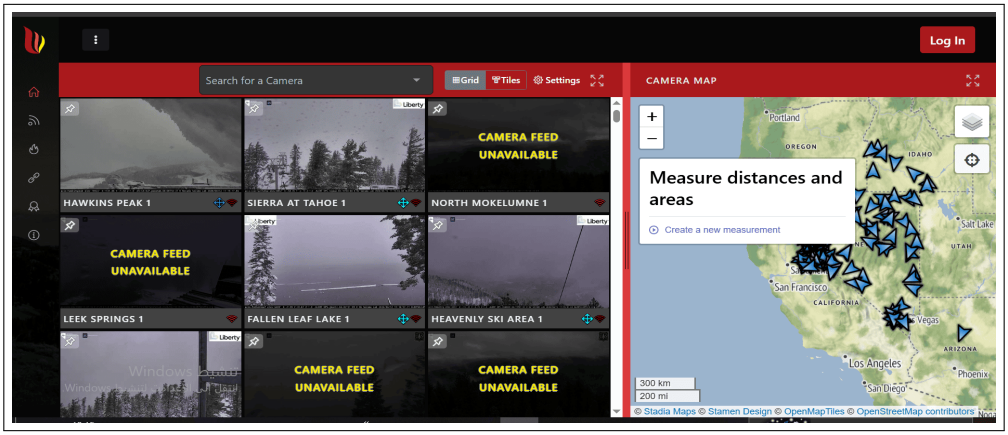


Figure 2.2: AlertWildFire

2.2.3 Smart Fire Detection and Alert System Drone: AI-IoT Integration

This project from "Université Saad Dahlab Blida" by students of KHENDOUKI on 2023-2024 is a smart system for detecting forest fires using drone technologies, artificial intelligence, and the Internet of Things. Drones monitor forests using a mounted camera, then the images are analyzed in real time and alerts are issued when a fire occurs[14].



Figure 2.3: Smart Fire Detection and Alert System Drone: AI-IoT Integration

2.2.4 Design And Production of an Intelligent Forest Fire Detection System

This project, "Ain Temichent University Center," by students SOUSSI and MIRNES on 2018-2019, presents a traditional system for detecting forest fires using the Internet of Things. This system relies on detecting fires only on sensors, after which it determines the presence of a fire or not[15].

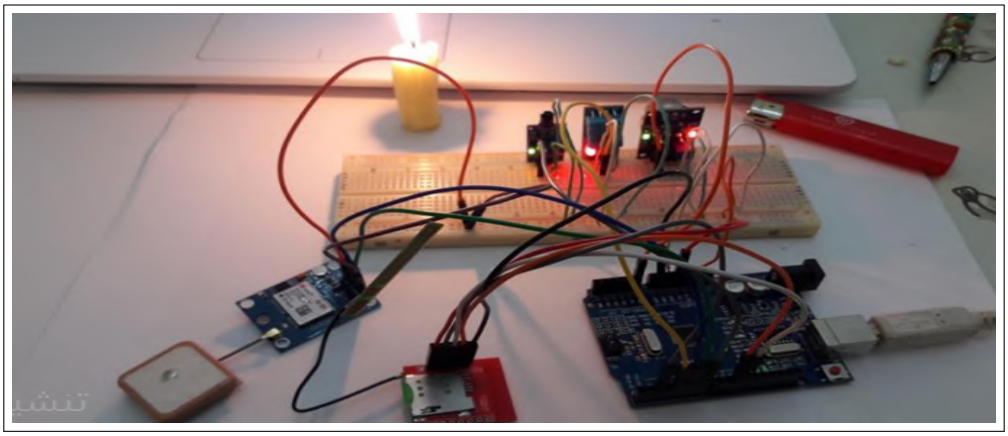


Figure 2.4: Design of an Intelligent Forest Fire Detection System

2.2.5 Wireless Sensor Based Monitoring System for Forest Tree Poaching and Fire Using IoT

This project is a research paper written by Ebenezer Abishek, c.Arulstephen, dated August 2018, mainly aims to monitor trees using Micro Mechanical System (MEMS) technology and employing vibration sensors, sound sensor and fire sensors. If a fire is

detected, the information is sent through IoT, and the forest ranger can view it and then an alert message is sent, including the location, to detect and avoid tree poaching and fires in the forest[16].

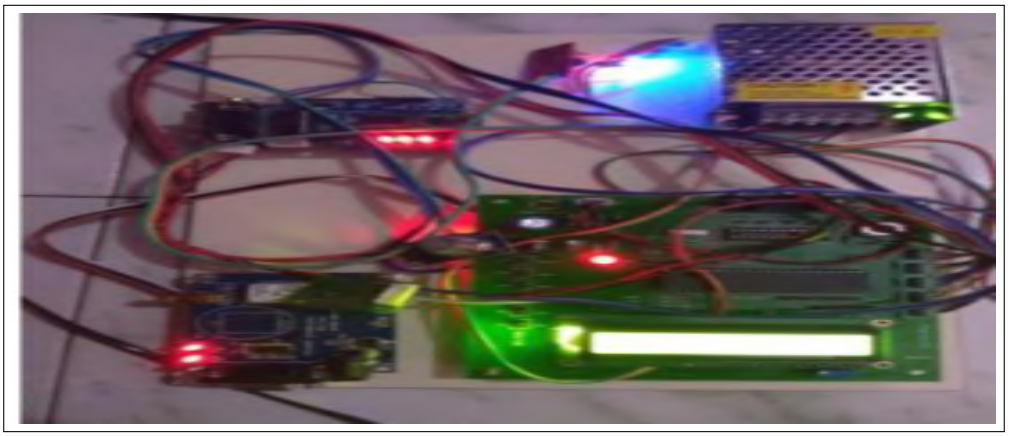


Figure 2.5: Wireless Sensor Based Monitoring System For Forest Tree Poaching and Fire Using Iot

2.3 A Comparison of Current Forest Fire Control Systems

System	Advantages	Limitations
Silvanet	• Very early fire detection before visible smoke. • High accuracy in detecting fires.	• High deployment cost due to large number of sensors. • Possible false alarms due to changes in environmental conditions. • Maintenance challenges in harsh forest environments.
AlertWildfire	• Wide-area monitoring using camera networks. • Real-time visual detection supported by AI .	• Delayed fire detection until smoke becomes visible . • Strong dependence on weather and lighting conditions.

Table 2.1 – continued from previous page

		• Limited effectiveness in remote or poorly covered regions.
Smart Fire Detection and Alert System Drone: AI-IoT Integration	• Real-time fire detection using AI. • Integration of drones for monitoring.	• Strong dependence on camera visibility conditions. • Limited drone battery life and flight duration. • Does not provide automatic fire extinguishing.
Wireless Sensor Based Monitoring System for Forest Tree Poaching and Fire Using Iot	• Real-time monitoring using wireless sensor networks. • Low-cost IoT-based architecture.	• Relies mainly on traditional sensors without AI analysis. • No image or video-based fire verification. • High dependence on sensor deployment density. • Absence of automatic fire suppression mechanisms.
Forest Fire Protection and Risk Monitoring System	• Provides wildfire risk monitoring. • Helps improve fire prevention strategies.	• Relies mainly on traditional monitoring methods. • Absence of real-time image analysis. • No automatic fire-extinguishing mechanism. • Limited monitoring coverage in remote areas

Table 2.1: A Comparison of Wildfire Control Systems.

On the basis of this comparison, we can say that modern firefighting systems, though advanced, still lack a solution that is both accurate and fast at the same time. Some are oriented toward early detection but suffer from poor coverage in forests due to environmental conditions, whereas the others rely on visual detection that is still vulnerable to smoke, weather, and so on. And all of this variation in capability really comes down to every system only solving for one part of the problem, without understanding it whole. So these shortcomings were the main impetus to present the system proposed in this memory, with the intent of addressing them effectively and practically.

2.4 Conclusion

In conclusion, this chapter examined some systems for detecting and responding to forest fires, then compared them to conclude the most important limitations they have in common.
In the next chapter, we will describe the architecture and design of our proposed scheme.



Chapter 3

Design of The Proposed System

3.1 Introduction

In this chapter, we will present the steps for designing our smart system for fighting forest fires, where we present the needs and main diagrams of the project, and this is done using the UML language.

3.2 Proposed Smart Firefighting System

After studying the problems faced by most systems in detecting existing forests, we propose an intelligent system that combines the Internet of Things and artificial intelligence to detect forest fires early and respond to them quickly. The system consists of a camera that continuously monitors the forests and a group of distributed environmental sensors that monitor environmental conditions continuously and in real time. The information collected is processed through artificial intelligence to detect real fire risks and natural environmental changes and reduce the number of false warnings. The system contains a control panel for direct monitoring and also provides fire suppression service using drones. The goal is to improve response time and reduce the amount of fire losses.

3.3 System Overview

In this work, an integrated smart system is proposed for detecting and extinguishing forest fires. Our system relies on four basic components: the sensing layer, the communication layer, the processing layer, and the implementation layer. The sensing layer contains devices for measuring temperature and smoke, which are spread in areas that the camera cannot reach (slopes, plateaus, etc.) in the forest. The sensing layer also contains a thermal camera that takes pictures and videos from the field center in the forest to follow events in real time.

Then WiFi and MQTT technology are used to transfer data from the sensors to the central processing unit, and LoRa is also used to cover large parts of the forest to enable devices to communicate with each other. After that, the processing layer processes the data, images, and videos using AI to detect the presence of fires.

The execution layer sends a command to the drone to go to the fire site and drop fire-extinguishing balls in case there is a fire. All of this is done automatically without human

intervention, which in turn is only there to monitor and inspect.
The system also consists of an interface to display data and forest status in real time.

3.4 System Architecture

The system architecture of the proposed solution provides a picture of how all the different parts work together to detect wildfires in real time and respond quickly.
Nodes deployed in areas that the camera cannot reach in the forest (plateaus, slopes, etc.) collect data and sense (smoke, humidity, temperature) and send it to the server to process the data. In turn, the thermal camera also performs a complete survey of the forest field and sends videos and images to the server for processing via the smart model trained to detect fire to determine whether there is a fire or not.

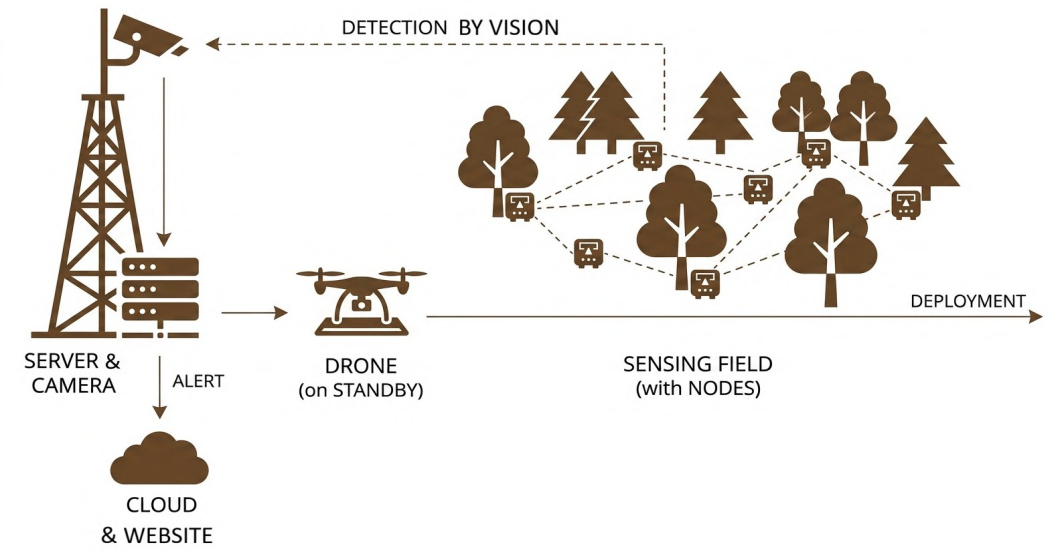


Figure 3.1: The system is in monitoring mode

If a rise in temperature and smoke data sent from a node is detected and it exceeds the permissible threshold, the server directs the camera towards the location of the node from which the rise in values was monitored to verify. If the camera cannot see anything and the values are still rising, the server gives a command to the drone to head towards the

fire site to verify, carrying fire-extinguishing balls with it. If a fire is detected by the drone actually at the intended location, it throws fire-extinguishing balls over the fire, then waits for a period of time to re-verify whether. Whether the fire has extinguished or not, if it does not extinguish, fire another ball at the fire again until it extinguishes.
If the fire is detected via the camera, we repeat the same previous extinguishing procedures.

