CHAPTER 1

INTRODUCTION

Forest fires are as old as the forests themselves. when there is no rain for months during summer, the forests become littered with dry leaves and twinges, which could burst into flames initiated by even the slight spark They pose a threat not only to the forest wealth but also to the entire regime to fauna and flora seriously disturbing the bio-diversity and the ecology and environment of a region, also there is a danger for wild life, domestic crops and to the nearest people.. So there is a necessary to avoid the excess of losses due to forest fire by controlling the fire in its early stages.

In the present technologies like ruled base image processing and MODIS systems there are many drawbacks like having high false alarm rate where alarm notification may not give a proper and exact notification at emergencies, also response time is quite big as they use robots to extinguish the fire accidents and temporal representation of the affected area. The main drawback is there is no facility of getting direct notifications to the authorities immediately at the time of fire burst, all these drawbacks may leads to the rapid spread of the forest fire throughout the larger area of the forest leading to major damage and loss. It requires human force in big number which is a risky process. In the proposed system we have used high sensitive sensors which reduces the false alarm rate by enhanced data collection. sensors and micro controller will continuously monitor and sends the data to the hub as value due to which minimization of the false alarm is achieved also response time is minimized. Cost is affordable and minimized the human efforts. The main aim of our project is detection and monitoring the forest fire and to minimize the effect of fire breakout by controlling in its early stage also to protect wild life and domestic crops by informing about the fire breakout to the respective forest department as early as possible. We have implemented the IOT technology to achieve our objective. There are numerous natural and human causes of wildfires. Natural causes include lightning, heatwaves, and dry weather conditions while human causes include faulty power lines, human negligence, and arson. However, over the past several decades, the frequency of wildfires has increased. With climate change occurring, there have been higher temperatures which increase the potential for a wildfire to spark. According to Running, the weather has been contributing to the wildfire intensity. Due to warmer temperatures, snowpacks have been melting sooner, causing longer dryer seasons which leave more tinder and brush on the forest floor, which helps fires to burn longer . With the increase in their intensity over the past several decades, fires have been burning more uncontrollably and for longer periods, burning more acres of forestland.

Wildfires have devastating consequences, causing significant damage to ecosystems, human lives, and infrastructure. Early detection of wildfires is crucial to minimize their impact and facilitate effective fire management. With the rapid advancements in Internet of Things (IoT) technologies, there has been a growing interest in developing IoT-based wildfire early detection systems. These systems leverage sensors, cameras, data analysis algorithms, and communication networks to enable real-time monitoring and prompt detection of wildfires. By providing timely alerts to emergency responders and authorities, these systems aim to mitigate the destructive effects of wildfires.

The objective of this paper is to provide a comprehensive exploration of an IoT-based wildfire early detection system, examining its architecture, components, working principles, advantages, challenges, and future prospects. By understanding the technical details and societal implications of such a system, we can contribute to the development of more effective wildfire management strategies.

Wildfires are caused by various factors, including human activities and natural phenomena such as lightning strikes. Their impacts extend beyond the immediate destruction of vegetation and habitats, affecting air quality, water resources, and climate patterns. Prompt detection is critical to initiate rapid response actions, such as evacuation plans and firefighting operations. However, current wildfire detection methods often rely on manual observation, which can be time-consuming and limited in scope. The integration of IoT technologies offers a promising solution to overcome these limitations and improve early detection capabilities.

The IoT-based wildfire early detection system incorporates a network of sensors and cameras strategically deployed in high-risk areas. These sensors can detect critical indicators such as smoke, heat, and gas emissions associated with wildfires. Cameras provide visual data, enabling better situational awareness and aiding in the identification and monitoring of fire behavior. The collected data is transmitted to a central platform for analysis, where sophisticated algorithms are employed to identify patterns and anomalies indicative of a wildfire.

The advantages of an IoT-based approach are numerous. Real-time monitoring allows for immediate detection and timely response, reducing the risk of uncontrolled fire spread. The system can cover a large geographical area, providing comprehensive coverage and enabling early detection in remote or inaccessible locations. Additionally, the scalability and adaptability of the system allow for customization to different environments and integration with existing emergency response systems.

1.1 LITERATURE SURVEY

Y.Hakan Habiboglu et al. [1] proposed a method to detect fire by smoke detection based on wavelet. In this smoke detection method, ideo based wildfire detection system that based on spatio-temporal correlation descriptors is developed. During the initial stages of wildfires smoke plume becomes visible before the flames. The proposed method uses background subtraction and color thresholds to find the smoke colored slow moving regions in video. These regions are divided into spatio-temporal blocks and correlation features are extracted from the blocks. Property sets that represent both the spatial and the temporal characteristics of smoke regions are used to form correlation descriptors. An SVM classifier is trained and tested with descriptors obtained from video data containing smoke and smoke colored objects. Experimental results are presented.

T. Celik et al. [2] further enhance system that uses a statistical color model with Fuzzy logic for fire pixel classification. The proposed system novel models for fire and smoke detection using image processing is provided. The models use different colour models for both fire and smoke. The colour models are extracted using a statistical analysis of samples extracted from different type of video sequences and images. The extracted models can be used in complete fire/smoke detection system which combines colour information with motion analysis.

Zhao, Y et al. [3] proposed another method for smoke detection based on Depthwise Separable Convolutions and Target-Awareness. In this approach algorithm based on depthwise separability and target awareness is proposed. Existing deep learning methods with convolutional neural networks pretrained by abundant and vast datasets are always used to realize generic object recognition tasks. In the area of smoke detection, collecting large quantities of smoke data is a challenging task for small sample smoke objects. The basis is that the objects of interest can be arbitrary object classes with arbitrary forms. Thus, deep feature maps acquired by target-aware pretrained networks are used in modelling these objects of arbitrary forms to distinguish them from unpredictable and complex environments.

S. Kundu et al. [4] proposed another method that uses a Highly Accurate Fire Detection Method Using Discriminate Method. In this method, Algorithms of image processing have been developed over the years for fire detection. Such algorithms have been discussed here. Although these algorithms are able to detect fire but unable to differentiate between fire and orange color. In our proposed method, we developed a discrimination algorithm to differentiate between fire and orange color. This algorithm is applied on the detected fire pixels to check whether they are fire pixels or orange color pixels.

H. Afzaal et al. [5] proposed another method that Robot-based forest fire detection and extinguishing model. In this method Temperature sensors are used to detect and actors to extinguish the fire. The actors are deployed randomly in the forests forming clusters. Clustering, sleep/active schedule and idle/working modes, is used to minimize consumption of energy. WSANs are modeled using graph theory for semi-formal representation of the model. Logical architecture of the system is presented which helped to develop algorithm and it is then transformed into an equivalent formal model using Vienna Development Method-Specification Language (VDM-SL). The model is analyzed through various existing techniques in VDM-SL Toolbox to prove its correctness.

