IMAGE PROCESSING BASED FIRE DETECTION

MINI PROJECT

Submitted by

Name of the students	University Roll No
BOOPATHI RAJ V	731121106003
DHARANEESWARAN K	731121106008
GAYATHRI P	731121106013
JANANI V	731121106018
KATHIRIPERUMAL S	731121106024
LUTHDHRAN CS	731121106029
MUTHURAMALINGAM B	731121106034
PREETHIGA A	731121106037
SELVARAGAVAN K	731121106042
SUBHASREE S	731121106047
THARANI M	731121106052
ANBURAJ S	731121106303
SIVABALAN R	731121106308

UNDER THE PROVISION OF **DR.P.K.KOWSALYA**

Department of Electronics and Communication Engineering

Government College of Engineering-Erode,

638316.

Affiliated to AnnaUniversity,Chennai

ACKNOWLEDGEMENT

It is our great fortune that we have got opportunity to carry out this Project Work under the supervision of **Dr.Kowsalya P K** in the Department of Electronics and Communication Engineering, Government College of Engineering (IRTT), Erode, PO, near Vasavi College, Tamil Nadu-638316, affiliated to Anna University.We express our sincere thanks and deepest sense of gratitude to our guide for his constant support, unparalleled guidance and limitless encouragement.

We wish to convey our gratitude to **Dr.Kowsalya P K** Department of Electronics and Communication Engineering, Government College of Engineering (IRTT), to the authority of GCE for providing all kinds of infrastructural facility towards the research work.

We would also like to convey our gratitude to all teaching and non-teaching staff of the Department of Electronics and Communication Engineering GCE for there whole-hearted cooperation to make this work turn into reality.

Department of Electronics and Communication Engineering

Government College of Engineering-Erode

CERTIFICATE

To Whom it may concern

This to certify that the project work entitled **IMAGE PROCESSING BASED ON FIRE DETECTION** is the bonafide carried out by **BOOPATHI RAJ V**, **DHARANEESWARAN K,GAYATHRI P,KATHIRIPERUMAL S,LUTHDRAN C S,MUTHURAMALINGAM B,PREETHIGA A,SELVARAGAVAN K,SUBHASREE S,THARANI M,ANBURAJ S,SIVABALAN R**, students of B.E in the Dept. of Electronics and Communication Engineering, GCE, Po, Near vasavi College, Erode, Tamil Nadu-638316.Affilated to ANNA UNIVERSATY (CHENNAI), During the academic year 2023-2024.

----------------------------	---------------------------
Signature of Guide	Signature of HOD
Dr.P.K.Kowsalya	Dr.R.Valarmathi
Assistant Professor	Professor & HOD
ECE Department	ECE Department
GCE, Erode	GCE, Erode

TABLE OF CONTENT

S.No	TITLE	PAGE NO
1	ABBREVATIONS AND ACRONYMS	6
2	ABSTRACT	7
3	SCOPE OF WORK IN PROJECT	8
4	**CHAPTER 1:** INTRODUCTION	9
5	**CHAPTER 2:** WORKING PRINCIPLE 2.1: CIRCUIT DIAGRAM	10 11
6	**CHAPTER 3:** HARDWARE MODEL 3.1: BLOCK DIAGRAM 3.2: ESP32 MODULE 3.3: ESP32 CAM 3.4: FLAME SENSOR 3.5: LCD DISPLAY 3.6: BUZZER	12 13 14 15
7	**CHAPTER 4:** SOFTWARE PROGRAM MODE 4.1: PROGRAMING CODE 4.2 PROGRAM DISCRIPTION	16 23

8	**CHAPTER 5:** OUTPUT	24
9	**CHAPTER 6:** FUTURE ENHANCEMENT & CONCLUSION	25

ABBREVIATIONS AND ACRONYMS

ESP-32	Espressif System
ESP-32 CAM	Espressif-32 Camera
TFT	Thin Flim Transistor
LCD	Liquid crystal Display
VCC	Power supply
GND	Ground
I/O	Input and Output

ABSTRACT

Fire detection plays a crucial role in ensuring safety and minimizing damage in various environments. Traditional fire detection methods often rely on sensors, such as smoke and heat detectors, which may not always be effective, especially in complex environments or early stages of fire. Image processing-based fire detection has emerged as a promising alternative, leveraging the capabilities of computer vision and machine learning techniques to detect fires based on visual cues. This project presents a comprehensive review of image processing-based fire detection techniques, covering various aspects including dataset selection, preprocessing methods, feature extraction techniques, and classification algorithms. Additionally, challenges and future directions in the field are discussed.

