FOREST FIRE DETECTION AND NOTIFICATION SYSTEM

This project report is submitted to

Yeshwantrao Chavan College of Engineering

(An Autonomous Institution Affiliated to Rashtrasant Tukdoji Maharaj Nagpur University)

In partial fulfillment of the requirement

For the award of the degree

Of

Bachelor of Technology in Electronics & Telecommunication Engineering

By

ANISH MENDHE

AYUSH MISHRA

NAKSHATRA SHINDEKAR

YASH KANGALE

Under the guidance of

GUIDE: DR. B.Y. MASRAM

DEPARTMENT OF ELECTRONICS & TELECOMMUNICATION ENGINEERING

Nagar Yuwak Shikshan Sanstha's

YESHWANTRAO CHAVAN COLLEGE OF ENGINEERING,

(An autonomous institution affiliated to Rashtrasant Tukdoji Maharaj Nagpur University, Nagpur)

NAGPUR – 441 110

2023-2024

CERTIFICATE OF APPROVAL

This is to Certify that the project report entitled **"FOREST FIRE DETECTION AND NOTIFICATION SYSTEM**" has been successfully completed by **Anish Mendhe, Ayush Mishra, Nakshatra Shindekar, Yash Kangale** under the guidance of **Dr. B. Y. Masram** in recognition to the partial fulfilment for the award of the degree of Bachelor of Technology in Electronics & Telecommunication Engineering**, Yeshwantrao Chavan College of Engineering** ***(An Autonomous Institution Affiliated to Rashtrasant Tukdoji Maharaj Nagpur University)***

Signature	*Signature*
Dr. B. Y. Masram	**Dr. M. S. Narlawar**
(Project Guide)	**(HOD, ET Dept.)**

Signature of External Examiner

Name: -

Date of Examination:

DECLARATION

We hereby declare that

a. The work contained in this project has been done by us under the guidance of my supervisor(s).
b. The work has not been submitted to any other Institute for any degree or diploma.
c. We have followed the guidelines provided by the Institute in preparing the project report.
d. We have conformed to the norms and guidelines given in the Ethical Code of Conduct of the Institute.
e. Whenever we have used materials (data, theoretical analysis, figures, and text) from other sources, we have given due credit to them by citing them in the text of the report and giving their details in the references. Further, we have taken permission from the copyright owners of the sources, whenever necessary.

Name of the Students Signature

1. Anish Mendhe

2. Ayush Mishra

3. Nakshatra Shindekar

4. Yash Kangale

ACKNOWLEDGEMENT

We would like to express our deep gratitude to our project guide Dr. B. Y. Masram, Professor, Department of Electronics and Telecommunication Engineering, YCCE, for her guidance unsurpassed knowledge, and immense encouragement. We are grateful to Dr. M. S. Narlawar, Head of the Department, of Electronics and Telecommunication Engineering, for providing us with the required facilities for the completion of the project work.

We are very much thankful to our Principal Dr. U. P. Waghe, YCCE, Nagpur, for his encouragement and cooperation in carrying out this work.

We express our thanks to all the teaching faculty of our department, whose suggestions during reviews helped us in the accomplishment of our project. We would like to thank all non-teaching staff of the Department for providing great assistance in the accomplishment of our project.

CONTENTS

Title **Page No.**

Title **Page No.**

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

As the frequency and severity of forest fires continues to escalate, it is essential that we take swift action to address their destructive impact. This project presents a cutting-edge 2-step Forest Fire Detection System that is specifically designed for both efficiency and rapid response. In its first stage, the system utilizes a Wireless Sensor Network of MQ-135 sensor, using NodeMCU as the main wireless communication module, to serve as the initial line of defence. These sensors diligently monitor the surrounding environment for any presence of highly flammable gases or smoke, and promptly signal the Raspberry pi to activate the camera to activate at the slightest indication of a potential fire. This camera, powered by the advanced YOLOv8 technology, specializes in the detection of flames and is able to accurately pinpoint any fire incidents with precision. The entire process is seamlessly integrated with a Raspberry Pi, which, upon receiving confirmation of a fire, immediately triggers the dispatch of an email notification for prompt action. This project serves as a testament to the potential of combining innovative technologies to address environmental challenges and contribute to the development of efficient and proactive fire detection systems.

CHAPTER - 1

INTRODUCTION

1. INTRODUCTION

1.1 OVERVIEW

Forest fires pose a significant global challenge, particularly during the summer months. According to the Forest Survey of India, an alarming number of over 300,000 forest fires were detected between November 2020 and June 2021 alone. Moreover, the Indian State of Forest Report (ISFR) for 2019 highlights that approximately 36% of the country's forest cover is susceptible to wildfires, with around 54.4% of forests in India being exposed to occasional fires. This escalating trend in forest fires has devastating consequences, including the destruction of vegetation, habitats for wildlife, and soil quality degradation. Additionally, the emission of toxic gases and CO2 exacerbates health risks and contributes to global warming.

Despite the awareness of these issues, effectively combating forest fires remains a challenge. Previous literature has identified several key challenges, including limitations in early detection methods and the delayed response of fire departments due to reliance on manual monitoring systems. Traditional approaches often fail to promptly detect fires in their initial stages, leading to heightened damage and difficulty in containment. Motivated by the urgent need for improved forest fire detection and response systems, this project presents a novel two-stage approach utilizing advanced sensor and camera technologies. The objective of this research is to enhance the efficiency and rapidity of fire detection, thereby reducing the extent of damage caused by wildfires. By integrating cutting-edge technologies such as the Wireless Sensor Network of MQ135 sensor using ESP modules and YOLOv8-equipped camera with the Raspberry Pi platform, this system aims to revolutionize forest fire detection. This work contributes to the development of proactive solutions for addressing the escalating threat of forest fires. By leveraging innovative technologies, the proposed system offers a more efficient and timely approach to fire detection, potentially mitigating the destructive impact of wildfires. Furthermore, the seamless integration of sensors, cameras, and computing platforms demonstrates the potential of technological advancements in solving environmental challenges.

1.2 PROBLEM STATEMENT

A 2-step wildfire detection system that combines sensor network monitoring for real-time gas and smoke detection with camera-based image analysis to substantially decrease false positives. By utilizing MQ sensors in a Wireless Sensor Network (WSN) to detect forest fire emissions and conducting on-board image processing with the YOLOv8 Object Detection Model on a Raspberry Pi, the system effectively confirms wildfire presence, minimizing unnecessary alerts and optimizing resource utilization.

1.3 THESIS OBJECTIVES

- Develop a Two-Step Forest Fire Detection System: Design and implement a forest fire detection system that utilizes a combination of sensor networks and camera technology for efficient and rapid wildfire detection.
- Utilize Advanced Sensor Technologies: Employ MQ-135 gas sensors in a Wireless Sensor Network (WSN) to monitor the surrounding environment for the presence of highly flammable gases or particles emitted during forest fires.
- Integrate Camera-Based Image Analysis: Integrate a camera equipped with the YOLOv8 Object Detection Model to visually confirm the presence of fire incidents with high precision, minimizing false positives and optimizing resource utilization.
- Achieve Real-Time Detection and Response: Develop algorithms and workflows to enable real-time processing of sensor data and immediate triggering of alerts upon detection of potential fire incidents, facilitating rapid response by relevant authorities.
- Enhance Efficiency and Accuracy: Improve the efficiency and accuracy of forest fire detection by seamlessly integrating sensor networks, camera technology, and computing platforms such as Raspberry Pi, ensuring timely and accurate notification of fire incidents.
- Contribute to Environmental Protection: Contribute to the development of proactive solutions for addressing the escalating threat of forest fires, thereby mitigating the destructive impact on vegetation, wildlife habitats, and public health.

1.4 THESIS OVERVIEW

- **Chapter 1** has the overview, objective and problem statement of project.
- **Chapter 2** is the review of literature which includes the most recent ideas on the "FOREST FIRE DETECTION AND NOTIFICATION SYSTEM," which were examined and the best of which were included in the suggested plan to provide better results.
- **Chapter 3** lays out the actual Work Done in order to comprehend the entire assembly and operation of the proposed system. It includes every important idea in the suggested strategy, as well as how the various components function and how the programming is done with the use of outlines.
- **Chapter 4** comprises of Results and Discussion of the plans which were carried out, with conclusions drawn based on the Results.
- **Chapter 5** gives the report's conclusion based on the results examined and the scope of future work.
- **Chapter 6** has the listing of the Literature that were explored in the making of this plan.

CHAPTER-2

REVIEW OF LITERATURE

2. REVIEW OF LITERATURE

2.1 OVERVIEW

This chapter reviews the various literature survey done on Forest Fire Detection and Notification System. This chapter also discussed the work done by other researcher in the field of Forest Fire Detection Methods.

2.2 LITERATURE SURVEY

Traditionally watchtowers, patrol, aerial patrols, etc. were used to monitor the forest and prevent or detect wildfires. However, these techniques use constant human monitoring and hence can be tedious and exhausting, leaving room for error. Remote sensing is also popular method for detecting forest fire that does not require constant human monitoring but Satellite imagery can be very expensive and satellites have limited resolution, hence the data recorded is not accurate instead an average of the area. Also, satellites require processing time for each image; hence they are not the best for wildfire detection. Recent advancements in technology such as IoT, artificial intelligence and 5G Networks have made possible to accurately detect forest fire with almost no need for human exposure [1]. Technologies such as IOT have enabled the development of several solutions for early fire detection and management [2].

A simple IOT based forest fire detection system using sensors is described by K. Mehta, S. Sharma and D. Mishra [8]. The Wireless Sensor Network is one of the best examples of a IOT based fire detection system; it enables the monitoring of a sizable forest utilizing several sensors that are capable of communicating with one another and effectively and precisely detecting forest fires [3]. To establish wireless communication links with many devices, a variety of approaches like Bluetooth, WIFI, RF, and LoRa wan can be employed. Many microcontrollers with these characteristics pre-installed are available on the market, such as the ESP32 and NodeMCU. At mounting heights of 2.5 to 3 meters and spacings of up to 5 to 10 meters, a WSN prototype utilizing ESP32 was able to detect fire within 30 seconds and gas within 10 seconds [4]. Sensor

network with LoRa wan can be used to cover a wider region [9]. It can be challenging, though, to supply power to these sensors in the forest. Because forests receive a lot of sunlight, employing solar energy to power the sensors offers a sustainable and environmentally responsible alternative [5].

In order to contain the fire before it spreads, we can further enhance these systems to incorporate a fire extinguisher feature that uses a pressured water flushing tap [10].

The creation of different Artificial Neural Network models and detection algorithms to identify forest fires using meteorological parameters [11] or images taken with a regular camera [7][12] have been made possible by advances in machine learning. Yolo, one of the newest algorithms for detecting forest fires, shows to be incredibly effective at quickly and accurately identifying flames, even in large-scale photos with low resolution [6]. Nevertheless, these techniques run the risk of setting off false alarms, which can waste money and inconvenience the fire service.

CHAPTER-3

PRINCIPAL COMPONENTS

3. PRINCIPAL COMPONENTS

3.1 PRINCIPAL COMPONENTS OF PROJECT

1. Raspberry Pi 4 Model B
2. NodeMCU / ESP8266
3. MQ-135 Sensor
4. HP w300 Webcam

3.1.1 Raspberry Pi 4 Model B

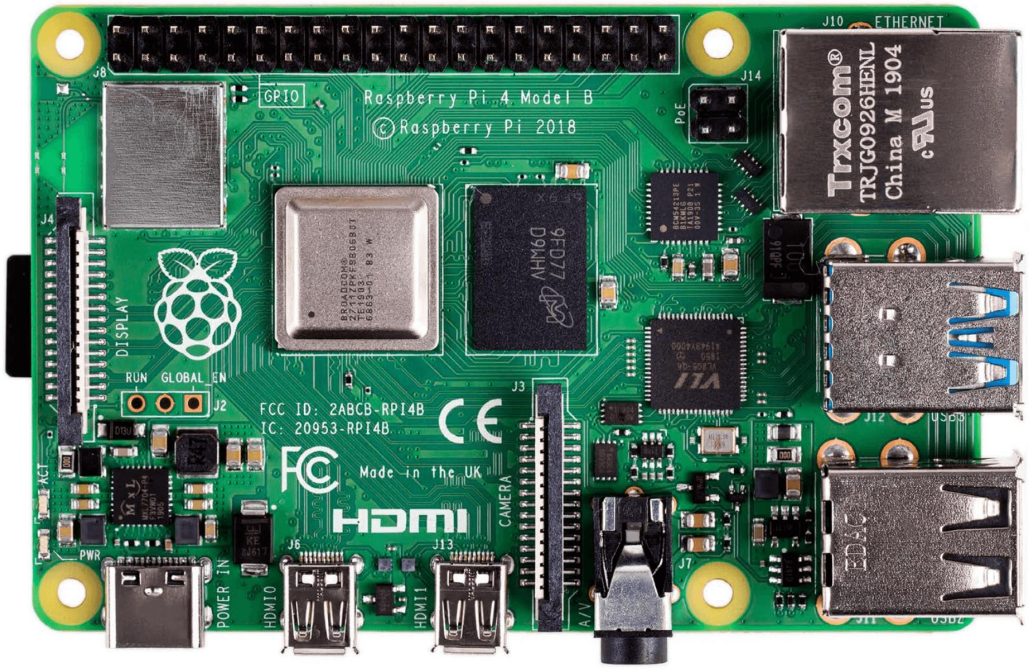

Fig 3.1 – Raspberry Pi

Raspberry Pi is a series of small, affordable, single-board computers developed by the Raspberry Pi Foundation, a UK-based charity organization. The Raspberry Pi 4 is a credit card-sized computer that plugs into a display, keyboard, and mouse, providing a low-cost, compact computing solution. The Raspberry Pi includes a GPIO header that

allows for easy connection to other hardware components. Raspberry Pi supports various operating systems, including Raspbian, which is a Debian-based Linux distribution specifically designed for Raspberry Pi. Raspberry Pi supports a variety of programming languages, including Python, Scratch, Java, C, and more.