1.2 PROPOSED SYSTEM

The Wildfire Early Detection System (WEDS) is a device that monitors conditions in areas prone to wildfires. Each unit contains a flame sensor and a camera to detect a fire. Based on the flame and the images received by the camera, the Raspberry Pi computer can determine if a fire is nearby. Once a fire has been detected, a signal is sent to the main central hub where it notifies first responders of the fire’s location. Because of the limited communication range, multiple devices would communicate with each other and would daisy chain back to a central hub . The unit would be mounted on a pole to provide wide viewpoints to cover as much area as possible. Ideal locations for these devices include densely forested areas that experience dry weather periods and/or the border of human communities.

Advantages

Wildfire early detection systems offer numerous advantages in mitigating the impact of wildfires. Here are some of the key advantages:

- **Rapid response:** Early detection systems enable a quicker response to wildfires. Once a fire is detected, emergency response teams can be alerted promptly, allowing them to initiate suppression efforts and containment strategies. Rapid response helps prevent fires from spreading uncontrollably, reducing property damage and the potential for loss of life.
- **Improved accuracy:** With advancements in technology, wildfire early detection systems have become more accurate in identifying and locating fires. These systems employ a range of sensors, including thermal imaging, smoke detection, and infrared cameras, to detect the presence of fires even in challenging conditions. This accuracy aids in timely decision-making and targeted firefighting efforts.
- **Real-time monitoring:** Wildfire early detection systems provide real-time monitoring of fire behavior and progression. Through the use of surveillance cameras, satellite imagery, and other monitoring tools, emergency responders can gain valuable insights into the size, direction, and intensity of a fire. This information enables them to make informed decisions, adapt firefighting strategies, and ensure the safety of both responders and affected communities.
- **Reduced Fire Spread:** Early detection allows for a quick assessment of the fire's location, size, and direction of spread. This information helps in developing effective firefighting strategies, such as establishing containment lines, deploying firefighting resources strategically, and coordinating firefighting efforts. By containing the fire early, the system can limit the fire's spread, reducing its overall impact.
- **Cost Savings:** Detecting wildfires at an early stage can result in significant cost savings. Early suppression efforts are generally more efficient and require fewer

resources compared to large-scale firefighting operations. By preventing the fire from growing into a larger and more destructive blaze, the associated costs of suppression, property damage, and ecosystem restoration can be minimized.

- **Environmental Preservation**: Early detection systems play a crucial role in preserving natural ecosystems and biodiversity. By detecting and responding to wildfires promptly, these systems help protect sensitive habitats, endangered species, and ecological balance. Minimizing the extent and intensity of wildfires can also reduce the long-term environmental impact and promote faster recovery of affected areas.
- **Infrastructure Protection**: Early detection allows for timely evacuation orders and implementation of protective measures for critical infrastructure, such as power plants, communication networks, and transportation systems. By safeguarding infrastructure from wildfires, the system helps pr, financial losses, and potential hazards to public safety.

CHAPTER 2

REQUIREMENT SPECIFICATION AND ANALYSIS

Requirement analysis is the process of creating a score for a systems effort. The result is neither coordinated nor integrated and often simply does not work. Requirement analysis is critical to the success or failure of a systems or software project. The requirements should be documented, actionable, measurable, testable, traceable, related to identified business needs or opportunities, and defined to a level of detail sufficient for system design

2.1 FUNCTIONAL REQUIREMENTS

In software engineering and systems engineering, a functional requirement defines a function of a system or its component, where a function is described as a specification of behavior betweenoutputs and inputs.

2.1.1 Connectivity

Since the WEDS device will be placed in the corners of the forest, the communication protocol needs to be reliable, low power, and have a long range. Because of these requirements, the best communications protocol would sending message over telegram.

2.1.2 Real-Time Data Capture

The system should be capable of capturing data from sensors and cameras in real-time. This includes collecting visual data from cameras and environmental data from sensors such as temperature, humidity, and smoke particle levels..

2.1.3Monitoring and Continuous Operation

The system should continuously monitor the environment and update the analysis in real-time. It should operate reliably and remain active at all times to ensure ongoing detection and response to wildfires.

2.1.4 User requirement

The user must create a chat id using the telegram bot. this id must be registered in the code. This enables the user to access the chat bot where the alert messages are sent when a fire is detected in rage of the equipment.

2.1.5 User Interface and Monitoring Dashboard

The system should provide a user-friendly interface and a monitoring dashboard that allows users to view real-time data, system status, and alerts. The interface should be intuitive and provide clear visualizations of the collected data.

2.2 NON-FUNCTIONAL REQUIREMENTS

In systems engineering and requirements engineering, a nonfunctional requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather thanspecific behavior.

2.2.1 Reliability

The system provides correct output as per input specification. The user can rely on the output provided by the system. The reliability of a wildfire early detection system can vary depending on various factors, including the technology used, system design, environmental conditions, and maintenance practices.

2.2.2 Performance

The WEDS device provide instant results using fast connectivity available to the device with the help of components such as raspberry pi and chat bot services provided by the telegram. The performance of a wildfire early detection system refers to its ability to effectively and efficiently detect wildfires in their early stages. It encompasses various aspects that contribute to the system's overall effectiveness in detecting and alerting about wildfires. Evaluation and continuous monitoring of these performance factors can help assess the effectiveness of a wildfire early detection system and identify areas for improvement. Regular testing, maintenance, and updates are important to ensure optimal performance and enhance the system's ability to mitigate the impact of wildfires.

2.2.3 Usability

The system can be implemented in suitable forest area where it can cover wide range of area and provide output. The usability of a wildfire early detection system refers to its ease of use, user-friendliness, and the overall user experience associated with operating and interacting with the system. It encompasses several key aspects that contribute to the system's usability. Considering and prioritizing usability factors during the design, development, and deployment of a wildfire early detection system can enhance user satisfaction, adoption, and overall system effectiveness. Regular user feedback and iterative improvements contribute to continuously improving the usability of the system over time.

2.2.4 Scalability

This system can able to provide more features such as integrating with existing system, The system can also be able to upgrade with additional functionality. Scalability of a wildfire early detection system refers to its ability to accommodate increased demands and expand its capabilities as the monitoring requirements and scope of the system grow. It involves the system's capacity to handle larger volumes of data, increased sensor coverage, and the ability to adapt to evolving needs. . Scalability is a critical aspect of a wildfire early detection system, as it allows the system to adapt to changing needs and accommodate growing monitoring requirements. By designing and implementing a scalable system, organizations can effectively expand their wildfire detection capabilities and enhance their ability to detect and respond to wildfires in a larger geographic area.

2.2.5 Maintainability

The completed WEDS device’s lifecycle must be as long as possible. Since the device operates from batteries and does not require power from the grid, it must be energy efficient. Maintainability of a wildfire early detection system refers to its ability to be easily and effectively maintained, serviced, and upgraded over time. It involves various practices and considerations that ensure the system remains operational, reliable, and up-to-date. By focusing on maintainability, organizations can ensure that their wildfire early detection system remains operational, reliable, and effective over its lifecycle. Regular maintenance activities, adherence to best practices, and continuous improvement efforts

2.3 SOFTWARE REQUIREMENTS

The project requires the following software to run.