SCOPE OF WORK IN THE PROJECT

This project aims to develop an innovative fire detection system that utilizes image processing techniques to analyze visual data from ESP32 CAM .

This could be particularly useful in scenarios where quick detection is crucial such as in industries, factory, forests, Electrical vehicle, hospital and buildings etc.

Deep Learning Models: Continued development of deep learning models, such as convolutional neural networks (CNNs), can improve the accuracy and efficiency of fire detection algorithms.

Real-time Detection: Enhancements in processing power and algorithms will enable real-time detection of fires, allowing for swift response and mitigation measures.

Multi-sensor Integration: Integration of image processing with other sensors, such as thermal and smoke detectors, can improve the robustness and reliability of fire detection systems.

Edge Computing: Deployment of image processing algorithms on edge devices like cameras or drones can reduce latency and enable fire detection in remote or inaccessible areas.

CHAPTER 1:

INTRODUCTION

In recent years, the devastating impact of fires on both human life and property has underscored the critical need for effective fire detection systems. Traditional methods, relying heavily on sensors and alarms, have limitations in terms of accuracy and response time, especially in large or complex environments. As a response, image processing-based fire detection systems have emerged as a promising solution, leveraging advancements in computer vision and artificial intelligence.

This project aims to develop an innovative fire detection system that utilizes image processing techniques to analyze visual data from surveillance cameras or other imaging devices.

The proposed system offers several advantages over conventional approaches. Firstly, it can detect fires in real-time, minimizing response times and potentially preventing catastrophic outcomes. Secondly, by analyzing visual data, the system can provide additional contextual information such as fire location, size, and spread pattern, aiding emergency responders in their decision-making process. Moreover, image processing-based fire detection systems can be integrated seamlessly into existing surveillance infrastructure, offering a cost-effective and scalable solution for fire monitoring and prevention.

In this project, we will explore various image processing techniques, including but not limited to background subtraction, color segmentation, and pattern recognition, to develop a robust fire detection algorithm. Additionally, we will train and fine-tune machine learning models using labeled datasets to enhance the system's accuracy and adaptability to different environments and fire scenarios.

Overall, this project represents a significant step towards leveraging cutting-edge technology to improve fire safety and emergency response capabilities. By harnessing the power of image processing and artificial intelligence, we aim to create a reliable and efficient fire detection system that can save lives and mitigate the impact of fire-related disasters.

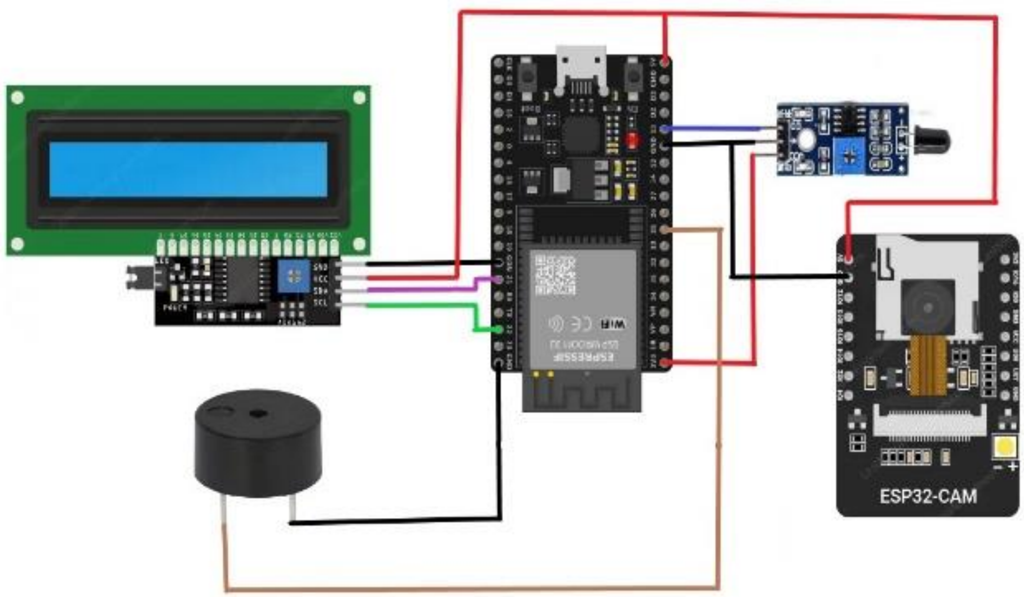
CHAPTER 2:

WORKING PRINCIPLE

Working of this system can be explained in the following steps:

- Capturing the images from camera (ESP32 CAM)
- Pre-processing the image
- Detecting the fire using image and flame sensor
- Streaming the live videos with help of web server (IP Address)
- Display alert message in LCD.
- Alert by using buzzer.

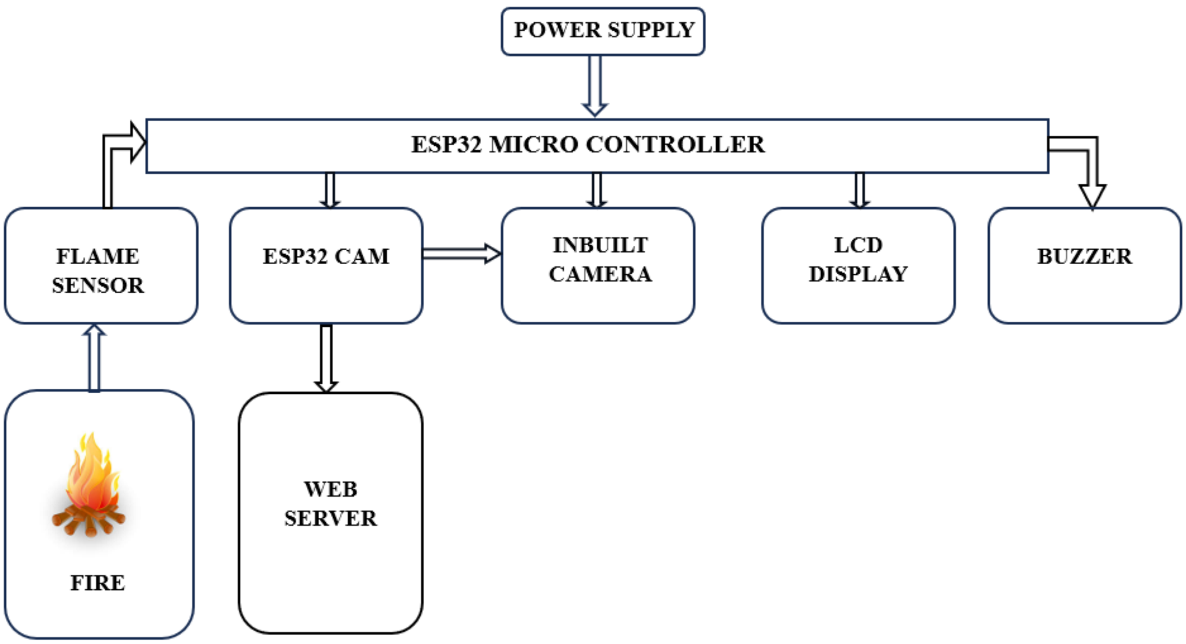
2.1:CIRCUIT DIAGRAM



CHAPTER 3:

HARDWARE MODEL

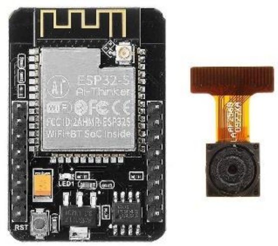
3.1: BLOCK DIAGRAM



3.2: ESP32 MODULE

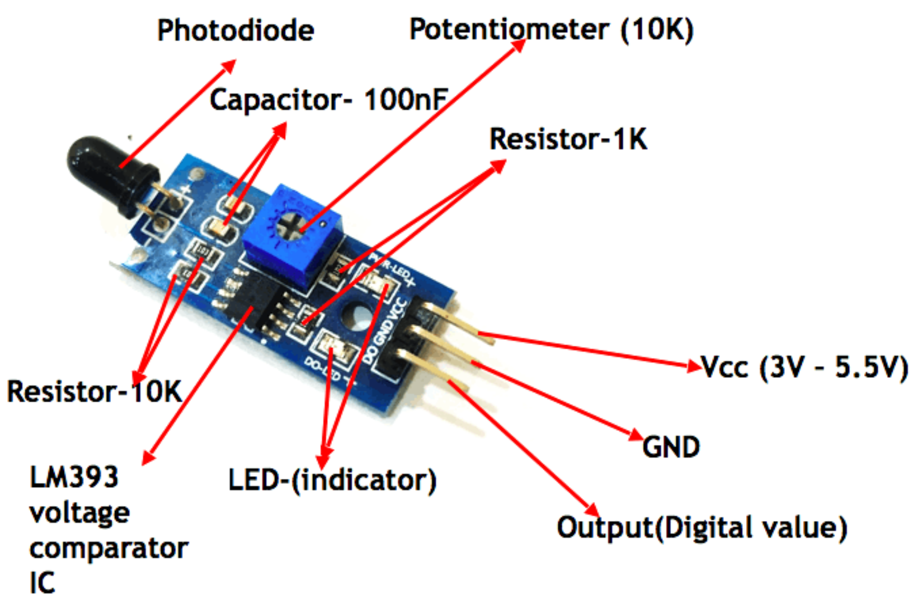
In an IoT project, an ESP32 module can be employed for tasks such as building a smart home automation system. The ESP32, equipped with Wi-Fi and Bluetooth, serves as a central controller, connecting and managing various smart devices like lights, thermostats, and sensors. It collects data from sensors, communicates with other devices through Wi-Fi, and can be remotely controlled via a smartphone app. The versatility, connectivity options, and processing power of the ESP32 make it an excellent choice for creating efficient and interconnected IoT solutions.

3.3: ESP32 CAM



The ESP32-CAM module is particularly well-suited for IoT projects involving visual data. For example, you could use it to develop a smart surveillance system. The ESP32-CAM captures images or video footage using its onboard camera and then processes or transmits this visual data. It can send alerts or store images on a cloud server based on certain conditions detected by the camera, providing a cost-effective solution for home security or monitoring applications within the Internet of Things realm. The ESP32-CAM's compact size and integrated camera make it an attractive choice for projects that involve visual information in IoT scenarios.

3.4: FLAME SENSORS:



A flame sensor is a crucial component in IoT projects focused on fire detection and safety. This sensor detects the presence of flames by responding to infrared radiation emitted during combustion. When integrated into an IoT system, the flame sensor can trigger alarms, notifications, or automatic responses to mitigate potential fire hazards. Its role in early fire detection makes it valuable for applications such as smart home safety systems or industrial environments, enhancing overall security and reducing the risk of fire-related incidents.

3.5: LCD DISPLAY:

An LCD 20x4 display is a versatile component in IoT projects, providing a 20-character by 4-line text interface for information output. Its clear and readable display is commonly used to present real-time data, sensor readings, or system status in IoT applications. Integrated into devices like weather stations or smart home control panels, the LCD 20x4 enhances user interaction by providing instant visual feedback. Its simplicity, coupled with a wide range of microcontroller compatibility, makes it a practical choice for conveying information in IoT projects where a visual interface is essential.

3.6: BUZZER:

A buzzer is a compact and audible signaling device commonly employed in IoT projects for alerting users to specific events or conditions. Integrated into systems such as smart home security or environmental monitoring, the buzzer provides an immediate and attention-grabbing security or environmental monitoring, the buzzer provides an immediate and attention-grabbing audio cue. It can be programmed to signal alarms, notifications, or status changes based on data from sensors or connected devices, enhancing the overall user awareness and interaction in the IoT environment. The simplicity and effectiveness of buzzers make them valuable for real-time audio feedback in various Internet of Things applications.