For the end user, Raspberry Pi 4 Model B provides desktop performance comparable to entry-level x86 PC systems. This product retains backwards compatibility with the prior-generation Raspberry Pi 3 Model B+ and has similar power consumption, while offering substantial increases in processor speed, multimedia performance, memory, and connectivity. The dual-band wireless LAN and Bluetooth have modular compliance certification, allowing the board to be designed into end products with significantly reduced compliance testing, improving both cost and time to market.

Specifications:

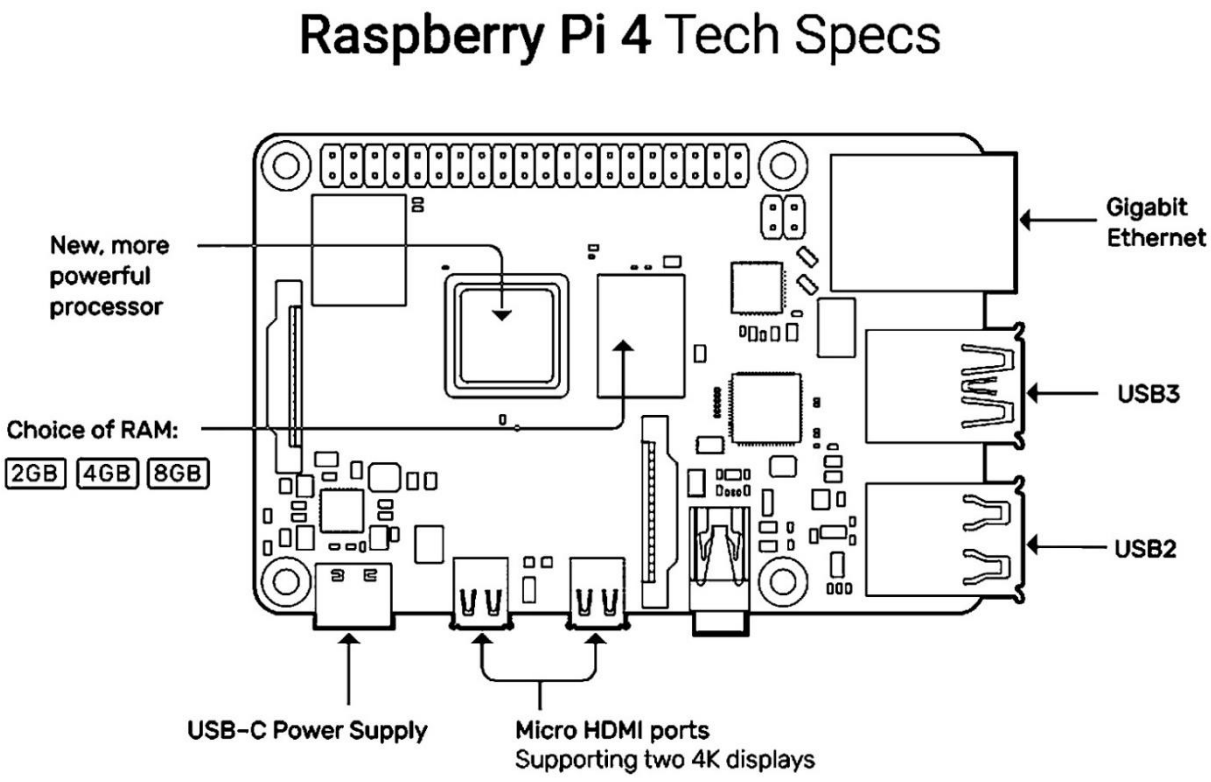

Fig 3.2 – Raspberry Pi Specifications

Processor: Broadcom BCM2711, quad-core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz.

Memory: 1GB, 2GB, 4GB or 8GB LPDDR4 (depending on model) with on-die ECC.

Connectivity: 2.4 GHz and 5.0 GHz IEEE 802.11b/g/n/ac wireless LAN, Bluetooth 5.0, BLE Gigabit Ethernet 2 × USB 3.0 ports 2 × USB 2.0 ports.

GPIO: Standard 40-pin GPIO header (fully backwards-compatible with previous boards).

Video & sound: 2 × micro-HDMI ports (up to 4Kp60 supported) 2-lane MIPI DSI display port 2-lane MIPI CSI camera port 4-pole stereo audio and composite video port.

Multimedia: H.265 (4Kp60 decode); H.264 (1080p60 decode, 1080p30 encode); OpenGL ES, 3.0 graphics.

SD card support: Micro SD card slot for loading operating system and data storage.

Input power: 5V DC via USB-C connector (minimum 3A1) 5V DC via GPIO header (minimum 3A1) Power over Ethernet (PoE)–enabled (requires separate PoE HAT).

Environment: Operating temperature 0–50ºC.

Production lifetime: Raspberry Pi 4 Model B will remain in production until at least January 2031.

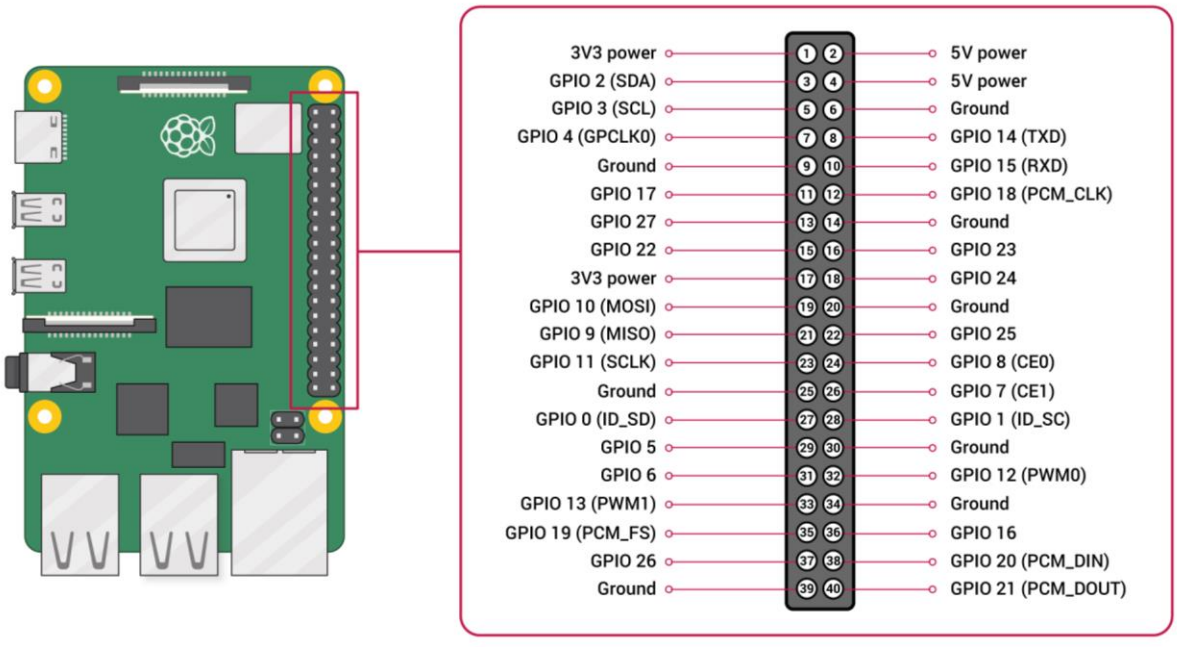

Fig 3.3 – Raspberry Pi pin diagram

Raspbian OS:

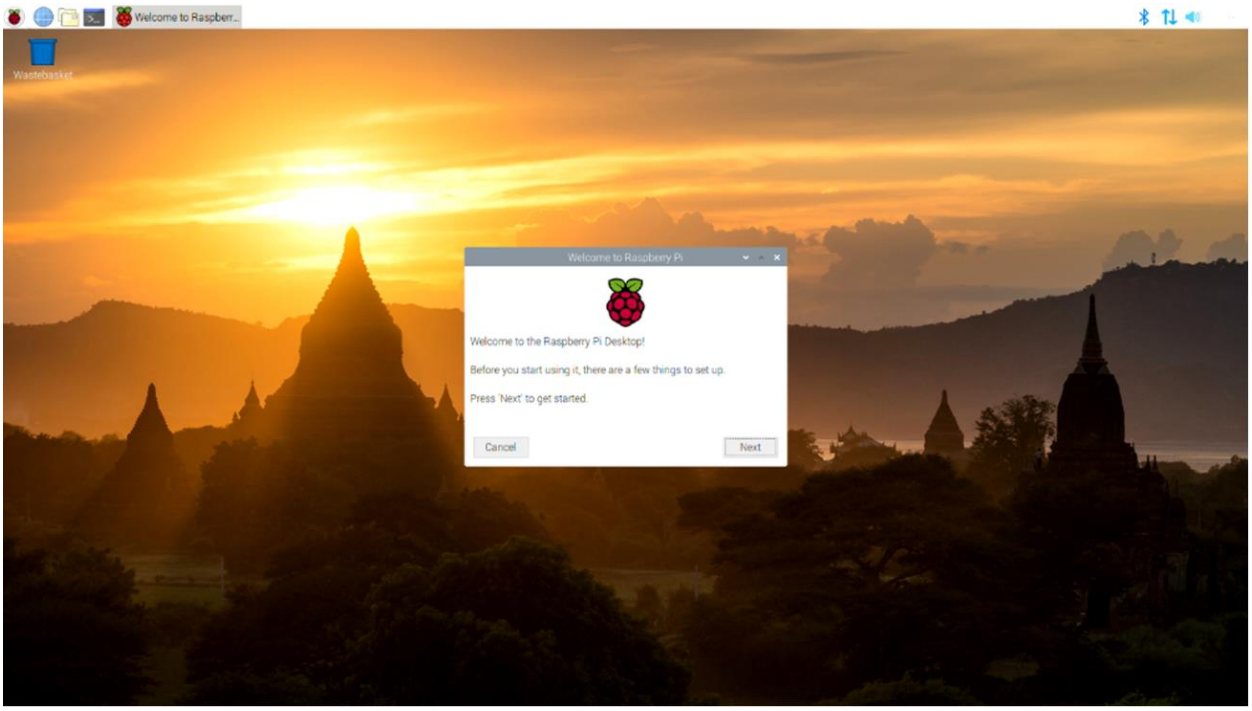

Fig 3.4 – Raspbian OS Interface

For the Raspberry Pi line of small single-board computers, Raspberry Pi OS (previously known as Raspbian) is an operating system with a Unix-like interface that is based on the Debian GNU/Linux distribution. Originally created on its own in 2012, the Raspberry Pi Foundation has been selling it as the main operating system for these boards since 2013.

For the Raspberry Pi with ARM CPUs, Raspberry Pi OS is well efficient. With the exception of the Pico microcontroller, it operates on all Raspberry Pi. With a customized version of the LXDE desktop environment and the Open box stacking window manager, Raspberry Pi OS has its own theme. A copy of VLC, the lightweight

Chromium web browser, and the computer algebra system Wolfram Mathematica are included in the default distribution.

The major improvements of the Raspberry Pi 4B compared to the Raspberry Pi 3B+ are:

- A slightly better CPU, with an improved clock speed.
- More RAM available (up to 8 GB).
- The 2 USB 2.0 ports replaced with USB 3.0 ports (which are way faster).
- Dual screen is now possible, with the 2x Micro-HDMI ports.
- Ethernet, Wi-Fi and Bluetooth are slightly faster with a Raspberry Pi 4B.