- Operating System: 64-bit Microsoft® Windows® 8/10 or MacOS 10
- Language: Python
- Text editor: PyCharm

Python Packages:

2.3.1 Software Description

Software is a set of instructions, data or programs used to operate computers and execute specific tasks. It is the opposite of hardware, which describes the physical aspects of a computer. Software is a generic term used to refer to applications, scripts and programs that run on a device.

2.3.2 Python

Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language. It was created by Guido van Rossum during 1985- 1990. Like Perl, Python source code is also available under the GNU General Public License (GPL). This tutorial gives enough understanding on Python programming language.

Python is a popular programming language. It was created in 1991 by Guido van Rossum. It is used for:

- web development (server-side),
- software development,

- mathematics,
- System scripting.

Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc). Python has a simple syntax similar to the English language. Python has syntax that allows developers to write programs with fewer lines than some other programming languages.

Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick. Python can be treated in a procedural way, an object-orientated way or a functional way.

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

- **Python is Interpreted** − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- **Python is Interactive** − you can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
- **Python is Object-Oriented** − Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
- **Python is a Beginner's Language** − Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

Python Features

- **Easy-to-learn** − Python has few keywords, simple structure, and a clearly definedsyntax. This allows the student to pick up the language quickly.
- **Easy-to-read** − Python code is more clearly defined and visible to the eyes.
- **Easy-to-maintain** − Python's source code is fairly easy-to-maintain.
- **A broad standard library** − Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.

- **Interactive Mode** − Python has support for an interactive mode which allows interactivetesting and debugging of snippets of code.
- **Portable** − Python can run on a wide variety of hardware platforms and has the sameinterface on all platforms.
- **Scalable** − Python provides a better structure and support for large programs than shellscripting.
- **Databases** − Python provides interfaces to all major commercial databases.
- **GUI Programming** − Python supports GUI applications that can be created and portedto many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix

2.3.3 PyCharm

PyCharm is an integrated development environment (IDE) specifically designed for Python development. It is developed by JetBrains and provides a wide range of tools and features to help developers write, debug, and test Python code more efficiently.

PyCharm offers a wide range of features to enhance productivity and facilitate Python development. Some notable Pycharm uses include:

- Code editor with syntax highlighting, code completion, and code refactoring capabilities.
- Intelligent code analysis and error detection.
- Debugger for debugging Python code.
- Git integration for version control.
- Virtualenv and Conda support for managing Python environments.
- Integration with popular Python frameworks, such as Django, Flask, and Pyramid.
- Support for web development technologies like HTML, CSS, and JavaScript.
- Database integration for working with various database systems.
- Unit testing and test coverage tools.
- Support for scientific libraries and tools like NumPy, Pandas, and Matplotlib.

Here are some key features of PyCharm:

- **Code Editor:** PyCharm offers a powerful code editor with features like syntax highlighting, code completion, code refactoring, and smart indentation. It also supports various code styles and provides integration with version control systems like Git.
- **Code Navigation:** PyCharm allows you to navigate through your code easily. It provides features like Go to Definition, Find Usages, and a powerful search tool to locate classes, functions, and variables within your project.
- **Debugger:** PyCharm includes a built-in debugger that helps you identify and fix issues in your Python code. You can set breakpoints, step through the code, inspect variables, and analyze the program flow.
- **Testing:** PyCharm has excellent support for unit testing and test-driven development (TDD). It allows you to create and run tests, view test results, and integrate with popular testing frameworks like pytest and unittest.

2.3.4 OpenCV

OpenCV (Open-Source Computer Vision Library) is an open-source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products.

- **Purpose**: OpenCV is primarily used for computer vision applications, which involve extracting useful information from visual data (images or videos). It enables developers to perform tasks such as image and video manipulation, feature detection, pattern recognition, object tracking, and more.
- **Cross-platform Compatibility**: OpenCV is designed to be cross-platform and works on various operating systems, including Windows, macOS, Linux, Android, and iOS. This allows developers to build applications that run on different platforms without major modifications.
- **Programming Language Support**: OpenCV is primarily implemented in C++ but also provides interfaces for other programming languages, including Python, Java, and

MATLAB. This enables developers to leverage OpenCV functionalities in their preferred programming language.

- **Image and Video Processing:** OpenCV provides a wide range of functions for image and video processing tasks, including reading and writing image files, resizing and cropping images, applying filters and transformations, handling color spaces, and performing arithmetic operations on images. It also supports video input/output operations, allowing developers to process video frames in real-time or from recorded videos.
- **Feature Detection and Tracking**: OpenCV offers various algorithms and techniques for feature detection and tracking. It includes methods for detecting keypoints, extracting descriptors, and matching features between images. These features are essential for tasks such as object recognition, image stitching, motion tracking, and augmented reality applications.
- **Object Detection and Recognition**: OpenCV provides pre-trained models and algorithms for object detection and recognition, such as Haar cascades and deep learning-based models. These tools enable developers to detect and identify objects in images or video streams, including faces, pedestrians, vehicles, and other custom objects.
- **Community and Support:** OpenCV has a large and active community of developers, researchers, and enthusiasts who contribute to its development and provide support. The community offers extensive documentation, tutorials, forums, and code examples, making it easier for users to learn and utilize OpenCV effectively.
- **Machine Learning Integration**: OpenCV integrates with machine learning libraries, such as TensorFlow and PyTorch, allowing developers to combine computer vision with machine learning techniques. This enables tasks such as image classification, object detection, semantic segmentation, and more, using trained models.

OpenCV has become a widely adopted library in the field of computer vision due to its extensive capabilities, cross-platform compatibility, and community support. It is utilized in various industries, including robotics, autonomous vehicles, healthcare, security, and entertainment, to address a wide range of computer vision challenges.

2.3 HARDWARE REQUIREMENTS

- System: Pentium Dual Core.
- Microcontroller: Raspberry Pi
- Intel Core processor (i3 above)
- RAM: 4GB
- Hard Disk: 512 GB

2.4.1 Raspberry Pi

Raspberry Pi is a series of small, affordable single-board computers (SBCs) developed by the Raspberry Pi Foundation. The Raspberry Pi boards are designed to promote computer science education and provide a platform for DIY projects and embedded systems. Here are some key details about Raspberry Pi. Raspberry Pi has gained popularity for its versatility, affordability, and accessibility. Its small size, low power consumption, and extensive community support make it an excellent choice for beginners and advanced users alike who are interested in exploring computer science, electronics, and DIY projects. The main method of detecting wildfires is using a camera paired with computer vision algorithms to recognize fires. To achieve this, a powerful enough computer is needed to compute and run the computer vision code. The system also needs to be low powered since this device will be placed in remote areas for long periods of time. The cheapest, most effective computer on the market is the Raspberry Pi Zero W. The Raspberry Pi Zero is a single board computer with a 1GHz single-core ARMv6 CPU with 512MB of RAM. The Raspberry Pi Zero consumes 150mA, however can consume as low as 120mA if the HDMI and LEDs are turned off . The Raspberry Pi that is used to control the WEDS module runs on the Raspberry Pi OS which can be downloaded onto an SD card from raspberrypi.org/software/. 16 We wrote our computer vision algorithms in Python for its simplicity. We used the OpenCV library and trained our models on an external machine with a much faster process.