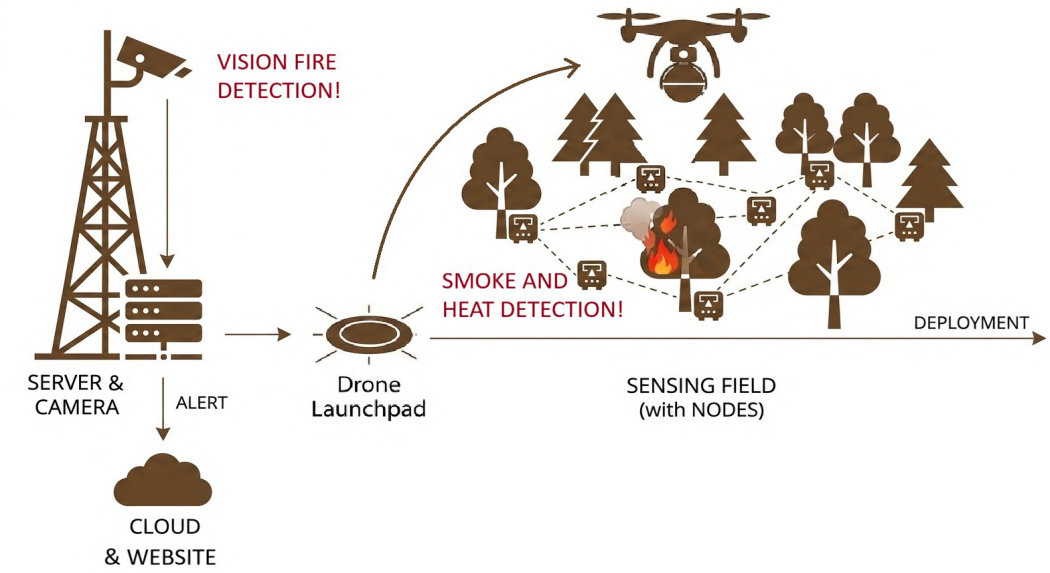


Figure 3.2: The system is in the process of being suppressed

Knowing that all these procedures, information, and data will be displayed on the website to keep officials and the station responsible for forests updated with all the updates, whether a fire alarm or the possibility of seeing the field through the tower camera or monitoring data and temperature and smoke values.
This method ensures that data moves quickly, responses happen rapidly, and the overall system for detecting and managing wildfires in forest areas is more reliable.
The proposed solution employs wildfire detection and management to ensure everything functions well. Wildfire detection is a part of this system, and it aids in rapid response.

3.5 System Layers

The proposed system for wildfire detection and response is organized into layers to make it work better. Each layer of the wildfire detection and response system has a job

to do. This makes the wildfire detection and response system work well. The main parts of the wildfire detection and response system are the sensing layer, the communication layer, the processing layer, the actuation layer and the user interface layer of the wildfire detection and response system.

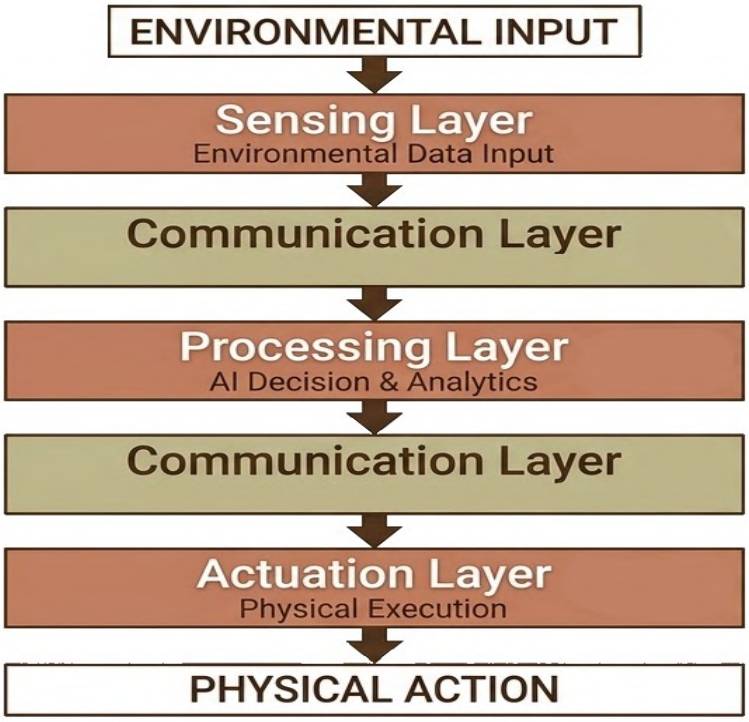


Figure 3.3: System Layers

3.5.1 Sensing Layer

The sensing layer is the base of the wildfire detection and response systems. Data is collected from the surroundings through nodes placed in difficult areas of the forest, as well as through a camera that scans the forest almost completely. These sensors watch things like temperature and smoke levels all the time. The main goal of the sensing layer of the wildfire detection and response system is to find any changes in the environment that might mean a wildfire is starting.

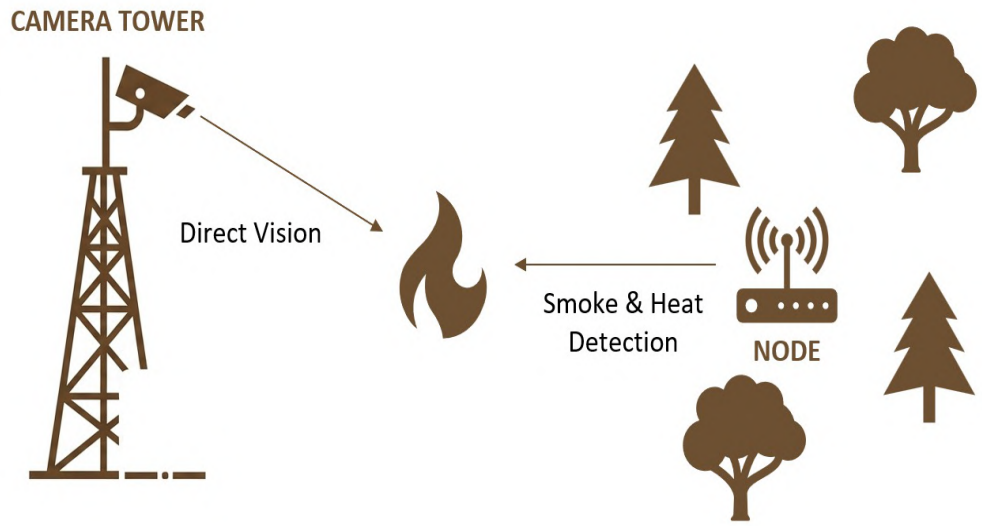


Figure 3.4: Sensing Layer

3.5.2 Communication Layer

The communication layer of the wildfire detection and response system is in charge of sending the data from the sensing layer to the processing unit. Such as WiFi and MQTT to ensure that data arrives in real time, as well as using LoRa to completely cover the forest and make devices connected to each other even over long distances. This layer of the fighting forest fires system is very important because it makes sure the sensors and the main system are always connected.

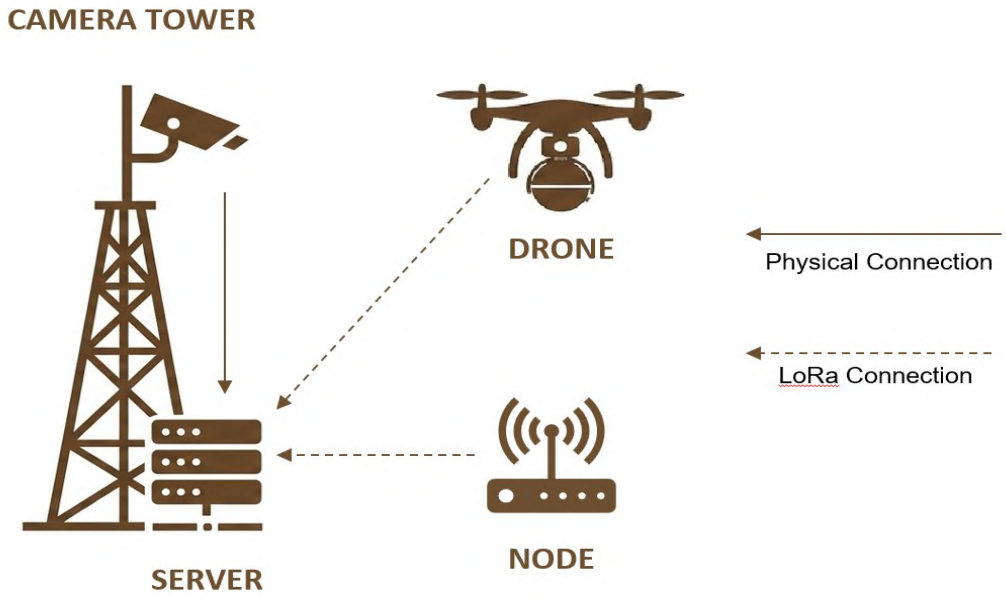


Figure 3.5: Communication Layer

3.5.3 Processing Layer

The processing layer of the wildfire detection and response system is the important part. It is where we look at the data we collected and use algorithms to find patterns and detect potential fires. This layer of the wildfire detection and response system helps us make decisions and avoids false alarms.