Complete hardware specifications comparison is given in the table below:

Table 3.1 – Raspberry Pi model comparison

	Raspberry Pi 3B+	**Raspberry Pi 4 B**
CPU	4x Cortex-A53 1.4 GHz	4x Cortex-A72 1.5 Ghz (or 1.8Ghz on recent models)
RAM	1 GB	1 GB, 2 GB, 4 GB or 8 GB
GPU	Broadcom IV @ 400 MHz	Broadcom VI @ 500 MHz
USB	4x USB 2.0 ports	2x USB 2.0 ports 2x USB 3.0 ports
HDMI	1x HDMI	2x Micro-HDMI
Ethernet	10/100/1000 Mbits/s (max speed: 300 Mbits/s)	10/100/1000 Mbits/s
Bluetooth	4.2 LS BLE	5.0
Power input	Micro USB	USB-C
Power (idle)	459 mA	600 mA
Wi-Fi speed	44 Mbits/sec	50 Mbits/sec

3.1.2 NodeMCU / ESP8266

Fig 3.5 – NodeMCU / ESP8266

The **ESP8266** is a low-cost Wi-Fi microprocessor that is made in Shanghai, China by Espressif Systems and includes integrated TCP/IP networking software and microcontroller functionality.

In August 2014, the chip gained popularity among English-speaking makers thanks to the ESP-01 module produced by Ai-Thinker, a third-party producer. With the help of this little module, microcontrollers can establish basic TCP/IP connections and connect to Wi-Fi networks by utilizing Hayes-style commands.

NodeMCU is an open-source development board and firmware with Lua programming specifically designed for Internet of Things applications. It consists of hardware based on the ESP-12 module and firmware running on Espressif Systems' ESP8266 Wi-Fi SoC.

NodeMCU is a low-cost, small and powerful open source IoT platform. It initially included firmware which runs on the ESP8266 Wi-Fi SoC from Espressif Systems, and hardware

which was based on the ESP-12 module. It is compatible with Arduino IDE and micro python making prototyping faster.

The ESP-12E module, which houses the ESP8266 chip with Tensilica Xtensa 32-bit LX106 RISC microprocessor, is included with the NodeMCU ESP8266 development board. This microprocessor runs at a configurable clock frequency of 80MHz to 160MHz and supports RTOS. For storing data and programs, NodeMCU contains 4MB of Flash memory and 128 KB of RAM. It is perfect for Internet of Things projects due to its powerful processing capacity, built-in Wi-Fi and Bluetooth, and Deep Sleep Operating capabilities.

A Micro USB jack and VIN pin (External Supply Pin) can be used to power NodeMCU. It has I2C, SPI, and UART interface capability.

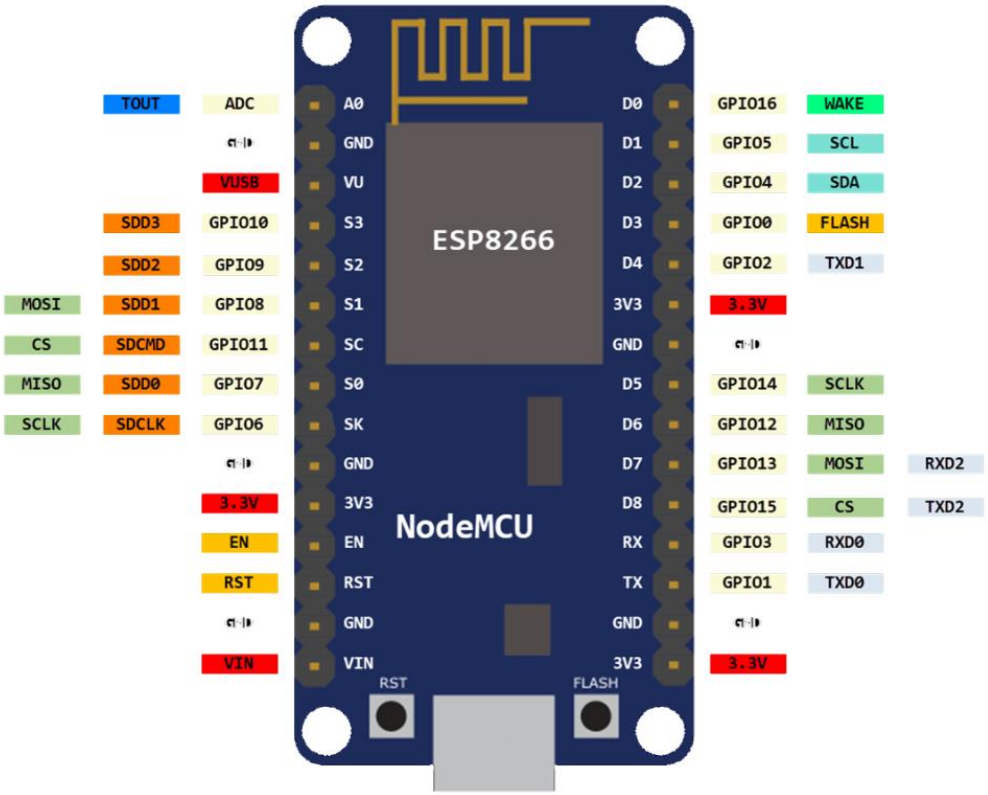

Fig 3.6 – NodeMCU / ESP8266 Pin Diagram

NodeMCU Development Board Pinout Configuration:

Table 3.2 – NodeMCU Pin Description

Pin Category	Name	Description
Power	Micro-USB, 3.3V, GND, Vin	**Micro-USB:** NodeMCU can be powered through the USB port **3.3V:** Regulated 3.3V can be supplied to this pin to power the board **GND:** Ground pins **Vin:** External Power Supply
Control Pins	**EN, RST**	The pin and the button reset the microcontroller
Analog Pin	A0	Used to measure analog voltage in the range of 0-3.3V
GPIO Pins	GPIO1 to GPIO16	NodeMCU has 16 general purpose input-output pins on its board
SPI Pins	SD1, CMD, SD0, CLK	NodeMCU has four pins available for SPI communication.

UART Pins	TXD0, RXD0, TXD2, RXD2	NodeMCU has two UART interfaces, UART0 (RXD0 & TXD0) and UART1 (RXD1 & TXD1). UART1 is used to upload the firmware/program.
I2C Pins		NodeMCU has I2C functionality support but due to the internal functionality of these pins, you have to find which pin is I2C.

NodeMCU ESP8266 Specifications & Features

- Microcontroller: Tensilica 32-bit RISC CPU Xtensa LX106
- Operating Voltage: 3.3V
- Input Voltage: 7-12V
- Digital I/O Pins (DIO): 16
- Analog Input Pins (ADC): 1
- UARTs: 1
- SPIs: 1
- I2Cs: 1
- Flash Memory: 4 MB
- SRAM: 64 KB
- Clock Speed: 80 MHz
- USB-TTL based on CP2102 is included onboard, Enabling Plug n Play
- PCB Antenna
- Small Sized module

How to program NodeMCU

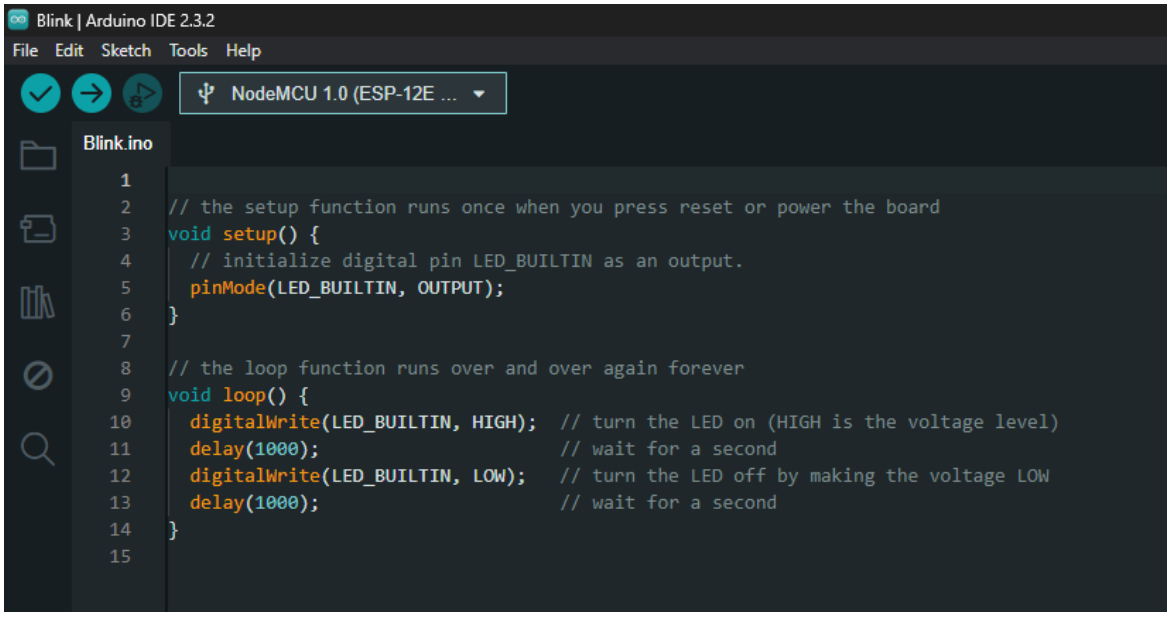

Fig 3.7 – Arduino Software

Once the circuit has been created on the breadboard, you'll need to upload the program (known as a sketch) to the ESP. The sketch is a set of instructions that tells the board what functions it needs to perform. The software used to create Arduino sketches is called the IDE which stands for Integrated Development Environment. The software is free to download and can be found at https://www.arduino.cc/en/Main/Software

Every Arduino sketch has two main parts to the program:

void setup () – Sets things up that have to be done once and then don't happen again.

Void loop () – Contains the instructions that get repeated over and over until the board is turned off.

Different Types of NodeMCU boards:

NodeMCU v0.9:

This is the NodeMCU development board's initial release. The ESP-12 module with the ESP8266 SoC is part of it. However, version 1.0 of the board swiftly superseded it because of its huge size and a few design problems.

NodeMCU v1.0 (NodeMCU V2):

The most popular and extensively used NodeMCU version, this board fixes many of the issues with the previous version and comes with the ESP-12E module. Another name for it is NodeMCU V2.

Version 2.0 (or V3) of the NodeMCU:

With a somewhat bigger size, this version provides capabilities similar to the V1.0 and is based on the ESP-12E core. NodeMCU V3 is another name for it, however it's not as widely used as V1.0/V2 version.

Table 3.3 Comparison between NodeMCU ESP8266 vs ESP32

Boards	**NodeMCU ESP8266**	**ESP32**
CPU	single-core processor	Quad-core Cortex-A72
RAM	64KB	2GB, 4GB, or 8GB
Internet Connectivity	Built-in WIFI	both Built-in WIFI & Ethernet port
Bluetooth Connectivity	not present	present
Price	cheaper than ESP32	higher
Operating Frequency	80 MHz (can be overclocked to 160 MHz)	160 MHz (can be overclocked to 240 MHz)
Analog-to-Digital Converter (ADC)	10-bit	12-bit
GPIO	17	34

3.1.3. MQ-135 Sensor

Fig 3.8 MQ-135 Sensor

It is a popular gas sensor that is widely used for detecting a variety of gases in the air. The sensor typically consists of a sensing element, a heater (to maintain a constant temperature for the sensing element), and electrodes. When the sensor comes into contact with a particular gas, the resistance of the sensing element changes, and this change is used to determine the gas concentration.

MQ-153 is capable of detecting / measuring:

- NH3
- NOx
- alcohol
- Benzene
- smoke
- CO2

Specifications:

- Operating Voltage is +5V
- Detect/Measure NH3, NOx, alcohol, Benzene, smoke, CO2, etc.
- Analog output voltage: 0V to 5V
- Digital output voltage: 0V or 5V (TTL Logic)
- The Sensitivity of Digital pin can be varied using the potentiometer

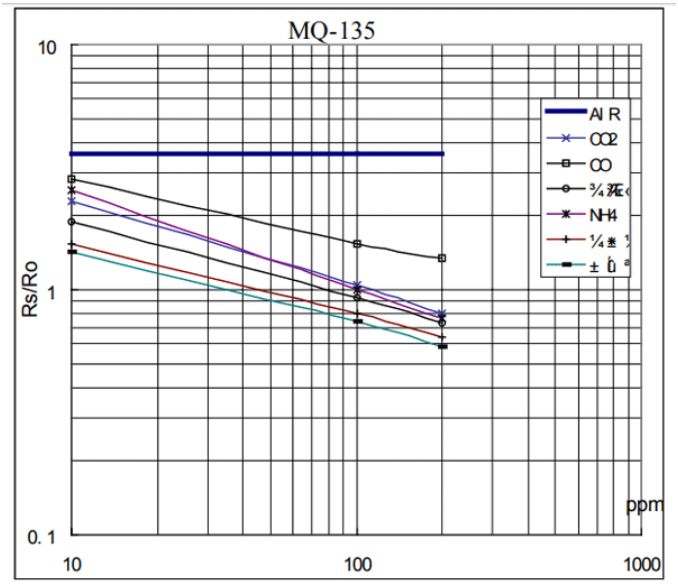

Fig 3.9 (Rs/Ro) v/s PPM graph taken from the MQ135 datasheet.