2.4.2 Camera

The Raspberry Pi Camera is a small camera module specifically designed for Raspberry Pi boards. It provides a cost-effective solution for capturing images and videos directly with the

Raspberry Pi, making it a popular choice for various applications. The WEDS system uses the Raspberry Pi Camera Module V2 to maintain a video feed of the surrounding area. The frames of the video serve as input to the system's computer vision algorithms. The module mounts to the body of the WED's enclosure and connects to the Raspberry Pi Zero W's CSI port via a 15cm ribbon cable. The camera features the Sony IMX219 sensor with a 3.68 x 2.76 mm sensor area [26]. It captures 1080p30 video but is also capable of 720p60 and 640 x 480p60/90. The Raspberry Pi Camera offers an affordable and convenient solution for capturing images and videos directly with Raspberry Pi boards. Its integration with the Raspberry Pi ecosystem and community support make it a popular choice for various projects and applications involving visual data capture and analysis.

2.4.3 Flame sensor

A flame sensor is a device used to detect the presence or absence of a flame in various heating systems and appliances. It serves as a safety feature by ensuring that the burner flame is present and operating correctly. The primary function of a flame sensor is to monitor the flame and provide feedback to the control system of the heating device. It helps ensure that the fuel is being burned efficiently and that any potential safety hazards, such as gas leaks or incomplete combustion, are prevented. Flame sensors are typically integrated into the larger Wildfire Early Detection System, which includes cameras, other sensors, and a central processing unit. The flame sensor's output is fed into the central processing unit, where it is analyzed along with data from other sensors and cameras to make informed decisions about the presence of a wildfire.

2.4.4 Humidity and Temperature sensor

A humidity and temperature sensor, also known as a hydrothermal meter or combined sensor, is a device that measures both the ambient temperature and relative humidity in the surrounding environment. It provides valuable data for various applications, including weather monitoring, HVAC systems, indoor climate control, and industrial processes. Here are some key details about humidity and temperature sensors. The DHT22 sensor, also known as the AM2302, is a popular digital temperature and humidity sensor.

CHAPTER 3

SYSTEM DESIGN

System design is the process of defining the architecture, components, modules, interfaces and data for a system to satisfy specified requirements. It is the application of systems theory to product development. There is some overlap with the disciplines of system analysis, system architecture and systems engineering

3.1 OVERVIEW OF THE PROPOSED SYSTEM

The Wildfire Early Detection System (WEDS) is a device that monitors conditions in areas prone to wildfires. Each unit contains a flame sensor and a camera to detect a fire. Based on the sensors and the images received by the camera, the Raspberry Pi computer can determine if a fire is nearby. Once a fire has been detected, a signal is sent to the main central hub where it notifies first responders of the fire's location. Because of the limited communication range, multiple devices would communicate with each other and would daisy chain back to a central hub.

3.2 SYSTEM ARCHITECTURE

The Fig 3.1 explains the architecture of the WEDS system.

1. When forest catches fire the WEDS system will detect the fire using sensors and camera.
2. The detected message from WEDS system is displayed on the monitor.
3. The WEDS system sends the alert message to a telegram channel.

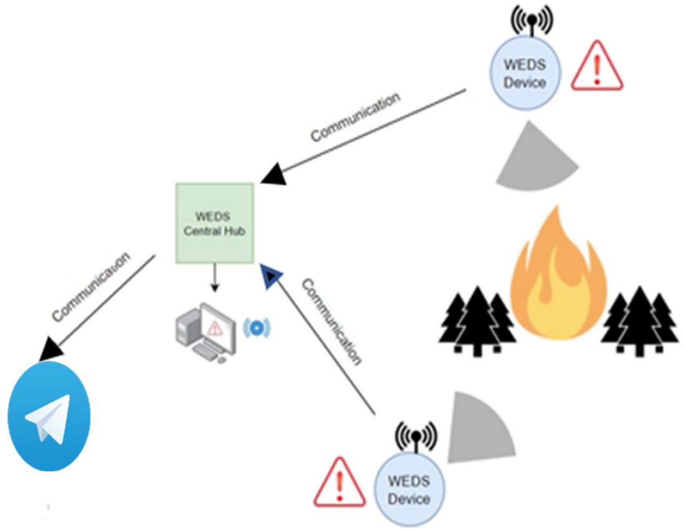


Fig 3.1: System Architecture

3.3 SYSTEM MODEL

The Fig 3.2 is a system model that helps in understanding and analyzing the system behavior, components and interaction.

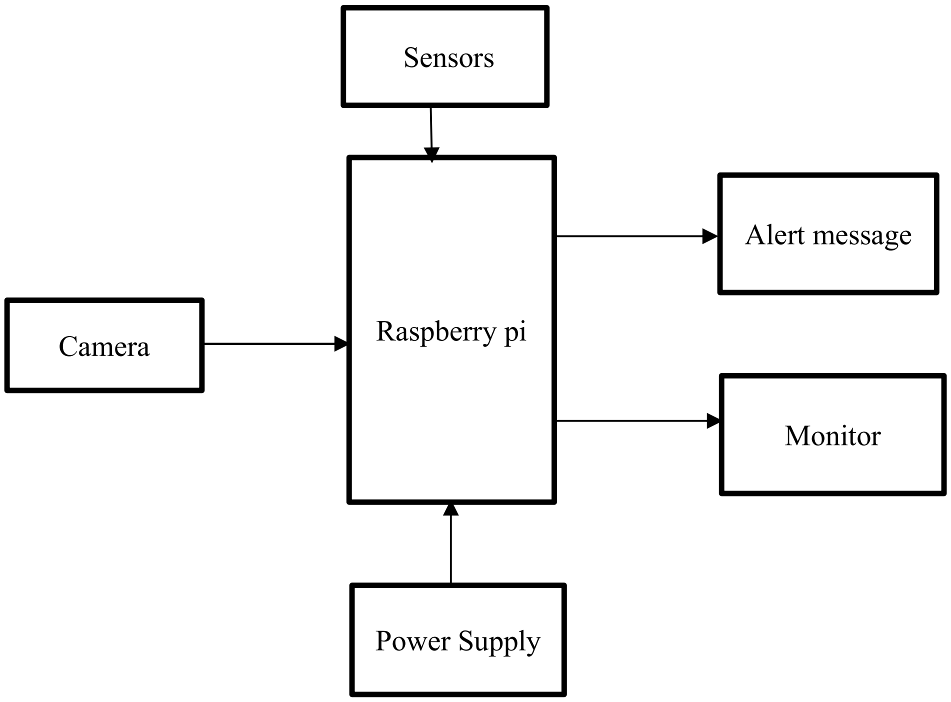


Fig 3.2: System Model

Raspberry Pi: The Raspberry Pi Zero is a single board computer with a 1GHz single-core ARMv6 CPU with 512MB of RAM. The Raspberry Pi Zero consumes 150mA, however can consume as low as 120mA if the HDMI and LEDs are turned off [13]. The Raspberry Pi that is used to control the WEDS module runs on the Raspberry Pi OS which can be downloaded onto an SD card from raspberrypi.org/software/.