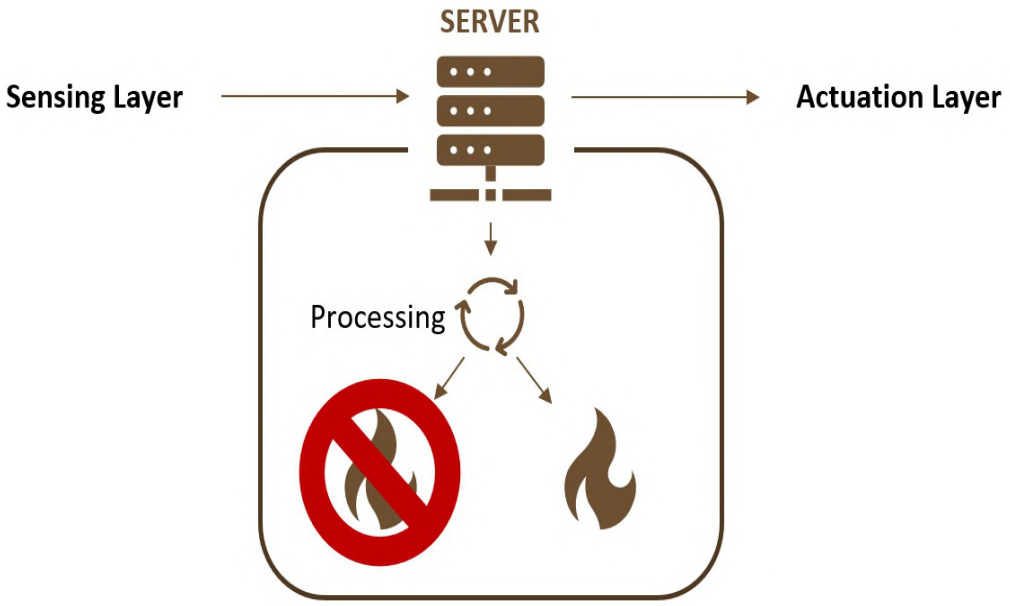


Figure 3.6: Processing Layer

3.5.4 Implementation Layer

The implementation layer of the wildfire detection and response system is what happens when a fire is detected. In this system we send a drone to the area where the fire is to try and put it out. This happens automatically. Helps us respond quickly to fires.

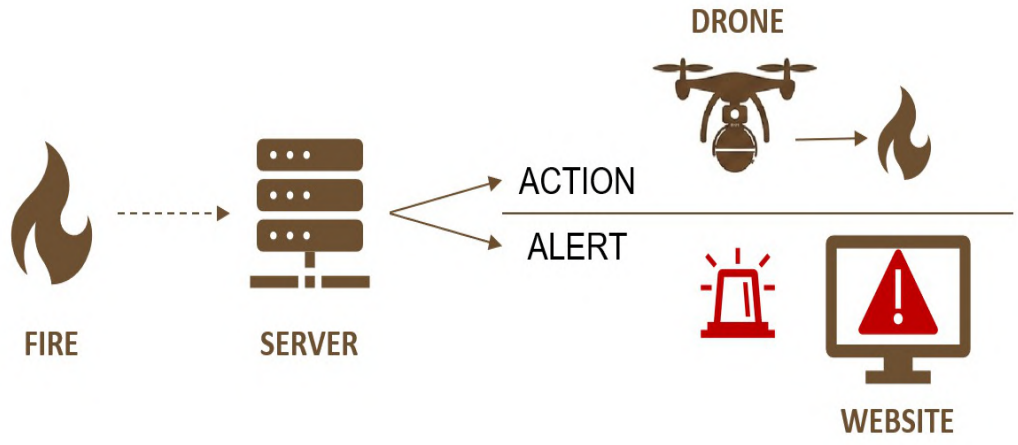


Figure 3.7: Action Layer

3.5.5 User Interface Layer

The user interface layer of the wildfire detection and response system is where people can watch what is happening in time and interact with the system. It has a screen that shows us what the sensors are saying, how the system is working, and if there are any alerts. This layer of the wildfire detection and response system helps us understand what is going on and track fires and how the system is working.

3.6 Communication Protocols

3.6.1 Wi-Fi

Based on the IEEE 802.11 standard, Wi-Fi is a wireless communication technology that allows devices to access local networks for data communication within the range of a few meters to a few kilometers. Its high speed data transfer and the possibility to easy integrate it was the reason of great use in IoT systems[17].

3.6.2 MQTT Protocol

MQTT (Message Queuing Telemetry Transport) is a lightweight messaging protocol for IoT and machine to machine communications. It is based on a publish/subscribe architecture and is optimized to low-bandwidth and low-power devices[18].

3.6.3 LoRa

The LoRaWAN specification is a Low Power, Wide Area (LPWA) networking specification designed to wirelessly connect battery operated 'things' to the internet in regional, national or global networks, and targets key Internet of Things (IoT) requirements such as bi-directional communication, end-to-end security, mobility and localization services[19].

3.7 Hardware Design

3.7.1 HZH XF100 Firefighting Drone

The HZH XF100 Firefighting Drone is versatile for use in mountains, grasslands, forests, and specific urban areas, equipped with a release mechanism, targeting dispenser, laser rangefinder, and four 25 kg fire-extinguishing bombs. This system is adept at rapid and effective firefighting across various terrains.

Properties

- Max Payload: 100 kg.
- Empty Weight: 52 kg.
- Max. Take-Off Weight: 190 KG[20].



Figure 3.8: HZH XF100 Firefighting Drone

3.7.2 Water-based Fire Extinguishing Bomb

This water bomber is purpose-built for the aerial fire suppression role. It is released in high-risk areas and triggered by heat. It is suitable for the fire fighting of any terrain.

Properties

- Designed for aerial firefighting missions.
- Compatible with drones and UAV systems.
- Suitable for forests and difficult terrains[21]



Figure 3.9: Water-based Fire Extinguishing Bomb

3.7.3 Ultra Long-Range Multi-Sensor PTZ Camera System

The Vega is a fully customizable multi-sensor EO-IR PTZ camera boasting our longest range visible day/night zoom camera and our most advanced cooled or uncooled thermal infrared cameras. It can be configured with our optional ZLID™ long-range NIR illumination, LRF high-precision rangefinder, gyro stabilization and more.

Properties

- 15.4–2075mm HD IR-Corrected Zoom Lens
- 27°–0.2° Horizontal Field of View gives a 135X Zoom Range

- Long-range monitoring capability up to 55 km
- 550X Zoom Ratio with 110° Wide-Angle Spotter Camera[22]



Figure 3.10: Thermal Camera

3.7.4 Bosch BME688 Gas Sensor Developer Kit

Bosch Sensortec's BME688 Development Kit allows for testing and developing use cases based on temperature, barometric pressure, humidity and gas sensing. The BME688 can detect gases by measuring their unique electronic fingerprint and therefore distinguish different gas compositions. This enables a broad spectrum of new applications.

Properties

- Gas detection capability.
- Temperature, humidity, and pressure measurement.
- High measurement accuracy[23]

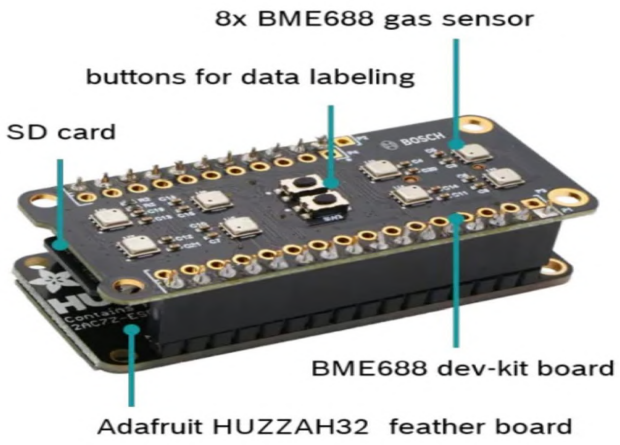


Figure 3.11: Bosch BME688 Gas Sensor Developer Kit

3.7.5 u-blox ZED-F9P GPS Module

Multi-band GNSS receiver delivers centimeter level accuracy in seconds.

Properties

- Concurrent reception of all constellations.
- Multi-band RTK with fast convergence times and reliable performance.
- Centimeter-level accuracy in a small and energy-efficient module[24].



Figure 3.12: u-blox ZED-F9P GPS Module

3.7.6 SGP41

The SGP41 is Sensirion's new VOC and NOx sensor designed as digital smart switch and regulation unit for air treatment devices such as air purifiers.

Properties

- Sensor output: Digital 16-bit raw signal.
- Processed output: 0-500 VOC Index.
- Measurement range: 0 - 1000 ppm[25].



Figure 3.13: SGP41

Due to the limited budget, lack of time, and the difficulty of providing all the necessary equipment and real conditions during the project completion period, a simulation environment was relied upon to initially test some parts of the system and verify its performance.

3.7.7 Raspberry Pi 5MP Camera OV5647

The Raspberry Pi Camera OV5647 is a 5MP camera module widely used for Embedded vision system and IOT applications and it is proven to work with Raspberry Pi. It allows image and video capture for monitoring and detection.

Properties

- Resolution: 5MP (2592×1944)
- Video: 1080p @ 30fps
- Sensor: OmniVision OV5647
- Pixel size: 1.4µm
- Interface: CSI (Camera Serial Interface)[26]



Figure 3.14: Raspberry Pi 5MP Camera OV5647

3.7.8 MQ-2 Smoke Detection Module

The MQ-2 is a metal oxide semiconductor (MOS) gas sensor and one of the most widely used sensors in the MQ series. We know that MOS sensors are known as Chemiresistors because there is a sensing material in them that works based on the change in its resistance when it is exposed to the reactive gas.

Properties

- Detects: Smoke, LPG, methane, hydrogen
- Concentration Range: 300-10000 ppm

- Heater Voltage: 5.0 V ± 0.2 V DC/AC
- Preheat Time: Over 48 hours[27]



Figure 3.15: MQ-2 Smoke Sensor

3.7.9 ESP32-CAM (OV2640)

The ESP32-CAM is a cost effective module which has a camera and microcontroller with inbuilt WiFi and Bluetooth capabilities.

Properties

- CPU: 240 MHz
- Flash: 4MB
- Camera: OV2640 (2MP)
- GPIO pins for sensors
- Wi-Fi + Bluetooth[28]



Figure 3.16: ESP32-CAM(OV2640)

3.7.10 Lithium Battery 3.7V (502525)

The 502525 battery is a rechargeable lithium-polymer (Li-Po) battery used to provide stable power for portable electronic devices and embedded systems. These batteries are characterized by high energy density, light weight, and small size, making them suitable for applications such as Internet of Things (IoT) systems and microelectronic circuits.

Properties

- Nominal voltage: 3.7V
- Rechargeable (Li-ion technology)
- High energy density
- Lightweight and compact size[29]



Figure 3.17: Lithium Battery 3.7V (502525)

3.7.11 ESP32

ESP32 is a low-power microcontroller developed by Espressif Systems. It integrates Wi-Fi and Bluetooth communication capabilities, making it widely used in IoT and embedded systems applications.

Properties

- Dual-core processor up to 240 MHz
- Integrated Wi-Fi and Bluetooth
- Low power consumption
- High Level of Integration[30]



Figure 3.18: ESP32

3.7.12 Raspberry Pi 5

The Raspberry Pi 5 is a high-performance single-board computer for embedded systems, AI, and IoT applications. 8GB RAM provides you with more processing power suitable for real-time data analysis and image processing tasks.

Properties

- 8GB LPDDR4X RAM
- Quad-core ARM Cortex-A76 CPU

- Dual-band Wi-Fi and Bluetooth

- High processing performance for AI applications[31]



Figure 3.19: Raspberry Pi 5

3.7.13 DHT22 Sensor

The DHT22 sensor is really good at measuring temperature and humidity. People often use the DHT22 sensor, in Internet of Things systems to keep an eye on the environment. The DHT22 sensor is very accurate when it comes to measuring temperature and humidity.

Properties

- Temperature range: −40°C to 80°C

- Humidity range: 0–100% RH

- Digital signal output[32]



Figure 3.20: DHT22 Sensor

3.8 Software Tools

3.8.1 Arduino IDE

Arduino IDE is a simple and effective open-source development environment (write, compile, and upload) for application code/program to Arduino microcontroller board and pretty much to other board types including ESP32. It has a simple, friendly interface which is good for embedded and IOT development[33].



Figure 3.21: Arduino IDE

3.8.2 Roboflow

Roboflow is a computer vision and AI project management platform that lives in the cloud. It has features for image annotation, dataset management, data augmentation, and model training, making image-based AI development easier[34].