3.1.4. HP w300 Webcam

Fig 3.10 HP w300 Webcam

Wired Camera that can be connected to the Raspberry Pi, to capture and store images and videos digitally. It is capable of recording videos at 30fps (frames per second) with a resolution of 1920 x 1080P

Specifications:

- Connectivity: USB
- Series: HP
- Connection Type: Wired USB Type-A

- Microphone Type: Dual microphone
- Focus Type: Fixed focus
- Diagonal Field Of View (DFoV): 72°
- Zoom Capability: 1x
- Camera Resolutions: 1080P FHD 30fps
- Total Camera Resolution (Megapixels): 1080p
- Special Features: Tripod support
- Compatible Operating Systems: Windows XP, Vista, 7, 8, 10, and Mac OS X 10.6 or higher
- Cable Length: 1500 mm
- Minimum System Requirements: USB-A
- Colour: Black
- Swivel: 270°
- Tilt: 15°

3.2 SOFTWARE DETAILS

1. YOLOv8
2. ESP-NOW Library
3. PAINLESS MESH Library
4. ARDUINO IDE
5. OPEN CV (CV2)
6. SMTP

3.2.1 YOLOv8

Ultralytics YOLOv8 (You Look Only Once Version 8) is a real-time object recognition model known for its speed, accuracy, and efficiency. This model belongs to the family of YOLO object detection models and can feed images to a neural network in one step.

1) YOLOv8 Architecture

- Backbone (CSPDarknet53): The backbone functions as the feature extractor. It consists of a modified CSPDarknet53, a convolutional neural network with repeated Cross-Stage Partial (CSP) modules.
- CSP modules enhance gradient flow, allowing for deeper and more efficient networks while reducing computation. The backbone extracts meaningful features from the input images at varying scales.
- Neck (SPP, PAN): The neck aggregates the features extracted from the backbone. In YOLOv8, it incorporates:
- SPP (Spatial Pyramid Pooling): Enhances receptive field and enriches feature representations by applying pooling operations with varying kernel sizes on the feature maps.
- PAN (Path Aggregation Network): Improves feature propagation and fusion, particularly between lower and higher layers of the network, boosting the model's ability to localize objects accurately.

- Head (Decoupled Head): The head generates the final predictions from the processed features. YOLOv8 utilizes a decoupled head featuring two parallel branches:
- Classification Branch: Determines the class (e.g., 'fire')
- Regression Branch: Determines the bounding box coordinates and object-ness score (confidence that an object is present).

2) YOLO Functionality

1. One forward pass:

   YOLOv8 performs object detection through a neural network and is computationally efficient compared to models that require multiple passes.
2. Boundary box prediction:

   YOLOv8 predicts the bounding box corresponding to each object for each cell in the image. This allows you to find multiple objects in one image.
3. Class prediction:

   YOLOv8 estimates class probabilities for each box. Identify the objects inside the bounding box and determine the probability score of the correct prediction.
4. Object similarity score:

   The model predicts a target score for each bounding box, indicating the probability that an object is present in the box. A higher object score helps filter out extraneous or background bounding boxes.
5. Non-Maximum Suppression (NMS):

   After prediction, YOLOv8 applies non-maximal suppression. This is a post-processing step that removes redundant or overlapping bounding boxes and keeps only the most reliable detections.

6. High-speed performance:

   YOLOv8 is optimized for fast performance and is suitable for real-time applications such as video analysis and object detection in dynamic environments.

7. Versatility:

   YOLO v8 is versatile and can be trained to recognize a wide range of object classes. Its adaptability allows it to be applied in various domains beyond traditional object detection.

8. Improved Accuracy:

   YOLOv8 builds on the strengths of previous versions and includes improved accuracy, especially in difficult scenarios such as detecting small objects and duplicate samples.

9. Easy integration:

   YOLOv8 is designed to be easily integrated into a variety of applications and frameworks, making it easy to deploy in a variety of projects.

3) Key Improvements in YOLOv8:

Efficiency: Advancements in network design and the introduction of techniques like SPPF (instead of SPP) lead to faster processing times.

Decoupled Head: Separating the object classification and regression tasks enhances accuracy in object detection.

Stronger Backbone: The modified CSPDarknet53 architecture provides a powerful foundation for feature extraction.

3.2.2 ESP-NOW

The ESP-NOW library is a feature provided by Espressif Systems for their ESP microcontrollers. It allows for efficient communication between ESP8266/ESP32 devices without relying on Wi-Fi networking infrastructure or TCP/IP. With ESP-NOW, devices can communicate directly over a local wireless connection, without needing a router. Operating in the 2.4 GHz band, it supports data rates of up to 250 kbps. Devices must be paired before communication, requiring the exchange of unique mac address and cryptographic keys.

ESP-NOW is a wireless communication protocol based on the data-link layer, which reduces the five layers of the OSI model to only one. This way, the data need not be transmitted through the network layer, the transport layer, the session layer, the presentation layer, and the application layer. Also, there is no need for packet headers or unpackers on each layer, which leads to a quick response reducing the delay caused by packet loss in congested networks.

Key Features:

- No Gateway and Receiving-IC Required:
  The switch directly sends data to other devices equipped with Espressif Wi-Fi SoCs, without additional SoCs required.
- Quicker Controllable Response:
  After you send a signal, the device will respond within 150ms.
- Reliable Bidirectional Data Transmission:
  The stable master-slave bidirectional data transmission ensures the high success rate of Wi-Fi packet delivery.
- Diversified Product Forms:
  Espressif's Wi-Fi switch supports such products as sticker switches, multi-button switches, touch switches, and knob switches.
- Easy-to-Configure

With its simple topology, Espressif's Wi-Fi switch is really easy when it comes to device pairing. The switch supports one-to-one, one-to-multi, and multi-to-multi control modes.

- Ultra-Low Power Consumption
  ESP32-C2 is powered off in the sleep mode. One CR2032 coin battery can last up to 5 years if the switch is flipped approximately 10 times a day.

Other Features:

- Coexists with Wi-Fi and Bluetooth LE, and supports various series of Espressif SoCs with Wi-Fi connectivity.
- It has a fast and user-friendly pairing method that is suitable for connecting “one-to-many” and “many-to-many” devices, while also controlling them.
- Occupies fewer CPU and flash resources.
- Can be used as an independent protocol that helps with device provisioning, debugging, and firmware upgrades.
- ECDH and AES algorithms make data transmission more secure.
- The window synchronization mechanism greatly reduces power consumption.

3.2.3 PAINLESS MESH

PainlessMesh is a library that handles all the technical aspects of setting up a basic mesh network with esp8266 and esp32 hardware. The idea is to free the programmer from worrying about the configuration or management of the mesh network so they can work with it.

True ad-hoc networking

As a genuine ad-hoc network, mesh eliminates the need for routers, central controllers, or planning. Any system with one or more nodes will form a fully working mesh by self-organization. The amount of heap RAM that can be devoted to the sub-connections buffer equals the maximum size of the mesh, therefore it should be rather large.

JSON-based

Painless Mesh's messaging is all done via JSON objects. There are a few causes behind this. Firstly, it renders the code and messages easily comprehensible and legible for humans. Secondly, it streamlines the process of integrating painlessMesh with web applications, JavaScript front-ends, and other applications. Although there is some performance loss, it hasn't affected me yet. If someone would like to help, switching to binary messaging would be very simple.

3.2.4 ARDUINO IDE

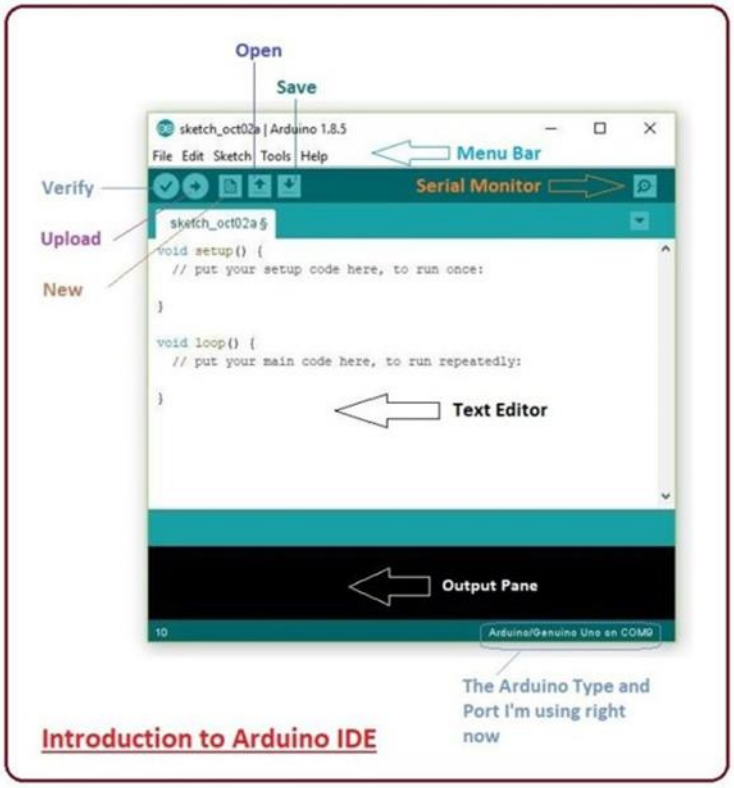

Fig 3.11 Arduino Software Interface

An authority programming presented by Arduino.cc, is for the most part utilized for composing, accumulating, and transferring the code in practically all Arduino modules/sheets. Arduino IDE is open-source programming and is effectively accessible to download and introduce.

Arduino IDE is open-source programming, planned by Arduino.cc and for the most part utilized for composing, accumulating, and transferring code to practically all Arduino Modules.

- It is an authority Arduino programming, making code assemblage too simple that even a typical individual with no earlier specialized information can consider making the plunge with the learning system.

- It is accessible for all working frameworks for example Macintosh, Windows, Linux and runs on the Java Platform that accompanies inbuilt capacities and orders that assume an indispensable part in troubleshooting, altering, and accumulating the code.
- A scope of Arduino modules accessible including Arduino Uno, Arduino Mega, Arduino Leonardo, Arduino Micro, and some more.
- Each of them contains a microcontroller on the board that is really customized and acknowledges the data as code.
- The principal code, otherwise called a sketch, made on the IDE stage will, at last, create a Hex File which is then moved and transferred in the regulator on the board.
- The IDE climate fundamentally contains two essential parts: Editor and Compiler where the previous is utilized for composing the necessary code and later is utilized for assembling and transferring the code into the given Arduino Module.
- This climate upholds both C and C++ dialects.

3.2.5 OPEN CV (CV2)

OpenCV, short for Open-Source Computer Vision is an open-source software library for computer vision and machine learning. It was initially created by Intel and is currently maintained by the OpenCV Foundation, a community of developers.

It is a vast open-source library for image processing, machine learning, and computer vision. Numerous programming languages, including Python, C++, Java, and others, are supported by OpenCV. It can recognize faces, objects, and even human handwriting by processing photos and movies. When it is integrated with various libraries, such as NumPy which is a highly optimized library for numerical operations, then the number of weapons increases in your Arsenal i.e. whatever operations one can do in NumPy can be combined with OpenCV.

The first OpenCV version was 1.0. OpenCV is released under a BSD license and hence it's free for both academic and commercial use. It has C++, C, Python, and Java interfaces and supports Windows, Linux, Mac OS, iOS and Android. When OpenCV was designed the main focus was real-time applications for computational efficiency. All things are written in optimized C/C++ to take advantage of multi-core processing.

Features:

- Open Source
  OpenCV is open source and released under the Apache 2 License. It is free for commercial use.
- Optimized
  OpenCV is a highly optimized library with focus on real-time applications.
- Cross-Platform
  C++, Python and Java interfaces support Linux, MacOS, Windows, iOS, and Android.

Functionality:

- Read the Image: OpenCV helps you to read the image from file or directly from camera to make it accessible for further processing
- Image Enhancement: You will be able to enhance image by adjusting the brightness, sharpness or contract of the image. This is helpful to visualize quality of the image.
- Object detection: As you can see in the below image object can also be detected by using OpenCV, Bracelet, watch, patterns, faces can be detected. This can also include to recognize faces, shapes or even objects.
- Image Filtering: You can change image by applying various filters such as blurring or Sharpening.
- Draw the Image: OpenCV allows to draw text, lines and any shapes in the images.

3.2.6 SMTP

The Simple Mail Transfer Protocol (SMTP) is a technical standard for transmitting electronic mail (email) over a network. Like other networking protocols, SMTP allows computers and servers to exchange data regardless of their underlying hardware or software. Just as the use of a standardized form of addressing an envelope allows the postal service to operate, SMTP standardizes the way email travels from sender to recipient, making widespread email delivery possible.