Camera: Since imaging cameras can "see" through darkness or smoke, they allow firefighters to quickly find the seat of a structure fire, or see the heat signature of visually obscured victims. They can be used to search for victims outdoors on a cool night, spot smoldering fires inside a wall, or detect overheating electrical wiring

Power Supply: At the Pi Hut we offer a range of different micro-development boards, which all require slightly different power sources! Some are very stringent in their required input; for example, the latest Raspberry Pi recommends 5V @ 2A as a minimum for stability, but some are more flexible, the Arduino can accept a range of voltage inputs (6 – 20V), and regulates this to desired level internally on board. The Raspberry Pi can function on lower current power supplies e.g., 5V @ 1A.

3.4 FLOW CHART DIAGRAM

A Flowchart is a picture of the separate steps of a process in sequential order. It depicts a process, system or computer algorithm.

The Fig 3.3 shows how the system starts from taking the inputs from different sensors and checks for the threshold value, if it is above threshold then the fire system gets alerted

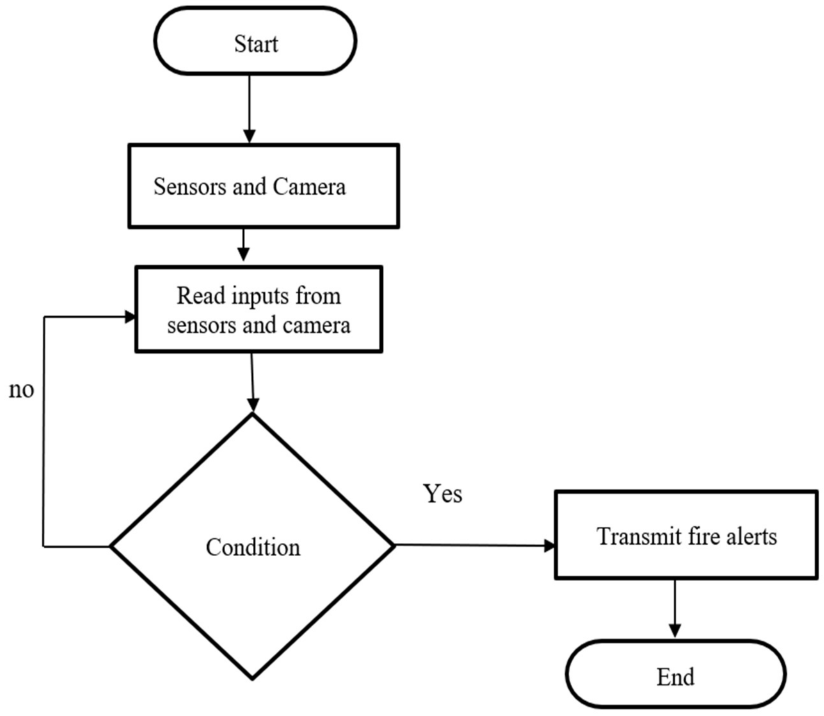


Fig 3.3: Dataflow diagram

3.5 USE CASE DIAGRAM

A use case is a set of scenarios that describing an interaction between a source and a destination. The two main components of a use case diagram are use cases and actors. It displays the relationship among them. A use case diagram can identify the different types of users of a system and the different use cases and will often be accompanied by other types of diagrams as well. The use cases are represented by either circles or ellipses. The Use Case diagram for User is shown in Fig 3.4.

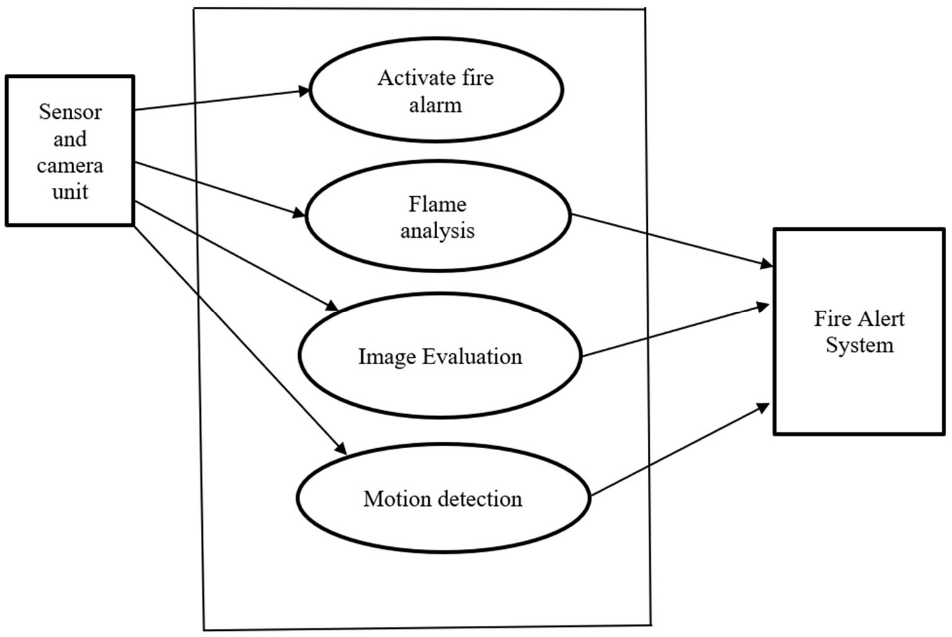


Fig 3.4: Use Case Diagram

3.6 SEQUENCE DIAGRAM

Sequence diagram depicts the processes involved and the sequence of messages exchanged between the processes needed to carry out the functionality.

In the Fig 3.5 the processes performed by the design and the display are shown and different actions for different processes are also shown. It begins with the sensors capturing the data from the environment and sending it to the system and the system displays the received information on the display screen.