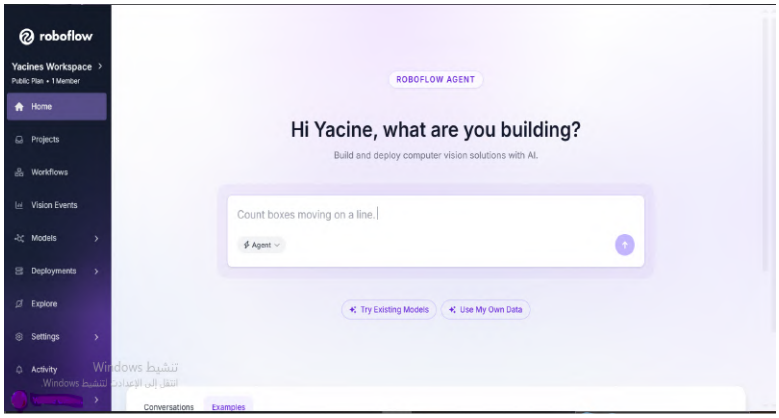


Figure 3.22: Roboflow

3.8.3 Visual Studio Code

Visual Studio Code (VS Code) is a free source code editor developed by Microsoft for Windows, Linux and macOS. It has support for many programming languages and it includes debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git[35].

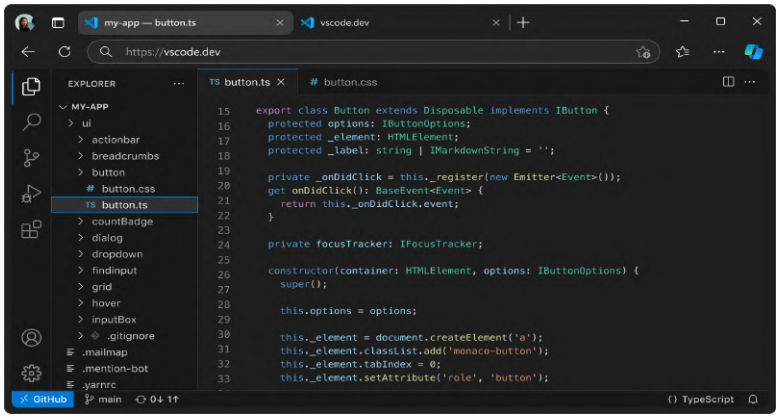


Figure 3.23: Visual Studio Code

3.8.4 Google Colab

Google Colab is a cloud-based platform that allows you to write and run Python programs directly from your browser. It is widely used in data analysis and machine learning, without requiring local installation[36].



Figure 3.24: google colab

3.9 UML Modeling

Unified Modeling Language (UML) is a general purpose language for visualizing, specifying, constructing, and documenting the artifacts of a system. It offers a number of diagrams that assist system designers to conceptualize system elements and their interaction as well as business processes and procedures in an understandable and organized manner. UML is adopted in various domains and at various system levels[37].

3.9.1 Use Case Diagram

Definition

A use case diagram is a graphical depiction of a user's possible interactions with a system. A use case diagram shows various use cases and different types of users the system has and will often be accompanied by other types of diagrams as well. The use cases are represented by either circles or ellipses. The actors are often shown as stick figures[38].

Use Case Of The System

The use case diagram shows the interaction among the main components of the proposed wildfire detection and extinguishing system. It shows the cooperation of sensing nodes, cameras, servers, drones, and website for real-time monitoring, fire detection and automatic response.

Node:

The node is responsible for sensing the environment parameters like smoke, temperature, and humidity. The data is collected and transmitted to the server through the communication network.

Camera:

The camera is used for monitoring of forests in real-time by taking the images and videos from the area under surveillance. Captured media is send to a server.

Server:

The role of the server is to receive data from the camera and nodes, analyze this data, give commands to the drone, and update the location when a fire is detected in the forest.

Drone:

The drone is responsible for extinguishing fires. As soon as it receives orders from the server, it goes to the fire site and drops fire-extinguishing balls.

Website:

The website allows users to watch in real time. Displays environmental data, node locations, alerts, forest status and video view from camera.

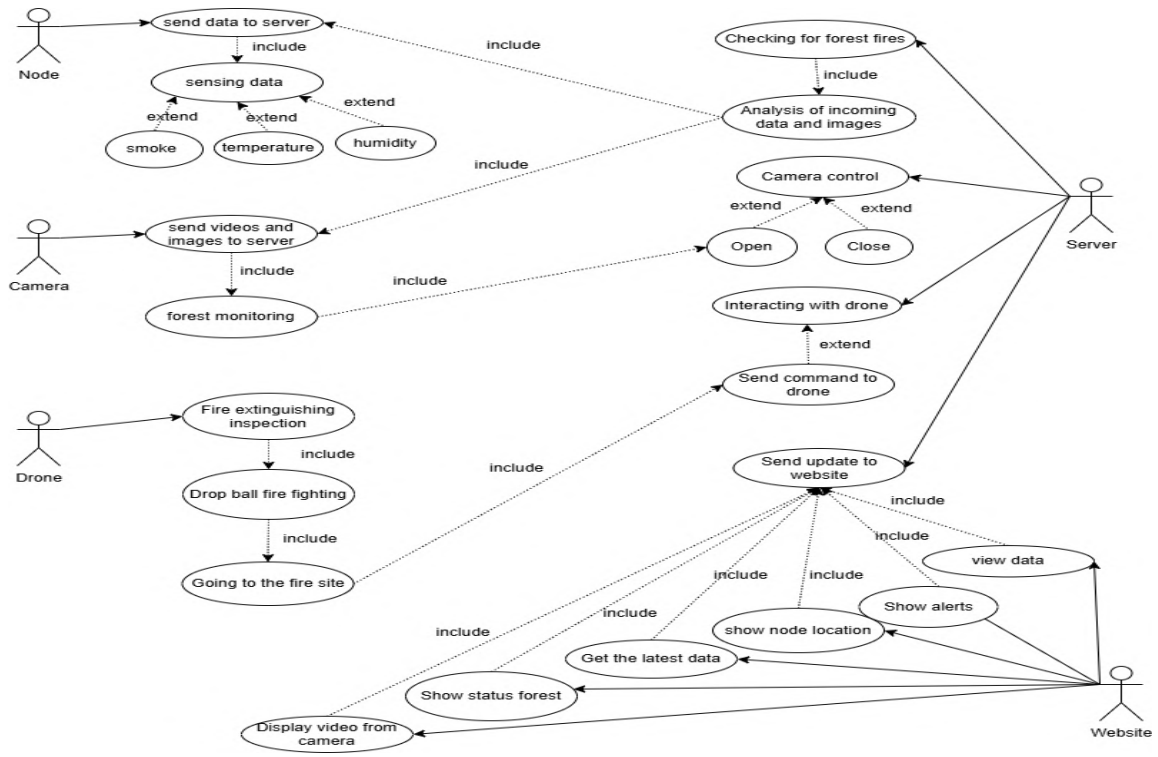


Figure 3.25: Use Case Diagram Of The System

3.9.2 Activity Diagrams

Definition

Activity diagrams are an essential part of the Unified Modeling Language (UML) that help visualize workflows, processes, or activities within a system. They depict how different actions are connected and how a system moves from one state to another[39].

Activity Diagram of Project

1. **Data Collection and Monitoring**

- Node senses the data and sends it to Server.
- Operate the camera at all times to monitor the forest.
- Send the video to the server.

2. **Website Updates**

- Sending data from server to website.
- View node locations on site.

- View the status of forests on site.
- Display alarms.
- Display video on site when the camera is turned on.
- Display data on site (temperature, humidity, smoke, etc.).

3. Data Analysis

- Analyzing data, videos, and photos sent to the server.

In the event that a fire is detected by crossing the threshold, the following is done:

- Sending an alert notification.
- Direct the camera to the location of the fire.
- Captures video and sends it to the server.
- Video analysis using the trained model to detect fire.

If the fire is not detected by the camera (due to plateaus, slopes...) and the values sent from the nodes are still increasing, we do the following:

- Change the state of the forest at the location (there is fire).
- Send a command to the drone to go to the target area.
- The drone heads to the fire site.
- When you verify that the drone is above the target, receive a fire-extinguishing ball.
- Waiting for the drone for 15 seconds in the same place.
- After waiting, the drone makes sure that the fire is extinguished.
- If it does not subside, fire another extinguishing ball again until the fire subsides.
- Then the drone returns to its station.
- Information is sent from the drone to the server that the fire has been extinguished.

- The state of the forest is changed (no fire).

If the fire is detected from the tower camera:

- We repeat the same previous extinguishing procedures.

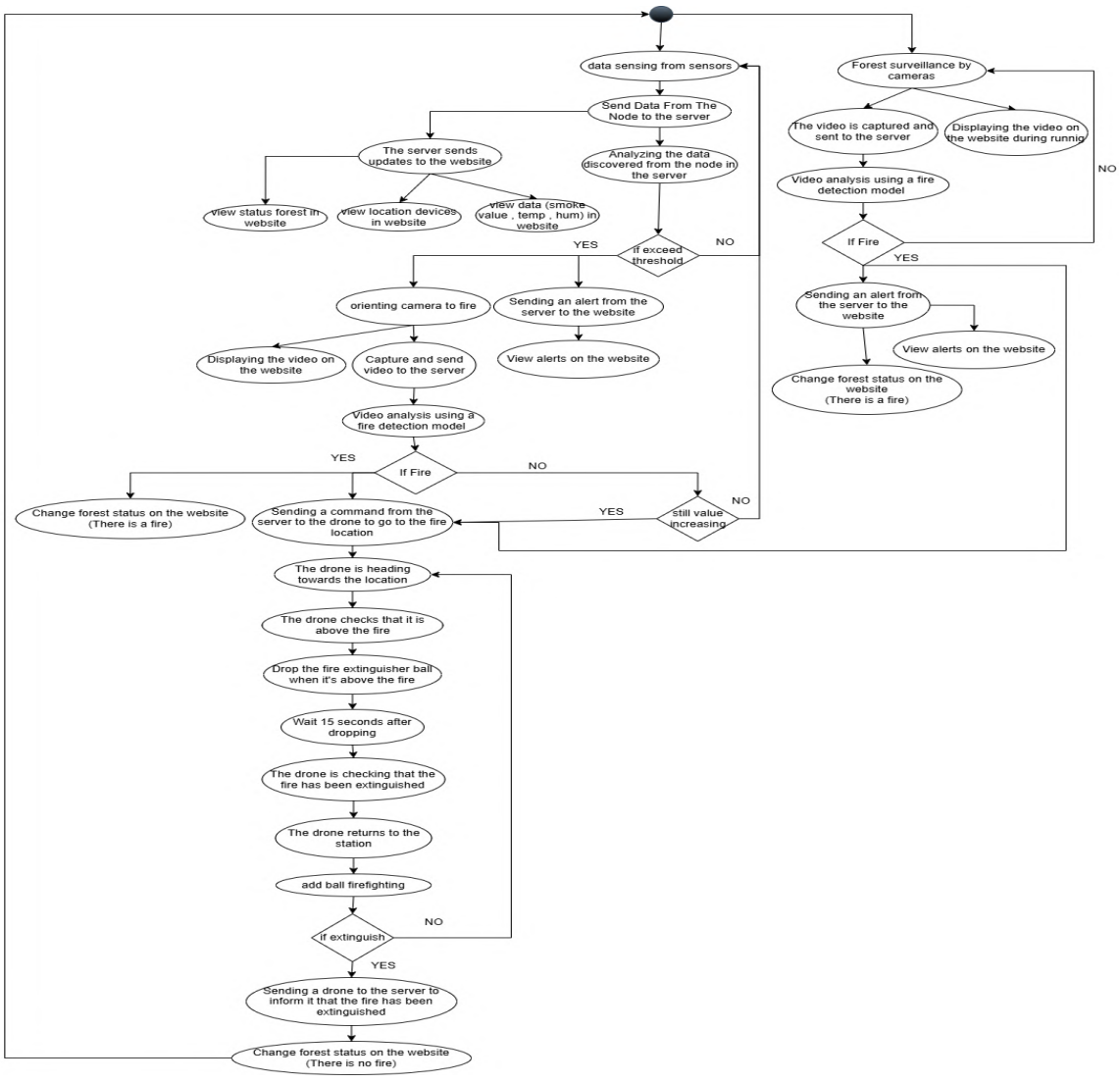


Figure 3.26: Activity diagram project

3.9.3 Sequence Diagram

Definition

It is a kind of interactive diagram that shows the flow of operations of properties, objects with other entities, and objects[40].