SMTP is a mail delivery protocol, not a mail retrieval protocol. A postal service delivers mail to a mailbox, but the recipient still has to retrieve the mail from the mailbox. Similarly, SMTP delivers an email to an email provider's mail server, but separate protocols are used to retrieve that email from the mail server so the recipient can read it.

The Simple Mail Transfer Protocol (SMTP) is a technical standard for transmitting electronic mail (email) over a network. SMTP enables data flow between computers and servers regardless of the underlying hardware or software, just like other networking protocols. Similar to how the postal service may function by using a standard method for addressing envelopes, SMTP standardizes the path taken by emails as they are sent from sender to receiver, enabling ubiquitous email delivery.

SMTP is a mail delivery protocol, instead of being a mail retrieval protocol. Mail is delivered to a mailbox by the postal service, but it is still the recipient's responsibility to take it out.

CHAPTER-4

WORKDONE

4. WORKDONE

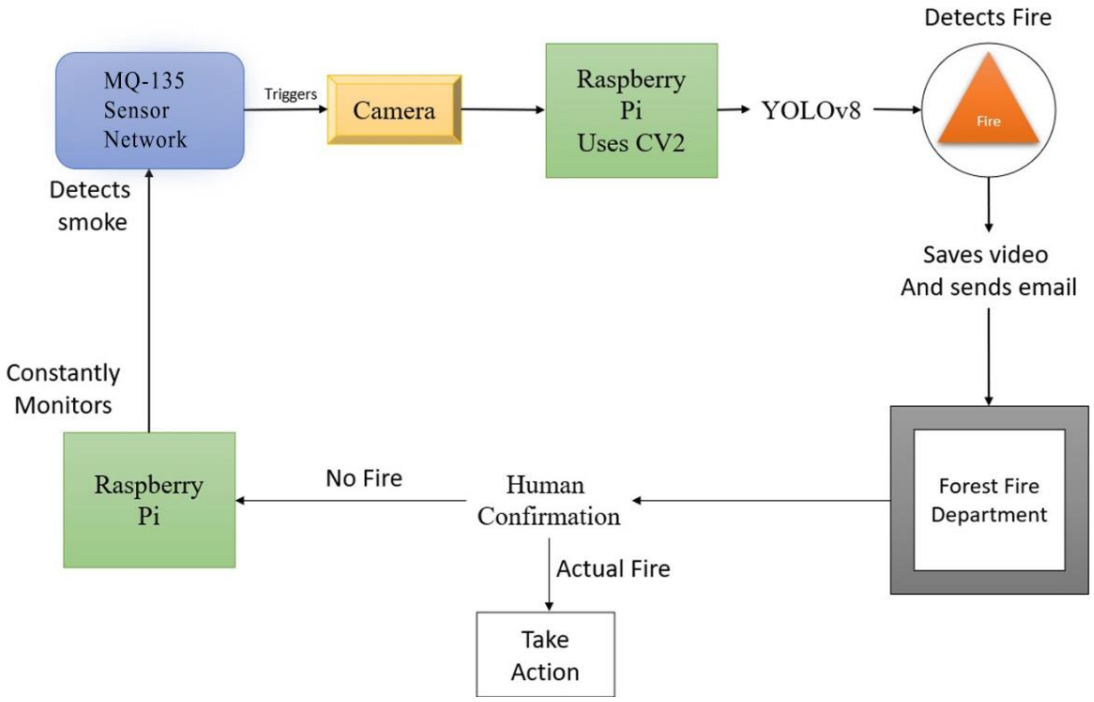

Fig 4.1 Block Diagram of the system

The Forest Fire Detection system uses the power of both IOT and Machine Learning to detect forest fire with a higher accuracy and power efficiency. It detects the gases and smoke generated during forest fire using Wireless Sensor Network of MQ sensors using NodeMCU and then use Raspberry pi to perform on-board image processing on the camera feed using YOLOv8 Object Detection Model that uses a single pass through its ANN to detect fires in image. Thus, this 2-step verification effectively reduces the accuracy and reduces the memory and power consumption of raspberry pi by only triggering camera after smoke detection. Also, the Real-time Processing using YOLOv8 allows the MCU to quickly send alert signal to the Fire department to act on the wildlife in its initial stage. The raspberry pi will form the Centre and main base station for the wireless sensor network. We

chose Raspberry pi as the main MCU due to its computing power, storage facility and connectivity

4.1 ASSEMBLY OF PROJECT

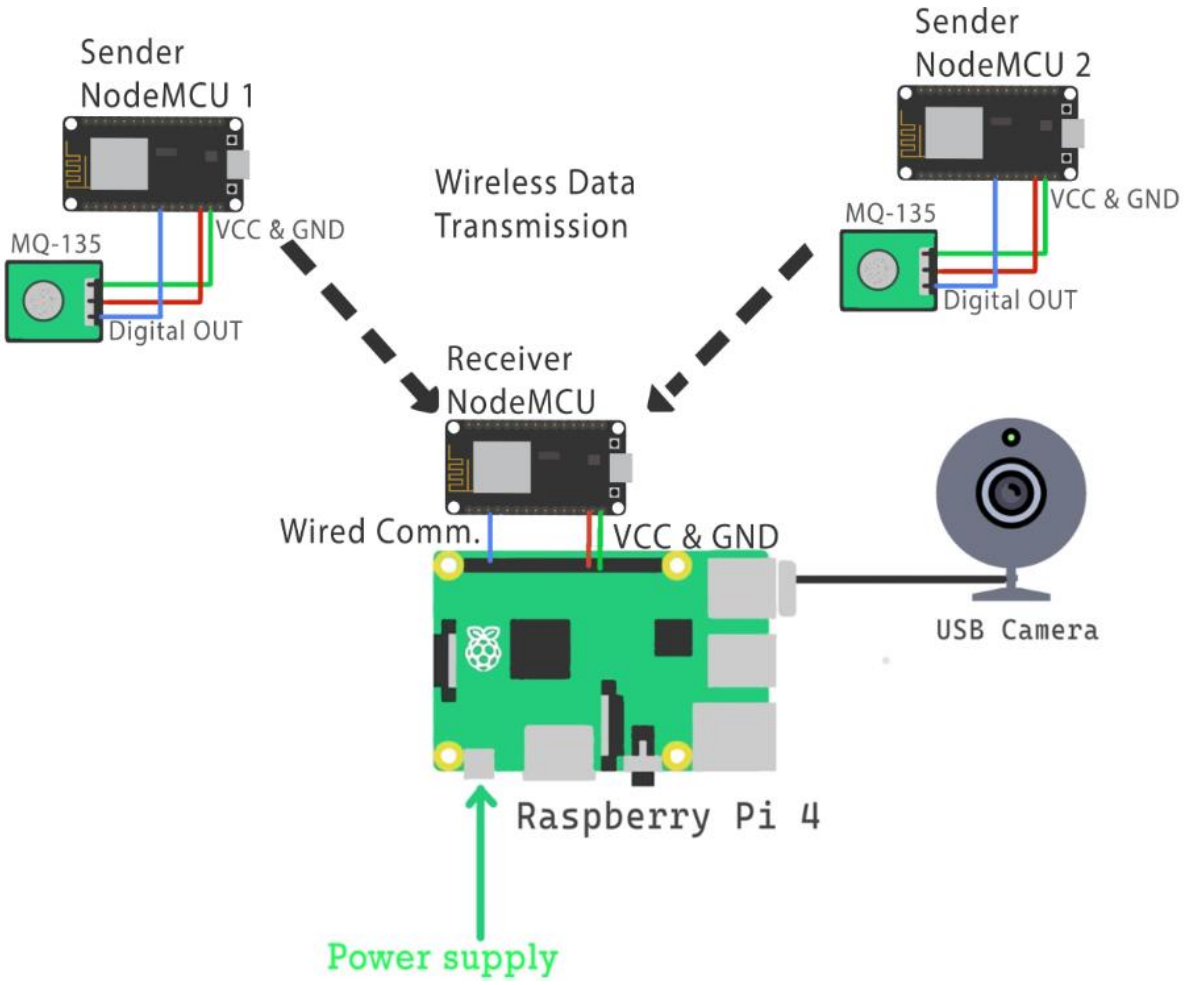

Fig 4.2 Assembly of Project

The System consists of three NodeMCUs. Two of them are programmed to be sender Nodes and one NodeMCU is programmed to be the receiver. The Receiver is connected to the Raspberry Pi using wires.

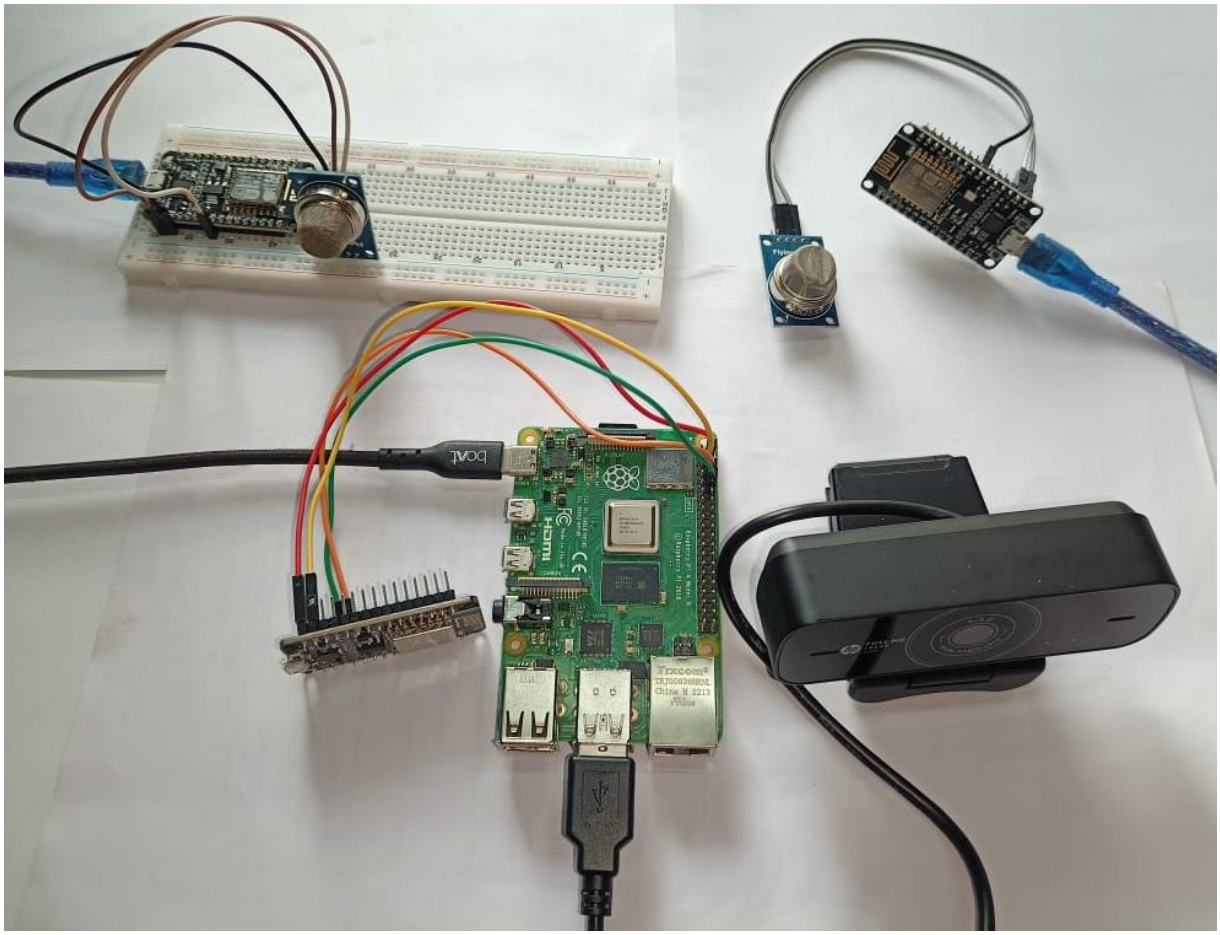

Fig 4.3 View of the Project

Sender NodeMCU:

- Power Supply: Each Sender NodeMCU can be powered by connecting a 3.3V / 5V power supply to its VIN pin and GND pin. Alternatively, they can be powered using a micro-USB cable connected to their programming / USB port.

- MQ-135 Sensor: Each Sender NodeMCU uses MQ135 sensor to monitor its environment for the presence of gases and smokes. The MQ135 sensors are powered using the 3V3 and GND pin of the NodeMCU. The DOUT Pin of MQ135 sensor, that outputs digital pulse upon detection, is connected to the Data pin of the NodeMCU.