CHAPTER 4:

SOFTWARE PROGRAM MODEL

4.1: PROGRAM CODE

```
//ESP 32 MODULE
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 16, 2);
void setup() {
  pinMode(13, INPUT);
  pinMode(25, OUTPUT);
  Serial.begin(9600);
  lcd.init();
  lcd.backlight();
  lcd.setCursor(0, 0);
  lcd.print("Fire detection");
  lcd.setCursor(3, 1);
  lcd.print("projects");
  delay(2000);
}
void loop() {
  lcd.clear();
  int val=digitalRead(13);
  Serial.println(val);
  if (val == 0){
 Serial.println("Fire detected");
    digitalWrite(25, HIGH);
```

```
    lcd.setCursor(5, 0);
    lcd.print("Alert");
    lcd.setCursor(2, 1);
    lcd.print("Fire Detected");
  }
  else{
    Serial.println("NO Fire");
    digitalWrite(25, LOW);
    lcd.setCursor(5, 0);
    lcd.print("Safe");
    lcd.setCursor(0, 1);
    lcd.print("No Fire Detected");
  }
  delay(1000);
}
```

//ESP 32 CAM

```
#include "esp_camera.h"
#include <WiFi.h>

//
// WARNING!!! PSRAM IC required for UXGA resolution and high JPEG quality
//            Ensure ESP32 Wrover Module or other board with PSRAM is selected
//            Partial images will be transmitted if image exceeds buffer size
//
//            You must select partition scheme from the board menu that has at least 3MB APP space.
//            Face Recognition is DISABLED for ESP32 and ESP32-S2, because it takes up from 15
```

```
//            seconds to process single frame. Face Detection is ENABLED if PSRAM is enabled as
well

// ===================

// Select camera model

// ===================

//#define CAMERA_MODEL_WROVER_KIT // Has PSRAM

//#define CAMERA_MODEL_ESP_EYE // Has PSRAM

//#define CAMERA_MODEL_ESP32S3_EYE // Has PSRAM

//#define CAMERA_MODEL_M5STACK_PSRAM // Has PSRAM

//#define CAMERA_MODEL_M5STACK_V2_PSRAM // M5Camera version B Has PSRAM

//#define CAMERA_MODEL_M5STACK_WIDE // Has PSRAM

//#define CAMERA_MODEL_M5STACK_ESP32CAM // No PSRAM

//#define CAMERA_MODEL_M5STACK_UNITCAM // No PSRAM

#define CAMERA_MODEL_AI_THINKER // Has PSRAM

//#define CAMERA_MODEL_TTGO_T_JOURNAL // No PSRAM

//#define CAMERA_MODEL_XIAO_ESP32S3 // Has PSRAM

// ** Espressif Internal Boards **

//#define CAMERA_MODEL_ESP32_CAM_BOARD

//#define CAMERA_MODEL_ESP32S2_CAM_BOARD

//#define CAMERA_MODEL_ESP32S3_CAM_LCD

//#define CAMERA_MODEL_DFRobot_FireBeetle2_ESP32S3 // Has PSRAM

//#define CAMERA_MODEL_DFRobot_Romeo_ESP32S3 // Has PSRAM

#include "camera_pins.h"


// ===========================

// Enter your WiFi credentials

// ===========================

const char* ssid = "*********";
```

```
const char* password = "********";
void startCameraServer();
void setupLedFlash(int pin);
void setup() {
  Serial.begin(115200);
  Serial.setDebugOutput(true);
  Serial.println();
  camera_config_t config;
  config.ledc_channel = LEDC_CHANNEL_0;
  config.ledc_timer = LEDC_TIMER_0;
  config.pin_d0 = Y2_GPIO_NUM;
  config.pin_d1 = Y3_GPIO_NUM;
  config.pin_d2 = Y4_GPIO_NUM;