Sequence Diagram of Project

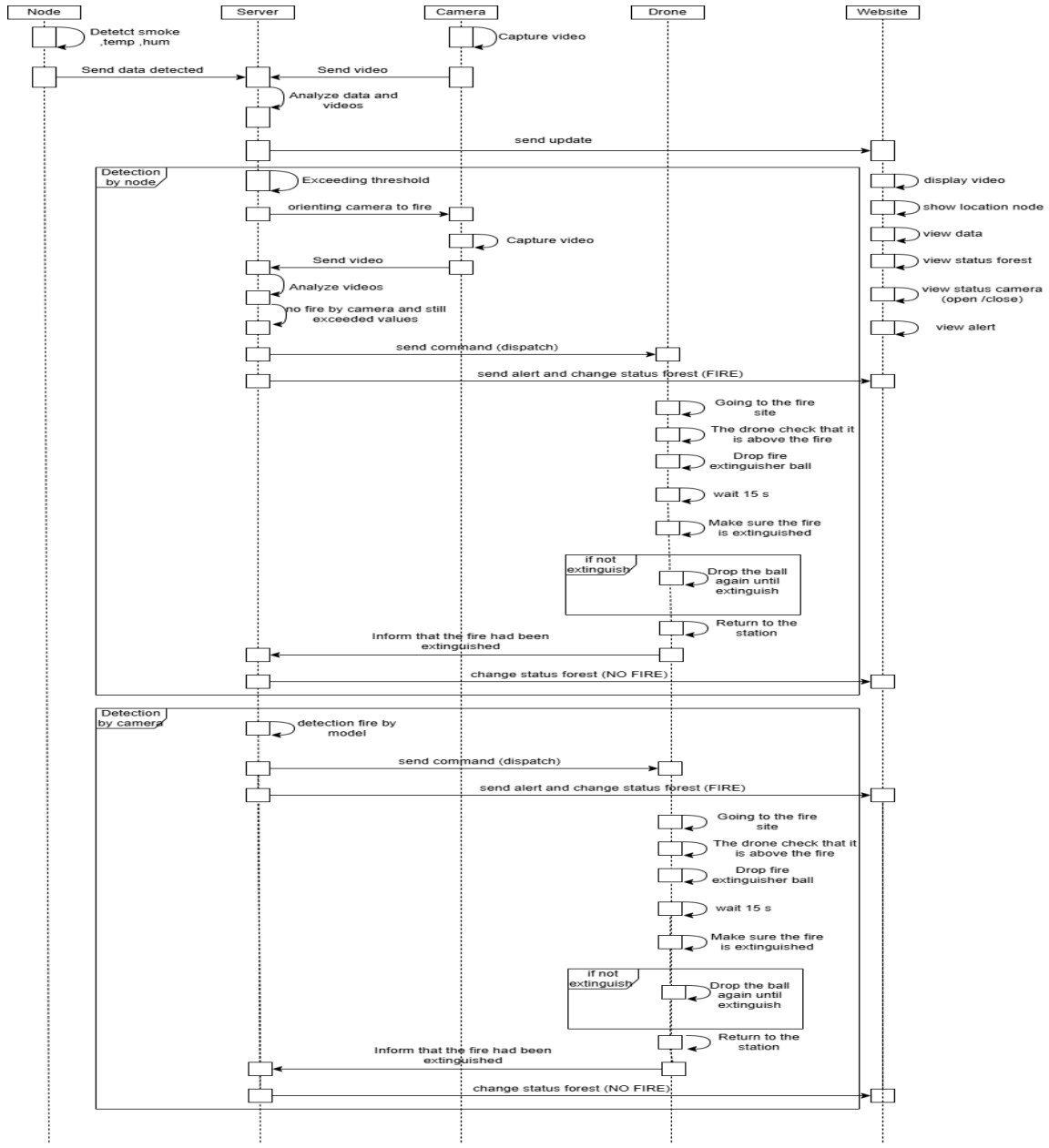


Figure 3.27: Sequence Diagram Project

3.10 Conclusion

In the chapter, the architecture, layered approach, and key subsystems of a novel early-wildfire detection and suppression system was described. The behavior and interactions of the system were also depicted with UML diagrams. This design gives a nice clean solid base to work from in the implementation, which we will cover in the next chapter.



Chapter 4

Implementation and Results

4.1 Introduction

The comprehensive completion of the project and its actual practical features requires more time than the time available for the master's thesis project, so we decided to simulate the intended system in the real world only, while focusing time and resources on making a small prototype of our system.
We will focus this chapter on exploring and testing both implementation methods and ultimately reviewing our findings.

4.2 Real System Implementation

The theoretical real-world system is designed to operate at an industrial scale, utilizing professional-grade equipment and advanced network architectures to protect vast forest environments. This section outlines the intended materials and the fully autonomous workflow of the system in a real-life scenario.

4.2.1 Sensing Layer

In a real-life deployment, the system relies on high-altitude surveillance and a dense mesh of ground sensors.

Industrial Camera Towers:

Instead of standard infrared, the system would utilize high-definition thermal imaging cameras (such as FLIR A500f/A700f Advanced Smart Sensors). These provide precise heat signatures even through thick smoke or at extreme distances, allowing for detection long before a flame is visible to the naked eye.

NetWorked Sensor Nodes:

Each node is housed in a weather-resistant, solar-powered enclosure. these units are equipped with industrial-grade gas sensors (CO, CO2) and high-precision NEO-M8N GPS modules to provide the server with sub-meter location accuracy.
The data collected by these nodes is later on processed in the server; the fire detection

decision is made using a dynamic threshold system that changes based on the current season and real-time weather forecasts.



Figure 4.1: High-altitude Thermal Camera Tower Overlooking a Dense Forest Canopy

4.2.2 Communication Layer

The real-world architecture shifts from short-range Wi-Fi to long-range protocols to handle the vast distances of a forest, LoRaWAN integration for the sensing nodes, due to its low power consumption and ability to transmit data over several kilometers, which makes it ideal for deep-forest deployment where power and infrastructure are unavailable. As well as 5G/satellite uplink for the drone's high-bandwidth needs, such as live 4K video feeds, the system utilizes 5G or satellite communication to ensure a constant connection with the central server regardless of the drone's distance from the base.

4.2.3 Processing Layer

The processing power is scaled to handle massive data streams and complex environment analysis.

Cloud-Based AI Server:

The server is envisioned as a high-performance cluster capable of running multiple heavy-weight machine learning models simultaneously. It processes data from hundreds

of towers and nodes in real-time, using advanced predictive analytics to estimate fire spread based on wind and humidity data.
False alarms are first avoided by attempting to confirm the node data with the camera and vice versa, then double-checked in case it got to the next step using the drone's live camera feed feature so the ground team confirms if there's a fire. In the case of a false alarm occurring, the worst-case scenario is the ground team being on alert and the drone going to the false fire location and finding nothing, wasting only battery life.

Edge-Computing Drones:

In this vision, the drone is the powerhouse of the edge. Equipped with onboard NVIDIA Jetson modules, the drone performs its own ML image processing locally, reducing the need for constant high-bandwidth streaming and allowing for near-instant decision-making.

4.2.4 Actuation Layer

The Autonomous Drone System:

The core of the actuation layer is the fully autonomous Unmanned Aerial Vehicle (UAV), designed for rapid response without human intervention. Utilizing SLAM (Simultaneous Localization and Mapping) and LiDAR, the drone navigates complex forest terrains and avoids obstacles automatically. It does not require a "Pilot" or a GCS interface for movement, as it calculates the optimal flight path based on GPS coordinates received from the server. The drone is an industrial-scale hexacopter capable of carrying a 100kg fire-extinguishing payload. It uses its onboard thermal vision to pinpoint the heart of the fire and releases the payload at the mathematically optimal altitude and angle to ensure maximum suppression.



Figure 4.2: An Industrial Hexacopter Drone Autonomously Releasing a Payload over a Localized Fire Spot

4.2.5 Simulation Workflow

A node detects a spike in CO2 or a tower identifies a thermal anomaly, which prompts the server to calculate the coordinates and dispatch the nearest autonomous drone. The drone then navigates to the site, confirms the fire via edge-AI, and neutralizes it. After that, the drone returns to a specialized solar-charging pad, recognizing it via QR or ArUco markers, and redirects itself for the next mission.

4.3 Practical System Implementation

Realising and testing a prototype of the planned system will prove the practicality and usefulness of the previously established design and conceptualization of our system. The choices made when it came to picking the hardware components and programming language were dictated by effeciency and cost effect, preferring cheaper but more reliable options over feature rich but costly components, all with the goal in mind being a functional prototype that serves as a solid proof of concept.

4.3.1 Sensing Layer

The system relies on two main systems to detect the presence and the location of forest fires as quickly as possible: a main camera tower, and nodes layed out across the forest.

Camera Tower:

We opted for a OV5647 infrared camere, due to its wide field of vision (160 degrees) allowing it to cover a vast area of space to spot as many fires as possible. but its main characteristic that made it at the top of our list of candidates, is its night time infrared spectrum vision, which fires and hot objects are known to be extremely visible in, making it an extremely good second best option to thermal cameras. The figures below demonstrate the wide range capabilities of the camera as well as its effectiveness in infrared vision and fire detection.
this figure focuses view on the camera tower in the miniature model.

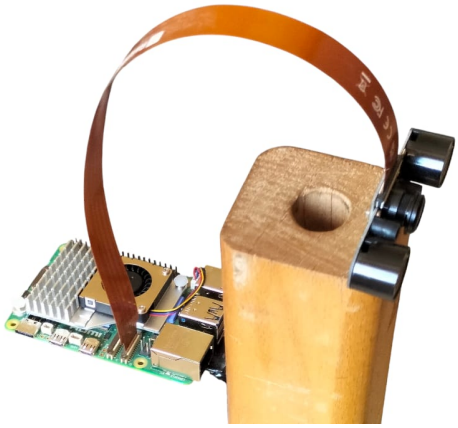


Figure 4.3: The Camera Tower

Nodes:

Nodes are what we call the device that is scattered around the forest to help the camera tower during fire detection by collecting data in areas that either can't be seen by the camera tower or are too far away. The components of the node are the following:

- ESP32 as the brains of the node that will perform the data collection forwarding.
- DHT22 as the secondary sensor that looks out for crucial temperature and humidity shifts in the environement.
- MQ-2 smoke sensor as the main sensor, it looks out for any slight change rise in smoke levels in the area.