Receiver NodeMCU:

- Power supply: The Receiver NodeMCU can be powered by connecting the 3.3V or 5V Pin of the Raspberry Pi to its VIN pin and the Ground pin of Raspberry Pi to its GND pin.

- Data: One of the data pins of the Receiver Node is connected to the GPIO pin of the Raspberry pi. This establishes a wired data transfer of Detection Message and NodeID from the Receiver NodeMCU to the Raspberry Pi.

Raspberry Pi:

- Power supply: A USB type C cable, connected to its USB C port, is used to supply stable and sufficient power supply to the Raspberry Pi.

- Receiver NodeMCU: The Receiver NodeMCU data pin is connected to the GPIO pin of raspberry pi for wired transfer of the sensor data from the network.

- Camera: The USB type-A cable of the Camera is connected to the USB port of the Raspberry Pi.

4.2 Working of the Project

4.2.1 Initialization and Sensor Monitoring

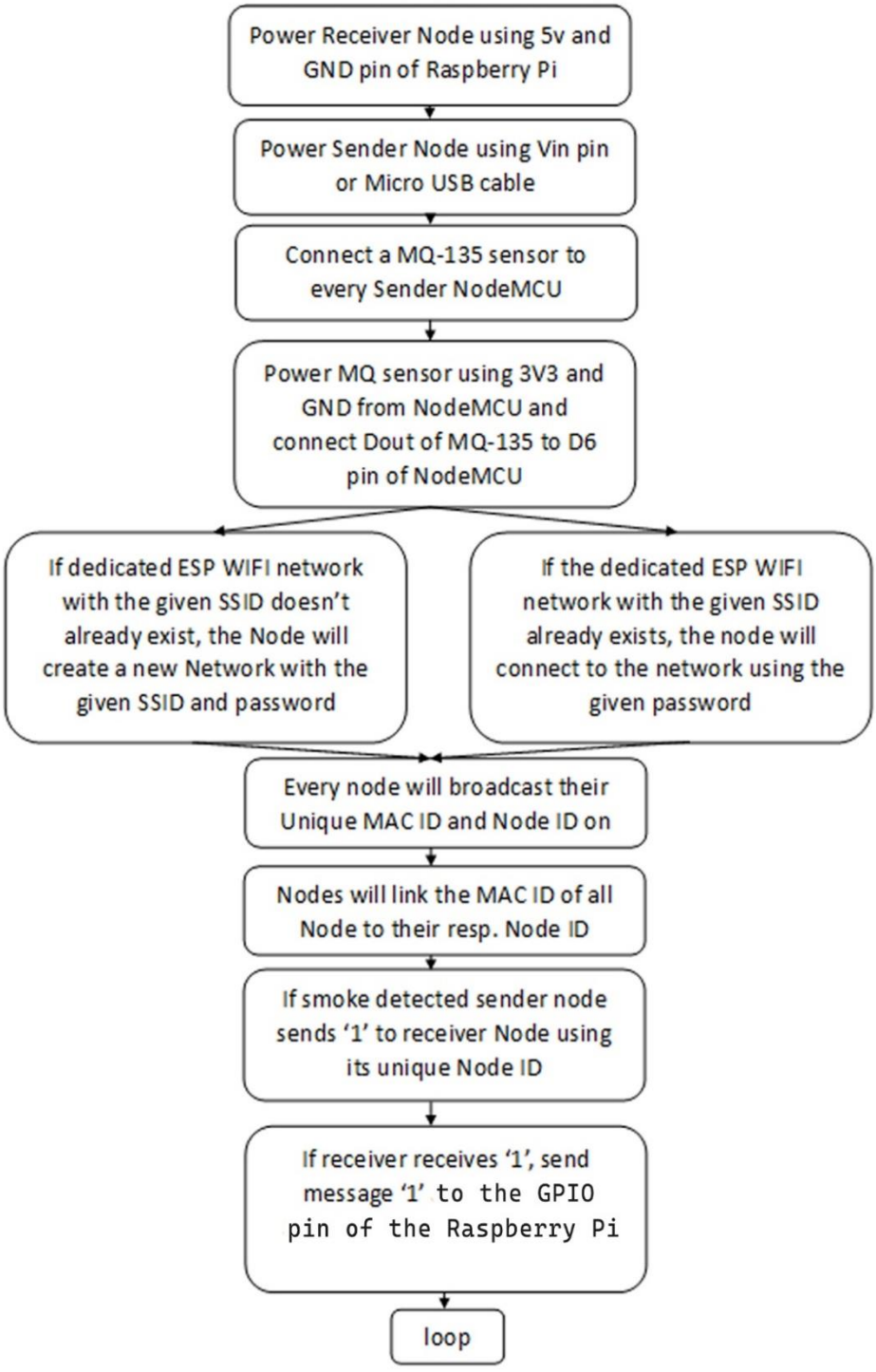

Fig. 4.4 Flow Chart of the Wireless Sensor Network

1. Start the Raspberry Pi and Receiver NodeMCU.

The Receiver NodeMCU is powered using the 5V (or 3.3V) GPIO pins of the Raspberry pi. Once the Raspberry pi is powered, the Receiver NodeMCU will also get switched on and it will create a new WIFI network for the sensor network.

The SSID and password of this WIFI network is available to every node in the system.

2. Power up the Sender NodeMCUs.

Once powered up, every node will connect to the WIFI network with the given SSID and password. The mac-address of all nodes will be paired up to their respective Node ID.

```
17:12:10.786 -> CONNECTION: New AP connection incoming
17:12:10.818 -> CONNECTION: painlessmesh::Connection: New connection established.
17:12:10.881 -> CONNECTION: newConnectionTask():
17:12:10.913 -> CONNECTION: newConnectionTask(): adding 577469764 now= 41700354
17:12:10.977 -> Changed connection
```

Fig. 4.5 New Connection Message

3. Smoke detection using MQ-135 sensor:

The MQ-135 sensors connected to every sender node, are active Low and hence output 0V at the Data pin of the NodeMCU upon smoke detection. When the Sender NodeMCU detects this change in input, it send the detection message to the receiver NodeMCU.

```
17:13:07.918 -> Received message by name from: Node02, 11
17:13:56.920 -> Received message by name from: Node02, 11
17:15:02.798 -> Received message by name from: Node01, 11
17:15:13.591 -> Received message by name from: Node01, 11
```

Fig. 4.6 Received message by Receiver

4. Reception by Raspberry pi.

The receiver NodeMCU, upon receiving the detection message from the Sender Node, sends a detection message to the Raspberry pi . When the Raspberry pi receives this message, it initiates the YOLO application.

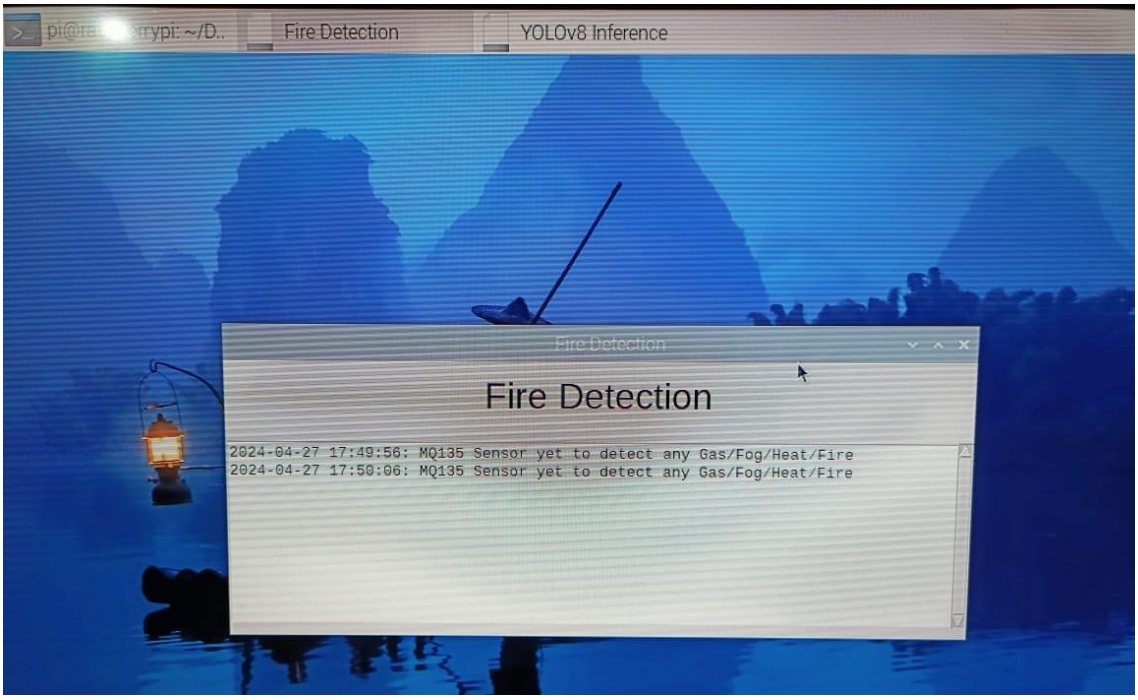

Fig. 4.7 Raspberry Pi: Sensor Status window

4.2.2 Fire detection using YOLOv8

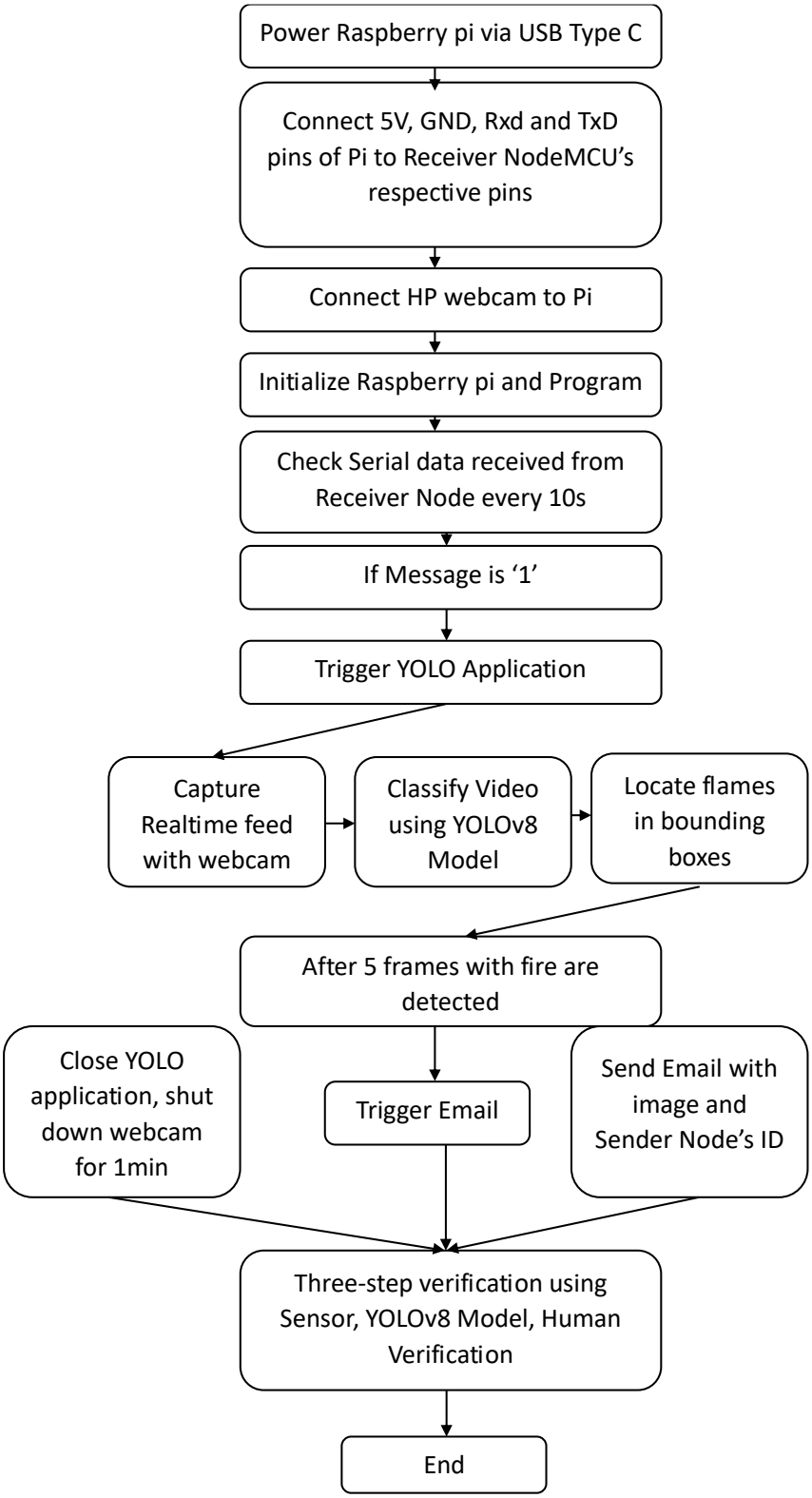

Fig. 4.8 Flow chart of Main system

1) Activation: Once the YOLO program is activated, it performs object detection on the live feed from the connected camera.