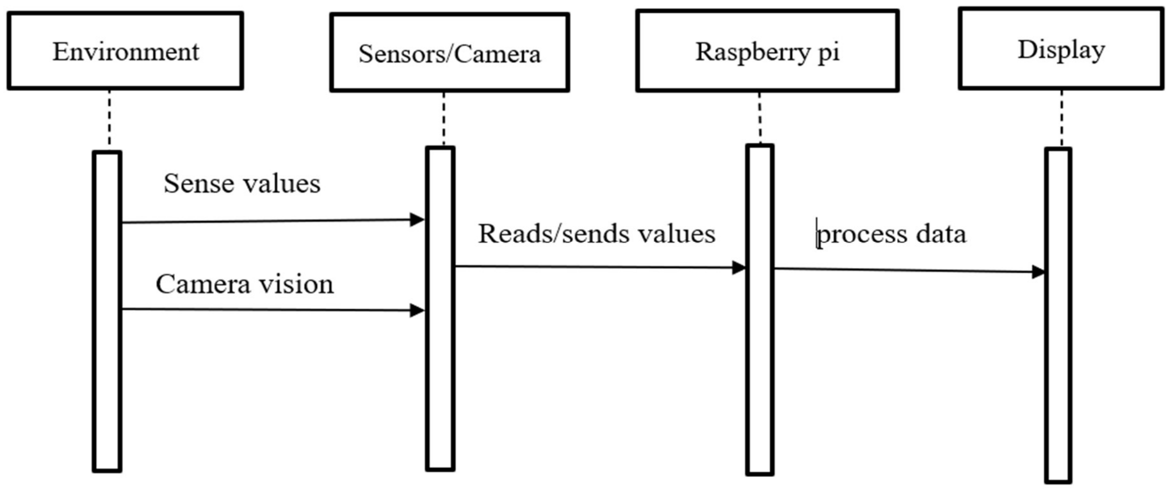


Fig 3.5: Sequence Diagram

CHAPTER 4

SYSTEM IMPLEMENTATION

A crucial phase in the system development life cycle is successful implementation of new system design. Implementations simply mean converting new system design into operation. The term implementation has different meanings, ranging from the conversion of a basic application to a complete replacement of computer system. Implementation is used here to mean the process of converting a new or revised system design into an operational one.

Implementing a Wildfire Early Detection System involves several fundamental components and processes. It begins with the selection and installation of hardware, including cameras and sensors, in strategic locations that are prone to wildfires. These hardware components capture real-time data from the environment, such as visual information and environmental variables like temperature and humidity. The captured data serves as the foundation for wildfire detection and analysis.

4.1 IMPLEMENTATION

There are several steps included in the implementation of the project

4.1.1 Initial project setup

Obtain the necessary hardware:

- Raspberry Pi board (any model, such as Raspberry Pi 4 or Raspberry Pi 3)
- MicroSD card (at least 8GB, preferably class 10 or higher)
- Power supply (compatible with your Raspberry Pi model)
- Keyboard and mouse (USB or wireless)
- HDMI cable
- Monitor or TV with HDMI input

Prepare the operating system:

Download the desired operating system for the Raspberry Pi. The official operating system is called Raspberry Pi OS, formerly known as Raspbian.

Write the operating system image to the microSD card:

- Insert the microSD card into your computer using an SD card adapter.
- Format the microSD card if necessary (use a tool like SD Card Formatter).

Set up the Raspberry Pi:

- Insert the microSD card into the Raspberry Pi.
- Connect the Raspberry Pi to a monitor or TV using the HDMI cable.
- Connect a keyboard and mouse to the Raspberry Pi.
- Connect the power supply to the Raspberry Pi to turn it on.

Perform the initial setup:

The Raspberry Pi should boot into the operating system. Follow the on-screen instructions to set up the system. This typically includes selecting your language, changing the default password, and expanding the file system.

Once the setup is complete, the Raspberry Pi will reboot.

Explore and configure:

After rebooting, you can explore the Raspberry Pi OS and configure it according to your needs. You can install additional software, connect to the internet, configure Wi-Fi, and more.

4.1.2 Camera and Raspberry Pi Interaction

Raspberry Pi has native support for camera modules, which allows you to easily interact with cameras and capture images or videos

Connect the camera module:

- Ensure your Raspberry Pi is powered off.
- Identify the camera connector on the Raspberry Pi board. It's a flat, ribbon-like connector usually located near the HDMI port.
- Insert the camera ribbon cable with the metal contacts facing away from the Ethernet/USB ports.
- Push the plastic catch down to secure the ribbon cable.

Enable the camera module:

- Power on your Raspberry Pi.
- Open the terminal or SSH into your Raspberry Pi.
- Run the sudo raspi-config command to open the configuration tool.
- Navigate to "Interfacing Options" and select "Camera."
- Choose "Yes" to enable the camera interface.
- Select "Finish" to exit the configuration tool.
- Reboot your Raspberry Pi to apply the changes.

4.1.3 Raspberry Pi and Sensors Interaction

Interaction between Raspberry Pi and Flame Sensor: The flame sensor is connected to the Raspberry Pi's GPIO (General Purpose Input/Output) pins. The Raspberry Pi constantly monitors the state of the flame sensor by reading the digital signal from the sensor. When the sensor detects a flame, it sends a high or low signal to the Raspberry Pi, depending on its design. The Raspberry Pi can then capture this signal and trigger specific actions or alerts, such as sounding an alarm, sending notifications, or activating fire suppression systems.

Interaction between Raspberry Pi and Humidity Sensor: Similar to the flame sensor, the humidity sensor is connected to the Raspberry Pi's GPIO pins or other compatible interfaces (e.g., I2C or SPI). The humidity sensor periodically provides humidity readings to the Raspberry Pi. The Raspberry Pi can read these values and analyze them to determine the current humidity level. Based on the humidity readings, the Raspberry Pi can control devices like fans, humidifiers, or dehumidifiers to maintain the desired humidity level in a controlled environment.

Interaction between Raspberry Pi and Temperature Sensor: The interaction between a Raspberry Pi and a temperature sensor is essential for obtaining temperature data and incorporating it into various applications. The process begins with the physical connection between the Raspberry Pi and the temperature sensor, typically using appropriate wiring or interfaces.

Once connected, the Raspberry Pi initiates communication with the temperature sensor to acquire temperature readings. This can be done by sending signals or commands to the sensor, requesting temperature data. The sensor, in turn, detects the surrounding temperature and provides the measured values to the Raspberry Pi.

CHAPTER 5

SYSTEM TESTING AND EXPERIMENTAL RESULTS

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of tests. Each test type addresses a specific testing requirement.

5.1 TYPES OF TESTING

Performance Testing

This covers real-time and far more cumbersome aspects, such as load testing, streaming analytics, time-bound outputs, and timing analysis, to validate and ensure consistent performance of data reading, writing, and data retrieval. Performance testing is the practice of evaluating how a system performs in terms of responsiveness and stability under a particular workload. Different performance testing types measures according to benchmarks and standards. Performance testing gives developers the diagnostic information they need to eliminate bottlenecks.

Scalability Testing

This includes the testing of all functional and non-functional use cases to ascertain whether the system is easy to scale to accommodate future upgrades. A scalability test is a type of load testing that measures the application's ability to scale up or down as a reaction to an increase in the number of users. In other words, it tests how the system is going to perform during a sudden spike or fall of user request loads. It is also referred to as performance testing, as such, it is focused on the behavior of the application when deployed to a larger system or tested under excess load. In Software Engineering, Scalability Testing is to measure at what point the application stops scaling and identify the reason behind it.