Unlike the real-world implementation, the threshold system has been kept static for the sake of simplicity and lower loads on the server hardware.

Additionally, the node is supposed to have an NEO-M8M GPS sensor that lets the server and the drone know the location of the node and the potential fire by extention, this sensor however was left out of the prototype due to its inaccuracy in the scale of the miniature model, its margine of error of 6 meters is larger than the model itself and any replacement that could give feasible results was simply out of budget. the alternative was assigning static location to the nodes in the server.
Below is an image of the node and its sensors up close.

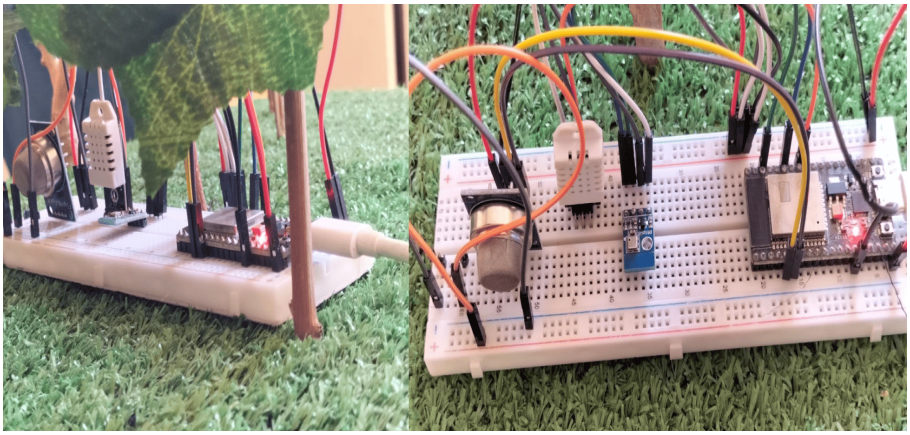


Figure 4.4: Node Sensors

4.3.2 Communication Layer

The chosen communication protocols for this demonstrative practical system is WIFI and MQTT for data communication, due to their seamless integration into esp32 chips, and for low power consumtion and latency. LoRaWAN wasn't chosen due to the change in drone communication architecture from the theoretical real world application to the miniature demonstration project where the drone's hardware isn't capable of reaching the edge computing power necessary to perform its job, and having to offload a lot of its processing needs, like live image streaming, to the server instead, therefore needing a lot more communication bandwidth than LoRaWAN can support, which Wi-Fi is more than capable of supporting.

4.3.3 Processing Layer

After data collection from the environement, the data processing occures in two main places, with the majority being in the first one which is the server, as well as the drone.

Server:

The server is a Raspberry Pi 5, which is the powerhouse of our entire system. It alone is responsible for running the machine learning model on two separate threads (one for processing the main tower camera live feed, while the other is processing the images received from the drone camera), as well as commanding the drone movement and giving it orders to drop payload or change mode, while also sending live node data and live camera tower feed to the website. It also is the MQTT server that hosts all communication between the the system devices. All system data processing and decision making is done by the server.

It also has the majority of the architecture built into it; the data flows through the scripts as follows:

- Node_Reception: collects data being published by the nodes.
- Model_Cam: joins the camera functions with the machine learning image processing model.
- Website_Forwarding: Sends the node and camera data over to the website.
- Drone_Translator: takes the output of the model and translates it into sensible directions that the drone can follow.
- Movement_Logic: decides the drone's movement and mode based on the angle determined by drone_translator.
- Image Receiver: receives camera feed from the drone's camera and runs it through the ml model to decide which phase of action it should be in.
- GCS: The user interface that allows a drone pilot to guide the drone exactly to the location of the fire, as well as allowing him to see through both the camera tower and the drone's camera.

The server is also hosting the website; all its files and databases are stored, executed, and hosted on the server.

False alarms are avoided by having a reasonably high detection threshold for smoke and heat data, as well as a reliable model for camera detection, with the worst case scenario being a pointless drone flight.

Fire Detecting ML Model:

1. Data Collection Stage

In this stage, we collected images of forest fires from different sources accurately in order to improve the performance of the detection model.

2. Data Processing Stage

After the data collection process, at this stage we process this data by identifying fire areas in the images using Roboflow.

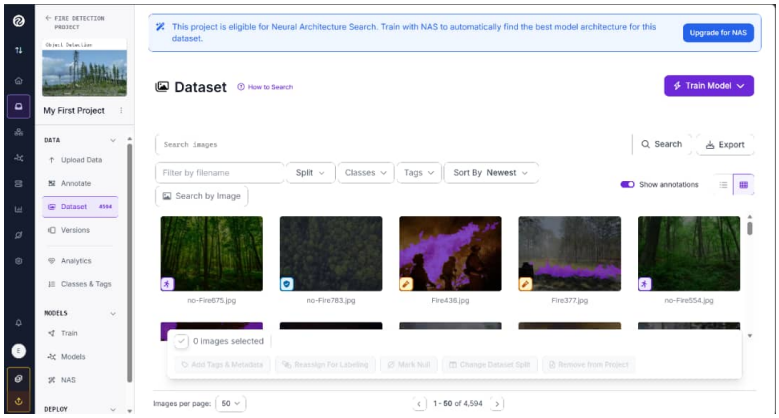


Figure 4.5: Data Processing

After the sorting process, we obtained 6231 images for training, 920 images for verification, and 459 images for testing, as shown in the image.

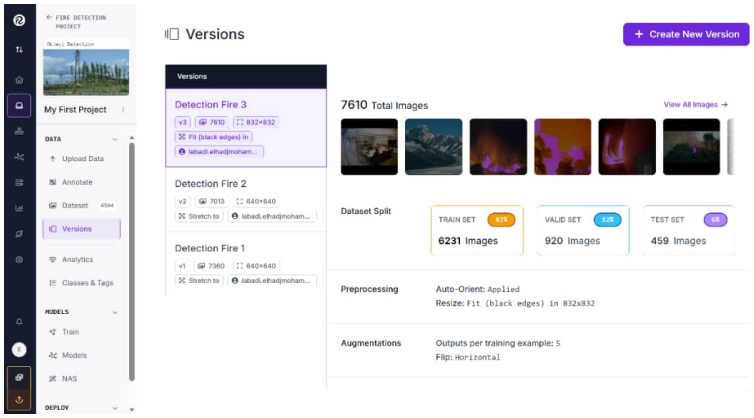


Figure 4.6: Sorted Data

3. Model Training Stage

At this stage, we trained the YOLOv8 model using the processed data. At this stage, we focus on obtaining a model that helps us detect fires that are difficult to detect. This model is trained on the Google Colab platform, which provides us with a GPU to enable us in the training process.

```
!yolo task=detect mode=train model=yolov8n.pt data={dataset.location}/data.yaml epochs=90 imgsz=832 batch=16 project=/content/drive/MyDrive/new2_

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
       4/90      4.06G      1.551      2.086      1.624          3       832: 100% ━━━━━━━━━━━━ 390/390 2.2it/s 3:01
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% ━━━━━━━━━━━━ 29/29 2.3it/s 12.4s
                   all        920       1096      0.378      0.354      0.307      0.151

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
       5/90      4.06G      1.507      2.016        1.6          9       832: 100% ━━━━━━━━━━━━ 390/390 2.2it/s 2:59
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% ━━━━━━━━━━━━ 29/29 2.5it/s 11.7s
                   all        920       1096      0.384      0.359      0.304      0.151

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
       6/90      4.06G       1.45       1.95      1.564         20       832: 100% ━━━━━━━━━━━━ 390/390 2.2it/s 3:00
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% ━━━━━━━━━━━━ 29/29 2.8it/s 10.4s
                   all        920       1096       0.39      0.389      0.332      0.172

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
       7/90      4.06G      1.423      1.911      1.539          7       832: 100% ━━━━━━━━━━━━ 390/390 2.2it/s 2:55
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% ━━━━━━━━━━━━ 29/29 2.5it/s 11.5s
                   all        920       1096      0.433      0.428      0.394      0.221

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
       8/90      4.06G      1.391      1.873      1.523         16       832: 100% ━━━━━━━━━━━━ 390/390 2.2it/s 2:58
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% ━━━━━━━━━━━━ 29/29 2.5it/s 11.4s
                   all        920       1096      0.474      0.443      0.414      0.213

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
       9/90      4.07G      1.384       1.84      1.512         16       832: 100% ━━━━━━━━━━━━ 390/390 2.2it/s 2:55
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% ━━━━━━━━━━━━ 29/29 2.4it/s 11.9s
                   all        920       1096      0.462      0.411      0.409      0.231
```

Figure 4.7: Model Training

4. Results Stage

After completing training the model, it achieved good accuracy after the testing process, where we exported the trained model as a “best.pt” file for use in our project.

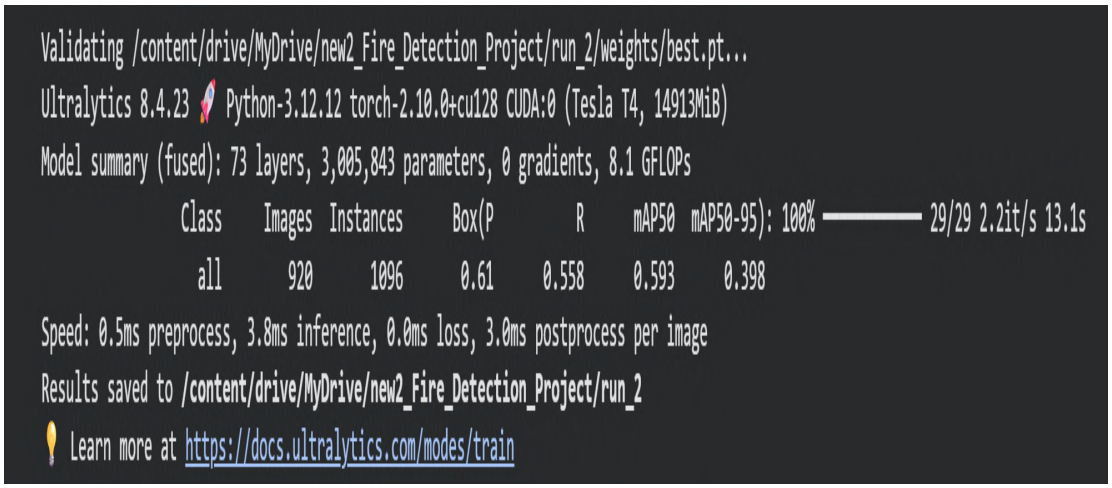


Figure 4.8: Results Model Training

These charts show the results of training the model in terms of accuracy, number of epochs, and other values by which the model’s accuracy is determined.