2) Image capture and processing: The camera uses the Open CV library (cv2) to capture video frames. The YOLOv8 model processes these images and places a bounding box around each target/flame in the image along with a confidence value.3. Image Accumulation: The system continuously accumulates detection results across multiple images to ensure flame identification.

3) Confirmation: Flame detection is confirmed when the system detects the presence of flame for five consecutive frames.

c. Notification:

1. Close the YOLO app and turn off your camera: After the flame detection is confirmed, the YOLO app closes to save memory and reduce power consumption. To reduce memory consumption, the camera will turn off for 1 minute.

2. Email Notification: The system launches an email program that uses Simple Mail Transfer Protocol (SMTP). An email notification with the ID of the NodeMCU that detected the smoke and the annotated image from YOLO detection is sent to the user for human verification.

d. Three-step verification:

The fire detection model uses a three-step validation process to reduce false alarms.

1. Sensor Verification: Initial detection by MQ-135 sensor.

2. Validation of YOLOv8: Analysis using a specially trained YOLOv8 fire detection model.

3. Human validation: Captured images are sent to the user or fire department for human validation, thereby improving the accuracy of the detection system.

This working model ensures an efficient and reliable fire detection system that uses IoT-based Wireless Sensor Network technology and advanced machine learning techniques to minimize false alarms and improve overall system accuracy.

CHAPTER-5

RESULT AND DISCUSSION

5. RESULT AND DISCUSSION

5.1 Sensor Network Detection

The WSN of NodeMCU was able to transmit the MQ sensor detection data to the Raspberry Pi within 15 seconds of detection every time. The nodes were able to transmit data very accurately and without any noise interference. Since the Nodes create their own local WIFI network, the absence of internet facility does not create any hindrance in the performance of the system.

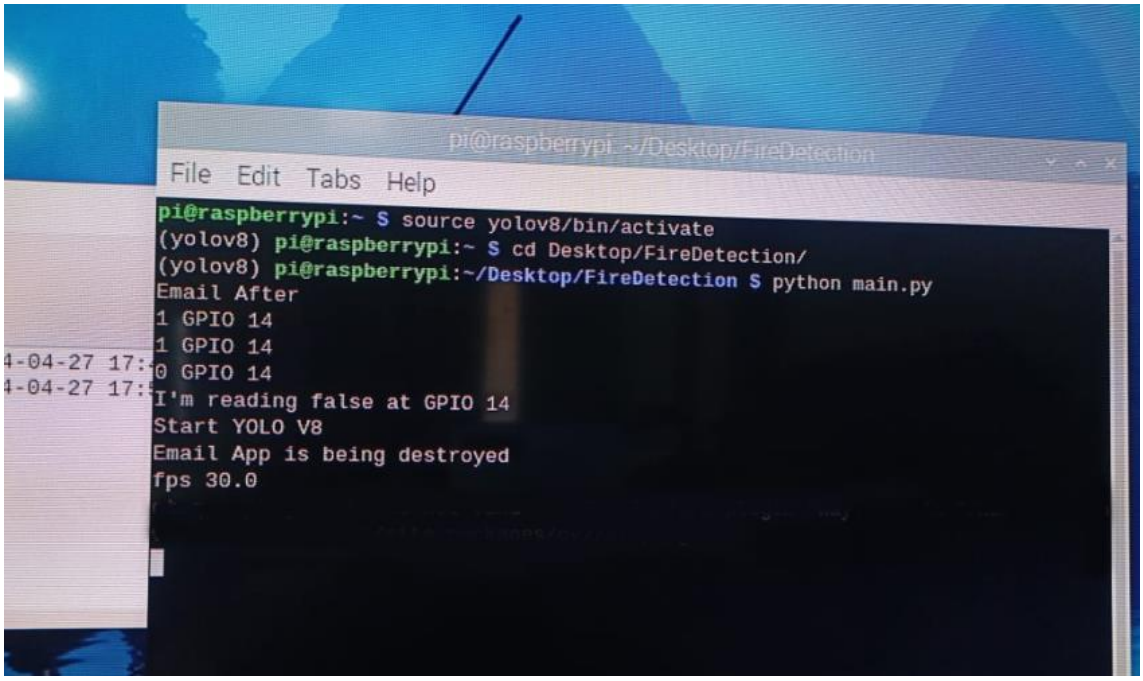

Fig. 5.1 Raspberry Pi: Output at Terminal

5.2 YOLO Detection

1. **Actual annotated frame from the model**

Fig. 5.2 YOLO detection

A matchstick was used to produce flames in Figure 5.2 above in order to verify the YOLO model of the system's detection capacity. With only 2GB of RAM and minimal processing capability, the model was nevertheless able to identify fires with accuracy and efficiency on the Raspberry Pi. The camera recorded a lot of lens flare because of the flame's close proximity, which made it difficult to discern the flame accurately. However, lens flare may be prevented in a real forest scenario if the camera is carefully set up at a high height.

2. Smartphone Camera

Fig. 5.3 Detection using Smartphone Camera

The YOLOv8 model for Fire Detection was applied on a video captured by Smartphone camera. When compared to the low-res webcam capture above, the model detected fire with a significantly higher accuracy and bounding boxes were also more accurately wrapped around the target object i.e. flames. This suggests that the camera used for the project should be high quality and resolution for more accurate detection.

3. Low Resolution Video

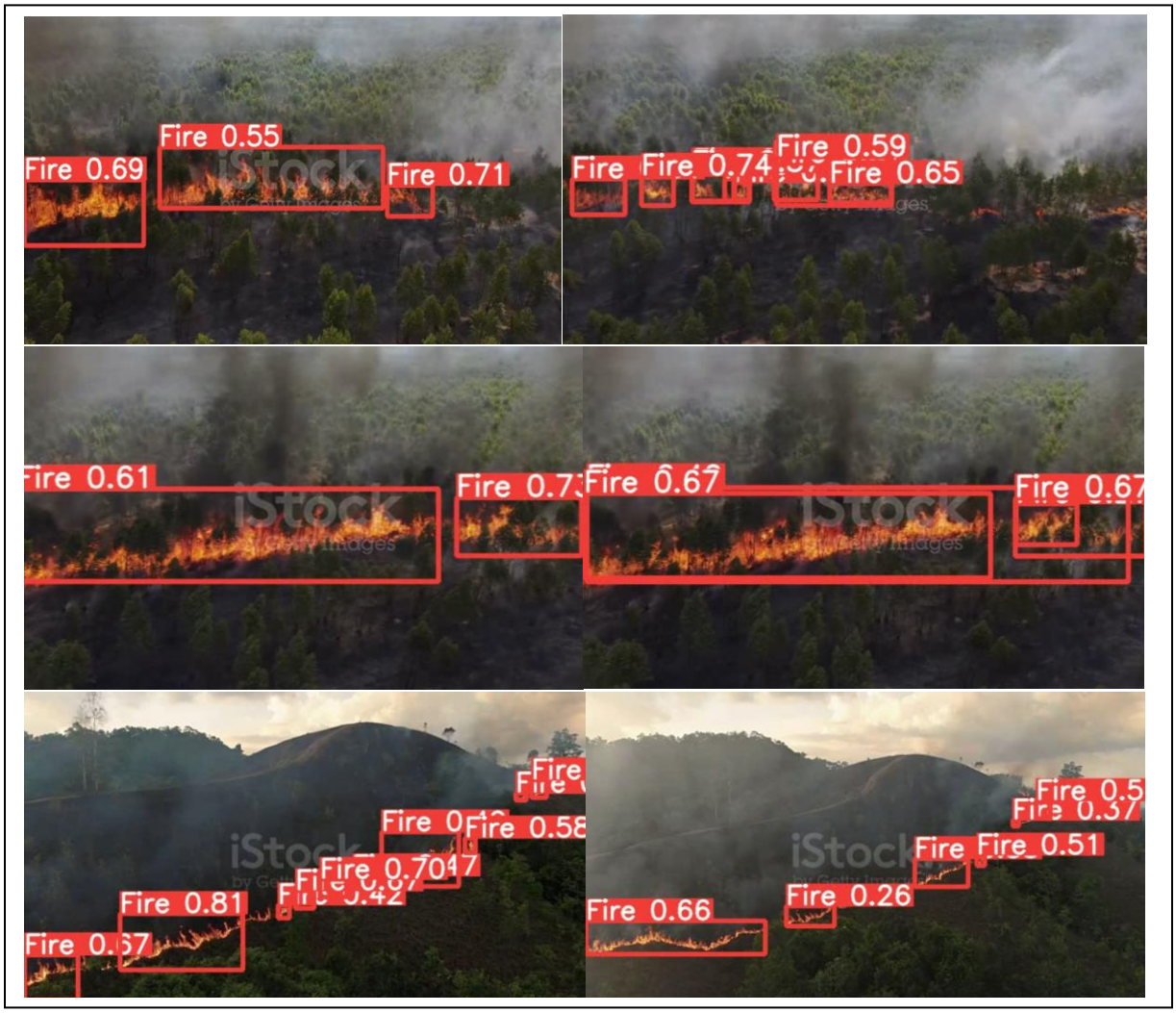

Fig. 5.4 Low Resolution video

Here, the YOLO model was used to detect fire in a low-resolution video as shown in Fig.4 colour accurate stock video and it showed promising results. It was able to enclose all flame object in the video with very less inaccurate detection and false positives.

4. High Resolution Video

Fig. 5.5 High Resolution video

Here, the model was used to detect fire in a high-resolution video as shown in Fig. 5 and it was able to accurately detect fire with high confidence value. Image above shows that model detected fire with a confidence value of 0.73 even when the fire was largely covered with smoke.

CHAPTER-6

CONCLUSION AND FUTURE SCOPE

6. CONCLUSION AND FUTURE SCOPE

6.1 CONCLUSION

In conclusion, the Raspberry Pi project successfully integrates cutting-edge technologies to enhance safety and monitoring capabilities. By combining the power of a Raspberry Pi, Wireless Sensor Network of NodeMCU and MQ135 gas sensor, YOLOv8 object detection model, and a camera module, we've created a versatile system capable of detecting flames and gas in real-time. This Raspberry Pi project represents a successful convergence of hardware and software technologies, showcasing the potential for innovation in safety and environmental monitoring applications. Its modular design and adaptability lay the foundation for future enhancements, emphasizing the project's significance in contributing to the growing landscape of IoT and intelligent systems. Moving forward, opportunities for improvement include refining machine learning models for even more accurate detection, exploring additional sensors for a comprehensive environmental assessment, and enhancing the user interface for greater user interaction.

6.2 FUTURE SCOPE

The fire detection system based on YOLOv8, Raspberry Pi, and the MQ-135 Wireless Sensor Network exhibits promising capabilities, and its future scope could be extended in various ways to enhance its functionality, usability, and applicability. Some potential areas for future development and expansion include:

- Enhanced Machine Learning Models:

Improving the machine learning models consistently, especially the YOLOv8 model, can result in enhanced precision when it comes to detecting fires. By training the model on bigger and more varied datasets, it is possible to enhance its capability to handle different environmental situations and types of fires.

- Integration of Additional Sensors:

Gas sensors for detecting other potentially hazardous gases, such as temperature sensors or humidity sensors, can be incorporated to provide a more comprehensive assessment of fire risk and environmental conditions during environmental monitoring.

- User Interface Improvements:

Making enhancements to the user interface can greatly improve the accessibility and overall experience for users. By implementing a user-friendly dashboard or a mobile application, users can easily navigate the system.

- Integration with Smart Home Systems:

By merging the fire detection system with smart home systems or IoT platforms, we can achieve smooth communication with other connected devices. This dynamic integration opens doors for advanced automation, such as activating fire alarms, shutting off gas supplies, or alerting emergency services.

- Cloud Integration and Data Analytics:

Integrating the system with cloud platforms for data storage and analytics can facilitate large-scale data analysis, allowing for the identification of trends, predictive maintenance, and continuous system optimization.

- Community and Government Collaboration:

Collaborating with local communities and government agencies to deploy the system in public spaces, forests, or industrial areas can contribute to early fire detection on a larger scale. This collaboration can involve sharing data for better monitoring and management of fire-prone areas.

- Energy-Efficient Power Solutions:

Exploring energy-efficient power solutions, such as advanced battery technologies or energy harvesting techniques, can make the system more sustainable and resilient, especially in remote or off-grid locations.

CHAPTER-7

REFERENCES

7.REFERENCES

[1] S. Pandey, R. Singh, S. Kathuria, P. Negi, G. Chhabra and K. Joshi, "Emerging Technologies for Prevention and Monitoring of Forest Fire," 2023 International Conference on Innovative Data Communication Technologies and Application (ICIDCA), Uttarakhand, India, 2023, pp. 1115-1121, doi: 10.1109/ICIDCA56705.2023.10099572.