Regulatory Testing

This testing determines the compliance of applications with privacy regulations. Compliance testing or conformance testing is a process conducted to determine if a process, product, or service meets a set of requirements, based on internal or external standards or regulations.

Integration Testing

Integration testing focuses on testing the interaction and compatibility between different components or subsystems of the wildfire early detection system. It ensures that the various components work seamlessly together. This testing may involve simulating data flows, communication between sensors and the central processing unit, and integration with external systems, such as weather monitoring services or emergency response systems.

Security Testing

Handling an onslaught of data is fundamental operation, and therefore, enterprises must conduct security testing to eliminate vulnerabilities and maintain the integrity of data protection, encryption/decryption, device identification and authentication among more. Security testing is a process of testing devices to find security vulnerabilities that hackers could exploit to access your network, modify your data, or steal your information. This can lead to significant financial losses, identity theft, and damage to the reputation of your business .

Functional Testing

This examines the qualitative and quantitative functional deliverability of deployed IoT applications in the actual conditions. Aspects, like network size, environment conditions, and topologies, are put to test. The key for IoT functional testing is a simulated real-life environment using real devices or simulators to test those core components. IoT technologies continue to emerge rapidly, and every use case differs as far as what would be considered the functional core components. Each device is and will be unique.

Reliability Testing

Reliability testing aims to assess the system's ability to perform consistently and accurately over an extended period. It involves subjecting the system to various scenarios and conditions to identify any potential weaknesses or failures. For a wildfire early detection system, reliability testing may involve long-term monitoring, exposure to extreme weather conditions, and evaluating the system's ability to handle false positives or false negatives.

Usability Testing

Usability testing assesses the system's user-friendliness and ease of use. It involves evaluating the user interface, system navigation, and overall user experience. For a wildfire early detection system, usability testing may involve testing the interface used by operators or emergency responders to monitor the system, interpret data, and initiate appropriate actions.

Field Testing

Field testing involves deploying the wildfire early detection system in real-world conditions to evaluate its performance and effectiveness in detecting and responding to wildfires. It may involve setting up sensors in various locations, conducting controlled burns or fire simulations, and assessing the system's ability to provide timely and accurate alerts in different terrain and environmental conditions.

5.2 RESULT

By using multiple sensors and cameras and analyzing data in real-time, the system can detect wildfires more accurately and faster than traditional methods. The use of threshold values and rules for data analysis can minimize false alarms and improve the reliability of the system. The system can provide real-time monitoring of wildfires and transmit alerts to relevant authorities and stakeholders, enabling rapid response and mitigation measures.

CHAPTER 6

SCREENSHOTS

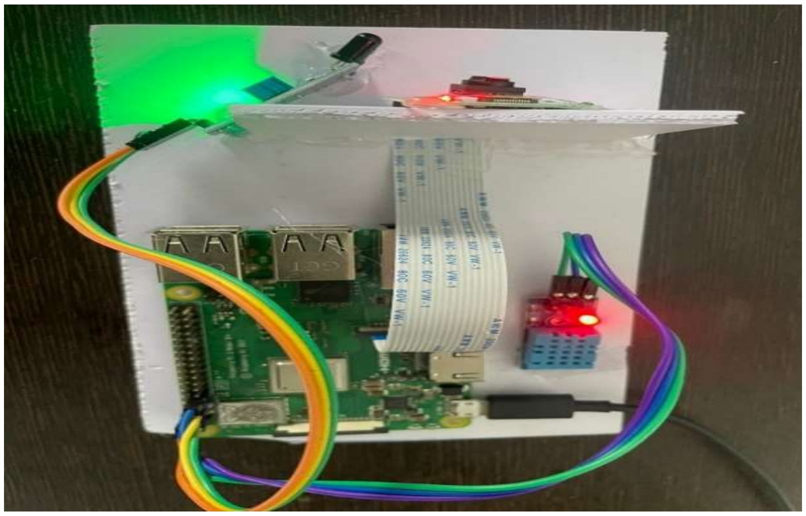


Fig 6.1: Prototype model

Fig 6.1 shows the prototype model containing raspberry pi, rapberry pi camera, flame sensor humidity and temperature sensor, micro USB cable, connecting wires.

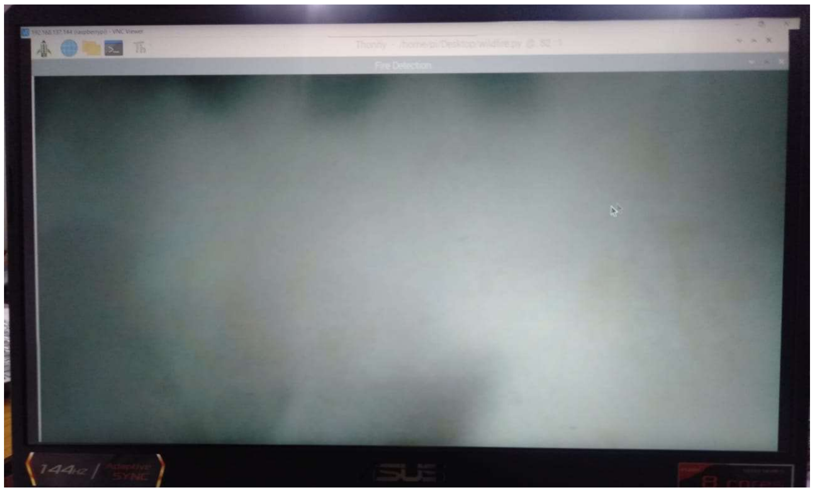


Fig 6.2: Before Fire detected

Fig 6.2 shows the camera vision before detecting fire.

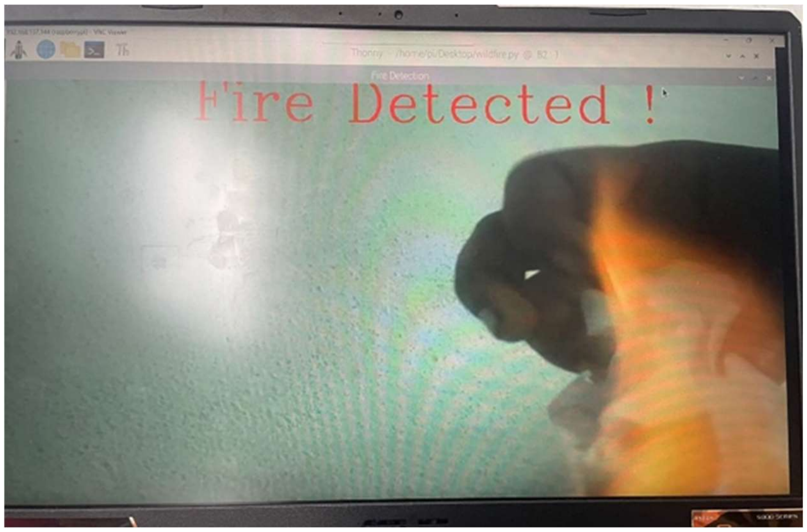


Fig 6.3: Fire detected

Fig 6.3 shows the detected fire by the camera and other sensors along with the on screen alert message.