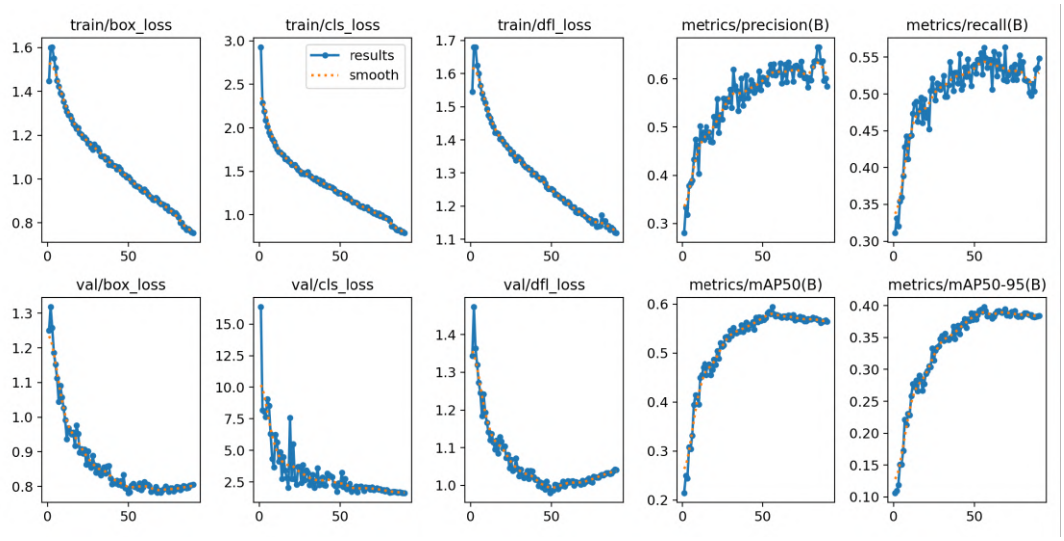


Figure 4.9: Performance curves and evaluation metrics for the training model

This image shows predictions of some images for detecting fire by measuring the confidence rate.



Figure 4.10: Testing YOLOv8 Model

Drone:

While the original design meant for the drone to be autonimous in more aspects than just the movement, hardware and budget constraints led to the ESP32-CAM being the microcontroller responsible on board the drone, which made that design unrealizable in this prototype. Controls are instead dictated by the server after image analysis, but not always. During lock mode, precise and responsive movement needs to happen, and relying on the ESP32-CAM to reliably live stream data to the server without latency is simply unrealistic. We opted for a quick, simple solution that involved the ESP32-CAM performing a simple but effective image processing algorithm that locates the brightest pixels on the frame that are almost certainly going to be the exact pixels of the fire during that stage for effective edge vision and processing.

4.3.4 Actuation Layer

After data collection and processing is done, action is the logical next step. Our system has two main actions that follow a detection of a forest fire: alerting and extinguishing, performed by the website and drone respectively.

Alert:

Alerts are triggered as soon as either a fire is detected by the camera or the node data being received reaches the threshold set in the server. Alerts are displayed on the first page of the website as well as a high-framerate live camera feed from the tower camera.

Drone:

As mentioned previously, the main component of the drone's hardware is the ESP32-CAM, which serves as the brain of the drone. It is also fitted with a gyroscope module that lets it orient itself towards the fire or towards its landing pad when returning from action, as well as a servo motor for the payload release mechanism that completes its purpose as a quick-action drone.
Below is a figure that showcases the circuit that is attached to the drone:

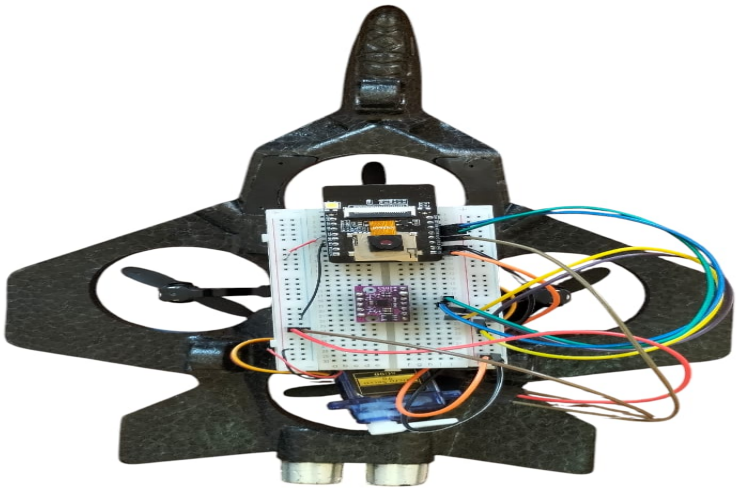


Figure 4.11: Drone

The main functions of the drone are the camera, WiFi and MQTT connection, the live camera feed hosting, gyroscope data collection and publication, and servo control. All these functions have to be cleverly distributed across the ESP32-CAM's limited resources, which is why we split the functions over the two cores; core 1 houses the heavy, extensive functions that allow room for delay, such as the Wi-Fi, MQTT, and livestream functions,

while core 0 is reserved for functions that can cause accuracy issues if delayed or lagged by other functionalities, such as the gyroscope, servo and lock mode image processing. After fire detection by either the camera tower or the nodes, the drone goes through 5 stages:

- Listen mode, listens for orders from the server.

- Search mode, after receiving an order from the server to travel forward at the angle that the fire was detected at. during this mode, an image is sent twice per second (for processing power conservation) to the server for fire detection. If a fire is not detected, it continues to send forward commands. If a fire is detected, the server orders the drone to go to the next stage.

- Lock mode, during this mode the live camera stream to the server cuts, and the drone utiizes and algorithm to detect the brightest pixel on camera, which at this stage is most definetly the fire, and sends the coordinates of those pixels to the server. During this mode, the drone also listens for a drop command from the server to drop the fire extinguishing payload onto the fire.

- Wait mode, which aims to confirm the success or the faliure of the action, this determines if a second payload should be carried back to the fire.

- Return mode, that turns the drone around to face the opposite angle of the fire relative to the camera tower (and the landing pad by extention). During this mode, the drone is utilizing a simple yet effective algorithm that recognizes QR codes instantly to be able to quickly locate the landing pad that has one printed on it. The previous mode decides if the drone goes back to listen mode or search mode to double down on the action.

4.3.5 User Interface Layer

The user interface was carefully considered, as the system is supposed to be autonomous after all. The two interfaces that users get to interact with are the website and the Ground Control System (GCS for short) in the server.

Website:

The website is a crucial part of the system that allows for alerts to be sent to users all over the area, which gives residents nearby a lot more time to relocate to a safer place. It also serves to aid the fire department by giving them an early warning about it with precise location and visual coverage that helps assess the severity of the situation in order to take the correct measures and reduce losses.

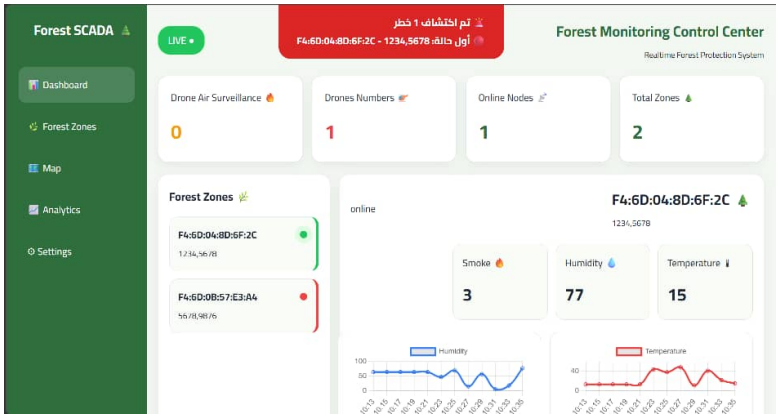


Figure 4.12: Website

Ground Control System

GCS is a graphic interface made using Python's Tkinter and integrated into the server; it was made as a compromise due to the unavailability of an autonomous drone or even a server-controlled drone. Its main purpose is to serve as the pilot for the drone, allowing a user to be able to guide the drone to the location of the fire without having to make a human decision. Its secondary purpose is to demonstrate the server's ability to fully dictate the drone's movements and actions and fulfill its purpose without error if it weren't for hardware limitations.

The main panels in GCS are the camera tower view panel, the movement control panel that highlights the direction that the drones should be moving in, the compass panel that shows the direction of the fire from the tower's view compared to the direction the drone is facing, the lock mode view panel that makes it easier for micro-adjustments to be made precisely, and finally, the drone camera view panel.

Below is a figure that shows the panels.

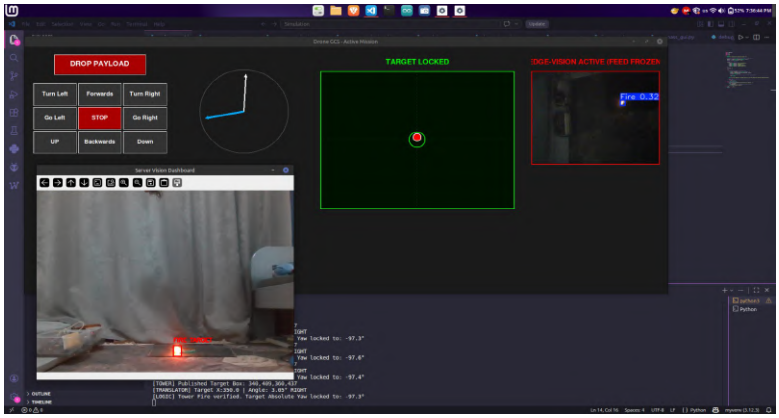


Figure 4.13: Image of GCS UI

4.4 System Testing

We will test the system by first stress testing each layer individually and then a full end-to-end test of the entire system.

4.4.1 Sensing Layer Testing

When it comes to the nodes, the performance lines up extremely well with our expectations, given that the MQ-2 sensor is given its proper startup period needed to heat up the chemical reaction that it uses to function. Testing revealed that the reaction time of the node sensors is just below one second.

The camera tower also functions extremely well despite the lack of thermal imaging components. Testing revealed that fires take AT MOST 15 seconds to be discovered, which is not enough time for them to become a threat just yet. That period lines up with the low-power mode we intentionally set for it, so performance can be enhanced and optimized to balance power consumption and safety.

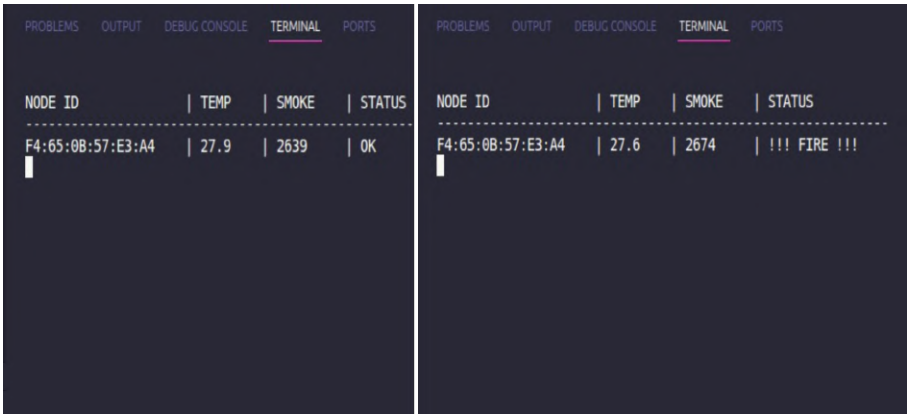


Figure 4.14: Testing Node

4.4.2 Communication Layer Testing

WiFi and MQTT are an extremely solid combo that olny started to drop in stability and rise in latency when tested across long distances ($>$ 15 meters: moderate drop in performance | $>$ 20 meters: severe drop in performance), or extremely thick and insulated barriers ($>$ 20 centimeters: moderate drop in performance | $>$ 25 centimeters: severe drop in performance).

4.4.3 Processing Layer Testing

The Raspberry Pi 5 sever handles all the tasks extremely well, with the moldel being able to process images in less than 100 ms per frame. as well as running all the frone commanding functions and website hosting with little to no latency nor delay. heating up of the chips is accounted for with an active cooler.