[2] P. Negi, S. Kathuria, R. Singh, V. Pachouri and N. Kathuria, "Forest Fire Detection via the Internet of Things-Based Systems," 2023 3rd International Conference on Pervasive Computing and Social Networking (ICPCSN), Salem, India, 2023, pp. 1144-1149, doi: 10.1109/ICPCSN58827.2023.00194.

[3] Y. Naing and T. L. Lai Thein, "Forest Fire Detection and Warning System for Disaster Prevention," 2023 IEEE Conference on Computer Applications (ICCA), Yangon, Myanmar, 2023, pp. 135-139, doi: 10.1109/ICCA51723.2023.10182015.

[4] C. F. I. Garcia and J. B. G. Ibarra, "Efficiency and Performance Evaluation of an Early Fire Detector Device Using an ESP32 Wireless Sensor Network," 2023 2nd International Conference on Vision Towards Emerging Trends in Communication and Networking Technologies (ViTECoN), Vellore, India, 2023, pp. 1-6, doi: 10.1109/ViTECoN58111.2023.10156954.

[5] N. Dharap, R. Porwal and S. Jindal, "Real-time Forest Fire Detection and Alert System Using Wireless Sensor Networks and Solar Energy," 2023 International Conference on Next Generation Electronics (NEleX), Vellore, India, 2023, pp. 1-5, doi: 10.1109/NEleX59773.2023.10420922.

[6] H. Jabnouni, I. Arfaoui, M. A. Cherni, M. Bouchouicha and M. Sayadi, "YOLOv6 for Fire Images detection," 2023 International Conference on Cyberworlds (CW), Sousse, Tunisia, 2023, pp. 500-501, doi: 10.1109/CW58918.2023.00086.

[7] S. Madkar, D. Y. Sakhare, K. A. Phutane, A. P. Haral, K. B. Nikam and S. Tharunyha, "Video Based Forest Fire and Smoke Detection Using YoLo and

CNN," 2022 International Conference on Power, Energy, Control and Transmission Systems (ICPECTS), Chennai, India, 2022, pp. 1-5, doi: 10.1109/ICPECTS56089.2022.10046717.

[8] K. Mehta, S. Sharma and D. Mishra, "Internet-of-Things Enabled Forest Fire Detection System," 2021 Fifth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), Palladam, India, 2021, pp. 20-23, doi: 10.1109/I-SMAC52330.2021.9640900.

[9] A. Ramelan, M. Hamka Ibrahim, A. Chico Hermanu Brillianto, F. Adriyanto, M. Rizqi Subeno and A. Latifah, "A Preliminary Prototype of LoRa-Based Wireless Sensor Network for Forest Fire Monitoring," 2021 International Conference on ICT for Smart Society (ICISS), Bandung, Indonesia, 2021, pp. 1-5, doi: 10.1109/ICISS53185.2021.9533237.

[10] Üstündağ and Ç. Gençer, "Control of automatic fire extinguishing operation for 5 forest areas with SCADA," 2021 5th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT), Ankara, Turkey, 2021, pp. 535-538, doi: 10.1109/ISMSIT52890.2021.9604546.

[11] T. Preeti, S. Kanakaraddi, A. Beelagi, S. Malagi and A. Sudi, "Forest Fire Prediction Using Machine Learning Techniques," 2021 International Conference on Intelligent Technologies (CONIT), Hubli, India, 2021, pp. 1-6, doi: 10.1109/CONIT51480.2021.9498448.

[12] S. Wang et al., "Forest Fire Detection Based on Lightweight Yolo," 2021 33rd Chinese Control and Decision Conference (CCDC), Kunming, China, 2021, pp. 1560-1565, doi: 10.1109/CCDC52312.2021.9601362.

[13] S. Wu and L. Zhang, "Using Popular Object Detection Methods for Real Time Forest Fire Detection," 2018 11th International Symposium on Computational Intelligence and Design (ISCID), Hangzhou, China, 2018, pp. 280-284, doi: 10.1109/ISCID.2018.00070.

[14] J. Zhang, W. Li, Z. Yin, S. Liu and X. Guo, "Forest fire detection system based on wireless sensor network," 2009 4th IEEE Conference on Industrial Electronics and Applications, Xi'an, China, 2009, pp. 520-523, doi: 10.1109/ICIEA.2009.5138260.

[15] D. Zhang, S. Han, J. Zhao1, Z. Zhang, C. Qu, Y. Ke, X. Chen, “Image Based Forest Fire Detection Using Dynamic Characteristics With Artificial Neural Networks”, International Joint Conference on Artificial Intelligence, 2009.

[16] Liyang Yu, Neng Wang and Xiaoqiao Meng, "Real-time Forest fire detection with wireless sensor networks," Proceedings. 2005 International Conference on Wireless Communications, Networking and Mobile Computing, 2005., Wuhan, China, 2005, pp. 1214-1217, doi: 10.1109/WCNM.2005.1544272.

Societal Relevance

The project concentrating on forest fire detection and notification systems bears great societal usefulness and relevance because of its ability to address a number of important issues:

- Environmental conservation: Natural ecosystems, ecosystem health, and biodiversity are all seriously threatened by forest fires. The initiative aids in the preservation of important forest ecosystems and the protection of endangered species by enabling early fire detection and quick reaction.
- Mitigation of Climate Change: Forest fires intensify climate change by releasing copious volumes of carbon dioxide into the atmosphere. The initiative contributes to the worldwide mitigation of climate change by lowering carbon emissions and assisting in the swift containment of flames.
- Disaster management and public safety: Early identification and notification of forest fires is essential for public safety as it enables authorities to quickly evacuate affected areas and reduce casualties. Through the project's timely notice system for communities at danger, disaster management activities are improved.
- Resource Preservation and Economic Stability: Communities that depend on forests may suffer large financial losses as a result of forest fires, which can destroy infrastructure, agricultural fields, and wood resources. The initiative helps protect these priceless resources and guarantees local communities' economic stability by enabling early diagnosis and containment.
- Policy Support and Decision-Making: Information gathered via the project's forest fire detection and warning system can help with the formulation of policies and the selection of course of action for land use planning, disaster risk reduction, and forest management. Policymakers may now implement evidence-based measures for both fire prevention and response thanks to the project's insightful analysis.
- International Cooperation and Exchange of Knowledge: The initiative promotes worldwide collaboration and knowledge exchange on best practices, technology, and

tactics for managing forest fires, given the global character of these events and their cross-border ramifications. The initiative supports international efforts to solve the problems caused by forest fires by encouraging collaboration between nations and stakeholders.

In summary, by addressing environmental, economic, and public safety problems, the project's focus on forest fire detection and alerting systems has great societal benefit and significance. The project supports resilience in the face of natural catastrophes and advances sustainable development goals by utilizing technology to improve early warning systems and response procedures.

CO-PO-MAPPING

				PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
VIII	ET2451 : Major Project	1	Design and analyze application based electronic systems.	3	3	3	3	3	3	3	3	3	3	3	3	3	3
		2	Implement core / multidisciplinary / industrybased electronics projects in cost effective manner.	3	3	3	3	3	3	3	3	3	3	3	3	3	3
		3	Communicate technical details effectively	3				3		3	3	3	3	3	3	3	3

APPENDIX

List of publications

Sr. No.	Authors	Title of paper	Name of International Journals/ International Conference	Place and Date of Publication with Citation Index
1	Dr. B. Y. Masram Prof. Aniket Pathade Anish Mendhe Ayush Mishra Nakshatra Shindekar Yash Kangale	A Smart Forest Fire Detection and Notification System using IOT and Machine Learning	International Conference on Electrical and Computing Technologies	Submitted

Project Group Information

Name: Anish Mendhe, Ayush Mishra, Nakshatra Shindekar, Yash Kangale

Mobile Number: 96890 86865, 74997 57079 ,9763469621, 88569 75812

Email ID: ankymendhe12@gmail.com, ayushmin1234@gmail.com, nakshatrashindekar44@gmail.com, yashkangale248@gmail.com

Project Preliminary Investigation Report

Name of Department:

Electronics and Telecommunications

Name of Project Guide:

Dr. B. Y. Masram

Students Details:

Roll No.	Name of Student	Email ID	Mobile No.
01	Ayush Mishra	ayushmin1234@gmail.com	7499757079
02	Anish Mendhe	ankymendhe12@gmail.com	9689086865
03	Nakshatra Shindekar	nakshatrashindekar44@gmail.com	9763469621
04	Yash Kangale	yashkangale248@gmail.com	8856975812

Title of the Project:

Forest Fire Detection and Notification System

Area of Project Work:

IOT and ML based

Problem Statement:

As of June 2023, 200k+ Forest Fires are detected in India Annually.

Traditional fire detection methods such as Patrols, Lookout towers require constant human supervision and hence, tend to be slower and very exhaustive.

Satellites Imagery provide a faster but costly alternative.

Modern AI based detection systems solve all of the above issues, however they create a lot of false positives.

Literature Review:

Title of Paper	Details of Publication with Date and Year	Literature Identified for Project
Real-time Forest Fire Detection with Wireless Sensor Networks. - Liyang Yu, Neng Wang, Xiaoqiao Meng	IEEE Date of Conference: 26-26 September 2005 Date Added to IEEE Xplore: 05 December 2005	Sensor Network for Forest Fire Detection. Clustering, Data Collection, Data Processing in Wireless Sensor Network.
Forest Monitoring System for Early Fire Detection Based on Convolutional Neural Network and UAV imagery	IEEE Date of Conference: 29-30 October 2020 Date Added to IEEE Xplore: 25 December 2020	Object Detection Based on Convolution Neural Network.
Forest fire detection system based on wireless sensor network	IEEE Date of Conference: 25-27 May 2009	Forest Fire Warning Forecast Model.

	Date Added to IEEE Xplore: 30 June 2009	

Current Limitations

1. Data Latency: Satellite-based systems may have a latency in data transmission, which means there can be a delay between the occurrence of a fire and the receipt of information by authorities.

2. Not all forested areas may be covered by detection systems, especially in remote or uninhabited regions, leaving these areas vulnerable to undetected fires.

Proposed Solution

A 2-Step Wildfire Detection System that uses Sensor Network for Realtime monitoring and Camera-Based Image Analysis to reduce the number of False Positives considerably.

Objectives and Scope of Work

Objectives:

1. **Early Detection**: Quickly identifying the presence of a fire in its early stages to prevent it from spreading uncontrollably.
2. **Rapid Response**: Promptly alerting authorities and firefighting teams to initiate a rapid response, including fire suppression and evacuation efforts.
3. **Minimizing Damage**: Reducing the extent of damage to forests, wildlife, property, and human lives by containing and extinguishing fires swiftly.
4. **Environmental Protection**: Preserving the ecosystem by preventing large-scale destruction of forests and the wildlife within them.
5. **Public Safety**: Ensuring the safety of nearby communities by alerting them to potential threats and facilitating evacuations if necessary.

Scope of Work:

Utilize a network of IoT sensors strategically placed in forested areas. These sensors could include temperature and humidity sensors, smoke detectors, air quality sensors, and video cameras. IoT sensors collect real-time data about environmental conditions in the forest. For instance, temperature spikes, increased humidity, or the presence of smoke. Transmit data wirelessly to a central monitoring system or cloud-based platform for analysis. This could involve wireless networking technologies like Wi-Fi, LoRa, or cellular networks. If a fire is detected, the system can automatically trigger alerts. This might include sending notifications to relevant authorities, nearby communities, and firefighters.

Feasibility Assessment:

I. Expected Outcomes of the Project

Accurate, Inexpensive, Low Power and Wide Range Fire Detection System able of differentiating between manmade and natural fires and notifying the fire department.

II. Innovation Potential

- Early Warning System using Machine Learning
- Mobile Application Deployment.

III. Task Involved

1. Realtime sensor data monitoring for abnormal behaviors.
2. Image analysis of the area around the sensor that triggers the camera.
3. Sending Notification to the fire department for Human Verification.

IV. Expertise Required

- Internet of Things.

- Data Science and Machine Learning.
- Embedded Systems.

V. Facilities Required

- Emergency response Coordination Centers
- Firefighting equipment and crew

Milestones and Time Plan

	Task	JULY 2020	AUG 2020	SEP 2020	OCT 2020	NOV 2020	DEC 2020	JAN 2021	FEB 2021	MAR 2021	APR 2021
Design	Conceptual Design	✓									
	Detailed design		✓								
	Design Modifications			✓							
	Final Design			✓							
Develop	Procurement (If any)			✓	✓						
	Prototyping				✓	✓					
	Modifications						✓				
Deliver	Testing and Validation						✓				
	Final Modifications							✓			
	IPR / patent draft								✓		
	Thesis and Poster								✓		

Name and Signature of Project Guide

Signature of HOD