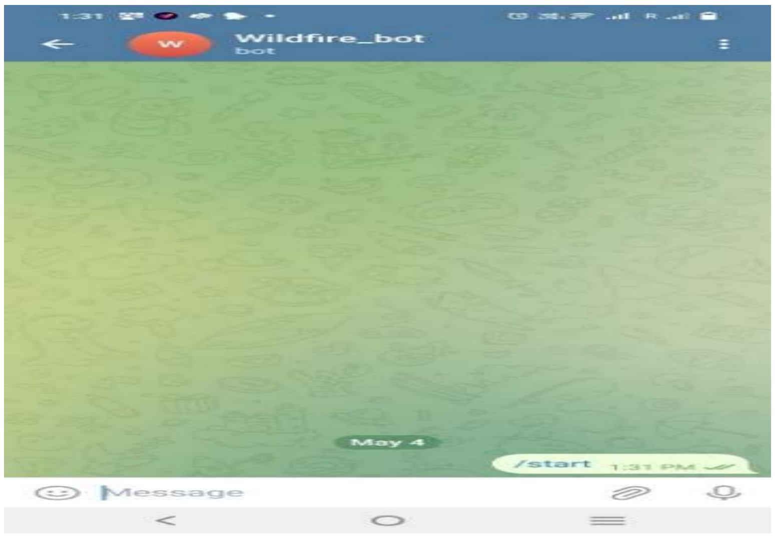
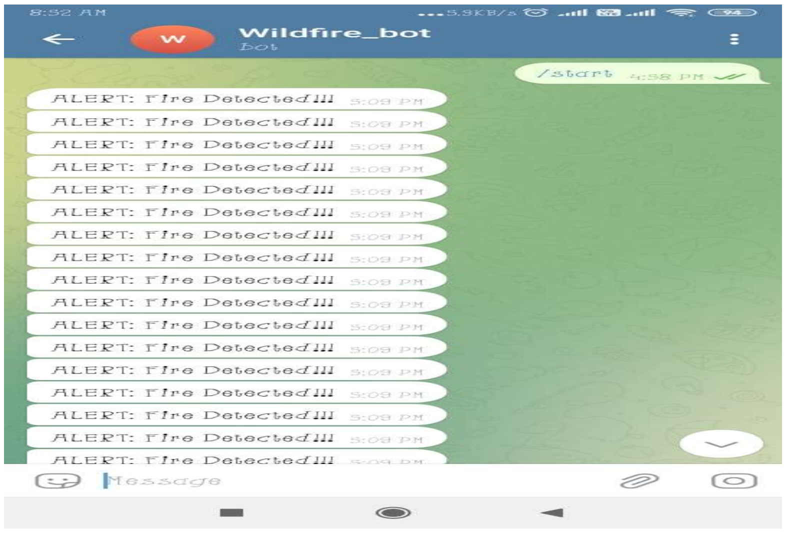


Fig 6.4: Telegram channel

Fig 6.4 shows wildfire bot responsible for sending alert message to the user through their chat ids.



/start 4:58 PM

ALERT: Fire Detected!!! 5:09 PM
ALERT: Fire Detected!!! 5:09 PM
ALERT: Fire Detected!!! 5:09 PM
ALERT: Fire Detected!!! 5:09 PM
ALERT: Fire Detected!!! 5:09 PM
ALERT: Fire Detected!!! 5:09 PM
ALERT: Fire Detected!!! 5:09 PM
ALERT: Fire Detected!!! 5:09 PM
ALERT: Fire Detected!!! 5:09 PM
ALERT: Fire Detected!!! 5:09 PM
ALERT: Fire Detected!!! 5:09 PM
ALERT: Fire Detected!!! 5:09 PM
ALERT: Fire Detected!!! 5:09 PM
ALERT: Fire Detected!!! 5:09 PM
ALERT: Fire Detected!!! 5:09 PM
ALERT: Fire Detected!!! 5:09 PM
ALERT: Fire Detected!!!

Fig 6.5: After fire detection

Fig 6.5 shows the alert messages in telegram channel when the fire is detected.

\

CHAPTER 7

CONCLUSION AND FUTUREENHANCEMENT

7.1 CONCLUSION

Use of machine learning algorithms can significantly improve the hiring process by automating the initial screening of candidates. This project, Fire Detection System has been developed using sensors and camera. This system has the ability to apply image processing techniques to detect fire. This system can be used to monitor fire and has achieved 90% accuracy for single camera. In conclusion, a wildfire early detection project utilizing Raspberry Pi, cameras, and sensors offers a cost-effective and versatile solution for monitoring and detecting wildfires in real-time. By integrating these components, the system can capture visual data, environmental measurements, and process them to provide early warnings and prompt responses to potential fire incidents. The system works on real time, as it extracts frames in every 2 seconds, it provides continuous monitoring. This system has high efficiency as it has incorporated techniques of Area detection, Color detection. The different parameters like threshold value, blind-spots will be handled properly in our future research. Thus application of proposed fire detection system gives us a better system performance in term of less false alarm and thus a higher system performance is achieved.

7.2 FUTURE ENHANCEMENT

For further accuracy use of Neural Networks for decision making can be made and GSM module can also be implemented for sending SMS to nearby fire station in case of severe fire. Water sprinklers can also be incorporated. By research and analysis, the efficiency of the proposed Fire detection system can be increased. The margin of false alarms can be reduced even further by developing algorithms to eliminate the detection of red colored cloth as fire. By proper analysis, suitable location height and length for camera installment can be decided, in order to remove blind-spot areas

REFERENCES

[1] Y. Hakan Habiboglu, O. Gunay and A. Enis Cetin, "Real-time wildfire detection using correlation descriptors," 2011 19th European Signal Processing Conference, Barcelona, Spain, 2011, pp. 894-898.

[2] T. Çelik, H. Özkaramanlı and H. Demirel, "Fire and smoke detection without sensors: Image processing based approach," 2007 15th European Signal Processing Conference, Poznan, Poland, 2007, pp. 1794-1798.

[3] Zhao. Y, Zhang. H, Zhang. X, Qian. W, Wildfire Smoke Detection Based on Depthwise Separable Convolutions and Target-Awareness. Preprints.org 2020, 2020040027.

[4] S. Kundu, V. Mahor and R. Gupta, "A Highly Accurate Fire Detection Method Using Discriminate Method," 2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Bangalore, India, 2018, pp. 1184-1189, doi: 10.1109/ICACCI.2018.8554799.

[5] H. Afzaal and N. A. Zafar, "Robot-based forest fire detection and extinguishing model," 2016 2nd International Conference on Robotics and Artificial Intelligence (ICRAI), Rawalpindi, Pakistan, 2016, pp. 112-117, doi: 10.1109/ICRAI.2016.7791238.