4.4.4 Full System Testing

The prototype focuses more on the camera as a main sensor, so that's what was used in the full system testing scenario. The test went like so:
A fire was lit if front of the camera at a slight offset from the exact center of the camera view, which it detected and alerted in the website after only 3 seconds.
The drone translation system worked very well and managed to pick up the slight angle the fire was at, and was communicated to the drone movement system all while a live camera feed showed up on the gcs coming from the drone.
The gcs showed an order given by the movenent system to the drone to lift off then to

turn to the side the fire was at, while giving it clear angle overlay to line it up perfectly. As soon as that was done the movemnt logic system showed an order throught the gcs to start moving forward while the second model thread was analysing each image being sent by the drone.

The drone successfully got on top of the location of the fire, and the gcs displayed a fire detection on the drone live camera feed, as well as a message saying that the drone is switching modes. The Lock mode view on the gcs lit up into action and started showing a red dot moving across the panel's surface, as well as micro adjustment orders to lign the fire up perfectly under the drone, which was succesfully performed, a button then lit up on the left hand pannel right after receiving an order to stop all movement.

The button says "drop payload", and it does exactly as described after clicking it.

The payload that the drone was carrying was dropped right next to the fire, approximately 10 to 12 centimeters away, which caused the fire to get extinguished anyway due to the heat of the fire affecting the outer layer of the fire extinguishing ball.

The drone stopped receiving orders for about 10 seconds, then went back to sending live camera feed to the server for the next 5 seconds.

The drone then started receiving orders through the GCS to turn around, facing the exact inverse of the angle it was ordered to before, right before receiving forward commands again.

The live camera feed from the drone to the server stopped, and a "Return mode" message showed up on the GCS.

As soon as the landing pad QR code was in view of the drone's camera, a "LAND" command was sent to it, just for the drone to drop about 8 centimeters away from its intended landing square.

4.5 Results and Discussion

The prototype underwent a full-system testing scenario focused on the camera tower's detection capabilities and the drone's autonomous response. Results and observations will be discussed briefly.

4.5.1 Results

The observable results are summarized below:

- Detection Speed: The OV5647 infrared camera tower successfully detected a simulated fire at a slight offset from center in approximately 3 seconds.
- Navigation Accuracy: The drone_translator script correctly identified the target angle, and the drone performed the corresponding liftoff and rotation maneuvers.
- Mode Transitions: The system successfully cycled through the Search, Lock, and Return modes.
- Extinguishing Success: During "Lock mode," the brightest-pixel algorithm allowed for precise micro-adjustments. The payload was released and landed 10 to 12 centimeters from the target; however, the heat from the fire successfully triggered the extinguishing ball despite the margin of error.
- Recovery: The drone utilized its QR code recognition algorithm to locate the landing pad, though it landed 8 centimeters away from the center of the intended landing square.

4.5.2 Discussion

While the system successfully demonstrated a complete detection/alert/action loop, the testing phase revealed several critical insights regarding the prototype's reliability and hardware limitations.

The dual-thread processing on the Raspberry Pi 5 proved to be highly effective, allowing the server to manage the tower and drone feeds simultaneously without noticeable processing lag, and the decision to offload time-critical tasks to the ESP32-CAM's Core 0 ensured that "Lock mode" remained responsive even when the Wi-Fi connection was under load.

That successful test, however, doesn't show the whole picture. Across multiple test runs, the system experienced intermittent hardware malfunctions that sometimes led to unexpected drone behavior. This is due to the processing power of the ESP32-CAM on board

the drone sometimes falling short of the required tasks given to it. As well as the power source that supplied it being inconsistent and underpowered at certain times when the ESP32-cam is in a high power demand period.

4.6 Conclusion

Despite the hardware limitations of the prototype, its performance serves as an excellent proof of concept that validates the functionality and feasibility of the proposed system.

These results confirm that the suggested theoretical real-life implementation would perform extremely well given the higher budgets and increased development time available in a professional setting. Such an environment would open the doors to more detailed software, higher-accuracy hardware, and experienced development teams.

General Conclusion

In this work, we proposed and designed our smart system to detect and respond to forest fires based on the Internet of Things and artificial intelligence. We presented the problem of forest fires, the losses they cause every year, and the limitations imposed on traditional methods of detection. Then we studied some current systems related to the topic. After that, we built our system through fire mechanisms and trained models for detection through artificial intelligence and automatic extinguishing operations using a drone.

In this project, the result was the Internet of Things and artificial intelligence could offer smart and efficient solutions to detect and respond to forest fires at the early stages as sensors and cameras helped in continuous monitoring of forests, while artificial intelligence models made it possible to perform intelligent analysis of the collected data and images to take precise actions in real time. Drones have also been instrumental in combating fires and accessing areas that are challenging for humans to reach.

Even though the system works well, it has several possible enhancements to be made in the accuracy of the AI models, energy consumption, smart cooperation among the sensor networks, and the communications technology to cover larger forest areas. These views open new horizons for the design of smart and sustainable systems for forest protection and fire-related risk reduction.

Bibliography

[1] Fire. https://www.marefa.org/fire, 18:30, 12-05-2020. Accessed: 04/04/2026.

[2] Firefacts: Triangles. https://www.nwfirescience.org/sites/default/files/publications/FIREFACTS_Triangles.pdf, 2015. Accessed: 20/02/2026.

[3] Fire extinguisher types, 2023, October 4. Accessed: 03/03/2026.

[4] Forest_fire. https://www.un-spider.org/category/disaster-type/forest-fire. Accessed: 20/02/2026.

[5] Over 800 fires recorded in algeria in 2025. https://www.atlas-mag.net/en/articles/over-800-fires-recorded-algeria-2025-0, 2025, October 17. Accessed: 28/02/2026.

[6] Fourth international symposium on fire economics, planning, and policy: Climate change and wildfires. https://www.govinfo.gov/content/pkg/GOVPUB-A13-PURL-gpo38206/pdf/GOVPUB-A13-PURL-gpo38206.pdf, 5 November 2012. Accessed: 19/02/2026.

[7] António Bento-Gonçalves. Algeria suffers from devastating wildfires, but faces big challenges in addressing them. https://www.preventionweb.net/news/algeria-suffers-devastating-wildfires-faces-big-challenges-addressing-them, 2 September 2021. Accessed: 01/03/2026.

[8] Isastur, agents extinguishing agents. https://www.isastur.com/external/seguridad/data/en/1/1_6_2_1.htm. Accessed: 21/02/2026.

[9] https://www.ibm.com/think/topics/internet-of-things. Accessed: 05/04/2026.

[10] Samuel Greengard and Britannica Editors. Internet of things. https://www.britannica.com/science/Internet-of-Things, 12/03/2026. Accessed: 2026-03-31.

[11] B.J. Copeland. Artificial intelligence. https://www.britannica.com/technology/artificial-intelligence, 12/03/2026. Accessed: 2026-04-01.

[12] Dryad Networks. Dryad: Ultra-early wildfire detection. https://www.dryad.net/, 2021. Accessed: 2026-05-29.

[13] AlertWildfire. Alertwildfire camera network for wildfire monitoring. https://www.alertwildfire.org/, 2013. Accessed: 2026-05-29.

[14] Université Saad Dahlab Blida. Smart fire detection and alert system: Drone ai-iot integration. https://di.univ-blida.dz/xmlui/bitstream/handle/123456789/31380/Smart%20Fire%20Detection%20and%20Alert-System%20Drone%20AI-IoT%20Integration.pdf?sequence=1&isAllowed=y, 2024. Accessed: 2026-05-29.

[15] Belhadj Bouchaib University Center of Ain-Temouchent. Forest fire detection and monitoring system. https://dspace.univ-temouchent.edu.dz/server/api/core/bitstreams/ea1fb7c0-40a4-40e9-bda6-5b03ac2aafaa/content, 2019. Accessed: 2026-05-29.

[16] C. Arulstephen, R. K. Raghulsharma, B. Ebenezer Abishek, and P. Satish Kumar. Wireless sensor based monitoring system for forest tree poaching and fire using iot. https://www.researchgate.net/publication/327208820_Wireless_Sensor_Based_Monitoring_System_for_Forest_Tree_Poaching_and_Fire_Using_Iot, August 2018. Accessed: 2026-05-29.

[17] Wi-fi. https://www.wi-fi.org/. Accessed: 07-05-2026.

[18] Mqtt: The standard for iot messaging. https://mqtt.org/, 2024. Accessed: 07-05-2026.

[19] What is lorawan®? https://lora-alliance.org/about-lorawan/, 2024. Accessed: 07-05-2026.

[20] Hzh xf100 firefighting drone. https://www.hongfeidrone.com/hzh-xf100-firefighting-drone-100kg-heavy-lift-uav-for-forest-wildland-fire-fight Accessed: 17-05-2026.

[21] Experimental study and application of unmanned aerial vehicle releasing fire-extinguishing bomb on transmission line wildfires, 2024. Accessed: 17-May-2026.

[22] Ultra long-range multi-sensor ptz camera system. https://www.infinitioptics.com/cameras/vega. Accessed: 17-05-2026.

[23] Bme688 development kit. https://www.bosch-sensortec.com/media/boschsensortec/downloads/product_flyer/bst-bme688-fl001.pdf. Accessed: 17-05-2026.

[24] Zed-f9p module. https://www.u-blox.com/en/product/zed-f9p-module. Accessed: 17-05-2026.

[25] Sgp41. https://sensirion.com/products/catalog/SGP41. Accessed: 17-05-2026.

[26] 5mp ov5647. https://docs.arducam.com/Raspberry-Pi-Camera/Native-camera/5MP-OV5647/. Accessed: 05-05-2026.

[27] Mq-2 smoke/gas sensor: Datasheet, pinout & working. https://www.theengineeringprojects.com/2024/02/mq-2-smokegas-sensor-datasheet-pinout-working.html, 21/02/2024. Accessed: 05-05-2026.

[28] Ai thinker esp32-cam. https://www.espboards.dev/esp32/esp32cam/. Accessed: 05-05-2026.

[29] Lithium polymer battery by capacity. https://www.batterylipo.com/lithium-polymer-battery-list-by-capacity/. Accessed: 06-05-2026.

[30] Esp32. https://www.espressif.com/en/products/socs/esp32. Accessed: 07-05-2026.

[31] Raspberry pi 5. https://www.raspberrypi.com/products/raspberry-pi-5/. Accessed: 07-05-2026.

[32] Digital-output relative humidity & temperature sensor/module dht22. https://cdn.sparkfun.com/assets/f/7/d/9/c/DHT22.pdf. Accessed: 07-05-2026.

[33] Arduino integrated development environment (ide). https://www.arduino.cc/en/software/. Accessed: 06-05-2026.

[34] Roboflow computer vision platform. https://roboflow.com/. Accessed: 07-05-2026.

[35] The open source ai code editor. https://code.visualstudio.com/. Accessed: 07-05-2026.

[36] Google colaboratory. https://colab.research.google.com/. Accessed: 07-05-2026.

[37] Object Management Group. What is uml. https://www.omg.org/uml/what-is-uml.htm, July 2005. Accessed: 20-Apr-2026.

[38] Use case diagram. https://en.wikipedia.org/wiki/Use_case_diagram, 2025. Accessed: 20-Apr-2026.

[39] Activity diagrams - unified modeling language (uml). https://www.geeksforgeeks.org/system-design/unified-modeling-language-uml-activity-diagrams/, 21-01-2021. Accessed: 08-05-2026.

[40] Careerride. Uml – what are the types of diagrams? https://www.careerride.com/UML-what-are-types-of-diagrams.aspx. Accessed: 20-Apr-2026.