  config.pin_d3 = Y5_GPIO_NUM;
  config.pin_d4 = Y6_GPIO_NUM;
  config.pin_d5 = Y7_GPIO_NUM;
  config.pin_d6 = Y8_GPIO_NUM;
  config.pin_d7 = Y9_GPIO_NUM;
  config.pin_xclk = XCLK_GPIO_NUM;
  config.pin_pclk = PCLK_GPIO_NUM;
  config.pin_vsync = VSYNC_GPIO_NUM;
  config.pin_href = HREF_GPIO_NUM;
  config.pin_sccb_sda = SIOD_GPIO_NUM;
  config.pin_sccb_scl = SIOC_GPIO_NUM;
  config.pin_pwdn = PWDN_GPIO_NUM;
  config.pin_reset = RESET_GPIO_NUM;
  config.xclk_freq_hz = 20000000;
```

```
  config.frame_size = FRAMESIZE_UXGA;
  config.pixel_format = PIXFORMAT_JPEG; // for streaming
  //config.pixel_format = PIXFORMAT_RGB565; // for face detection/recognition
  config.grab_mode = CAMERA_GRAB_WHEN_EMPTY;
  config.fb_location = CAMERA_FB_IN_PSRAM;
  config.jpeg_quality = 12;
  config.fb_count = 1;

  // if PSRAM IC present, init with UXGA resolution and higher JPEG quality
  //                      for larger pre-allocated frame buffer.
  if(config.pixel_format == PIXFORMAT_JPEG){
    if(psramFound()){
      config.jpeg_quality = 10;
      config.fb_count = 2;
      config.grab_mode = CAMERA_GRAB_LATEST;
    } else {
      // Limit the frame size when PSRAM is not available
      config.frame_size = FRAMESIZE_SVGA;
      config.fb_location = CAMERA_FB_IN_DRAM;
    }
  } else {
    // Best option for face detection/recognition
    config.frame_size = FRAMESIZE_240X240;
#if CONFIG_IDF_TARGET_ESP32S3
    config.fb_count = 2;
#endif
  }
```

```
#if defined(CAMERA_MODEL_ESP_EYE)
  pinMode(13, INPUT_PULLUP);
  pinMode(14, INPUT_PULLUP);
#endif

  // camera init
  esp_err_t err = esp_camera_init(&config);
  if (err != ESP_OK) {
    Serial.printf("Camera init failed with error 0x%x", err);
    return;
  }

  sensor_t * s = esp_camera_sensor_get();
  // initial sensors are flipped vertically and colors are a bit saturated
  if (s->id.PID == OV3660_PID) {
    s->set_vflip(s, 1); // flip it back
    s->set_brightness(s, 1); // up the brightness just a bit
    s->set_saturation(s, -2); // lower the saturation
  }
  // drop down frame size for higher initial frame rate
  if(config.pixel_format == PIXFORMAT_JPEG){
    s->set_framesize(s, FRAMESIZE_QVGA);
  }

#if defined(CAMERA_MODEL_M5STACK_WIDE) ||
defined(CAMERA_MODEL_M5STACK_ESP32CAM)
  s->set_vflip(s, 1);
```

```
  s->set_hmirror(s, 1);
#endif

#if defined(CAMERA_MODEL_ESP32S3_EYE)
  s->set_vflip(s, 1);
#endif

// Setup LED FLash if LED pin is defined in camera_pins.h
#if defined(LED_GPIO_NUM)
  setupLedFlash(LED_GPIO_NUM);
#endif

  WiFi.begin(ssid, password);
  WiFi.setSleep(false);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");

  startCameraServer();

  Serial.print("Camera Ready! Use 'http://");
  Serial.print(WiFi.localIP());
  Serial.println("' to connect");
```

```
}
void loop() {
  // Do nothing. Everything is done in another task by the web server
  delay(10000);
}
```

PROGRAM DESCRIPTION:

Fire Detection:

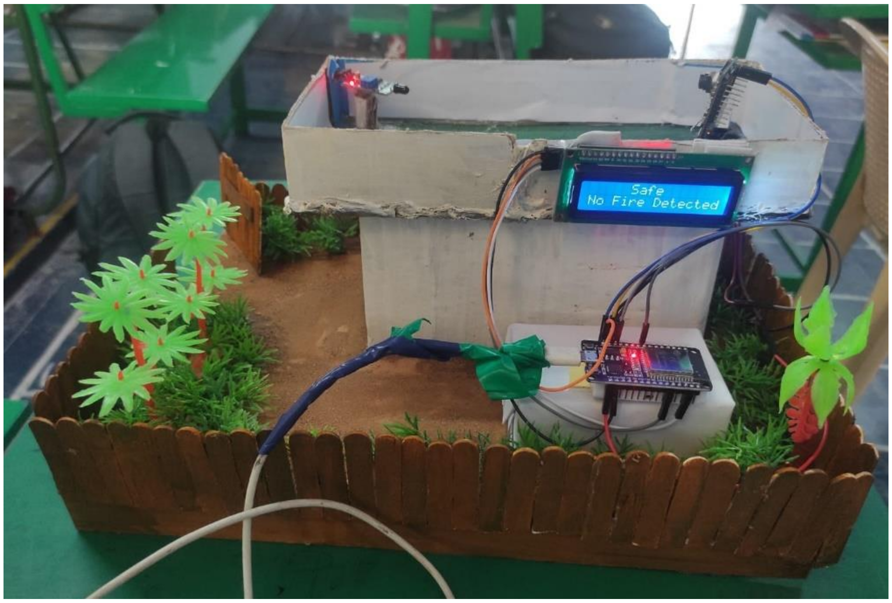
It uses an Arduino-compatible board with an LCD display (LiquidCrystal_I2C) and a digital pin (13) connected to a fire sensor.In the setup function, it initializes the pins, LCD, and prints a startup message on the display.The loop function continuously reads the state of the fire sensor. If the sensor detects fire (val == 0), it prints "Fire detected" and sets the LCD display to show an alert and "Fire Detected" message. Otherwise, it prints "NO Fire" and sets the display to show "Safe" and "No Fire Detected".

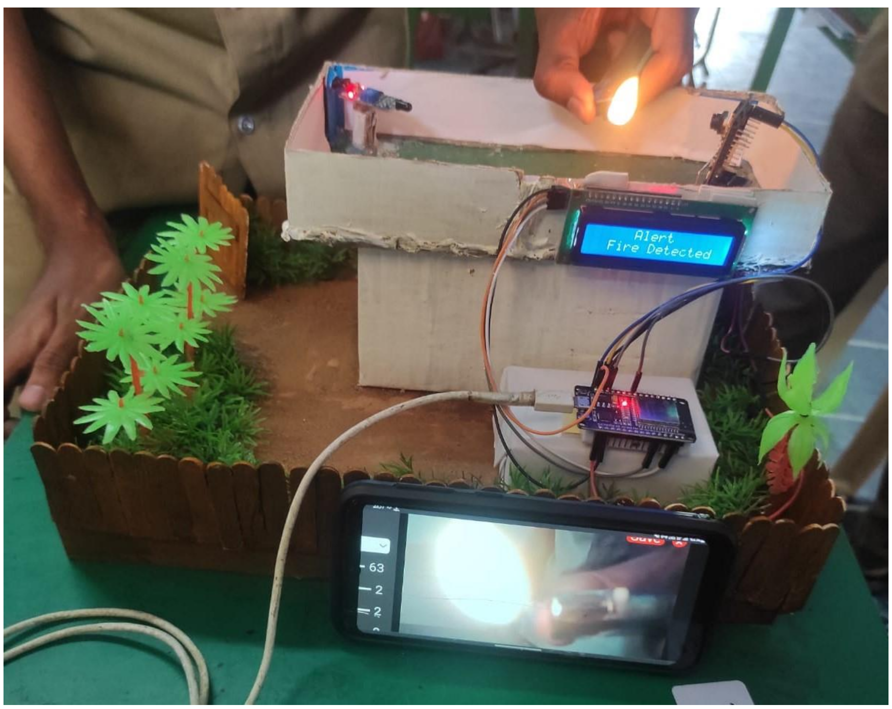
Camera Server Setup:

It includes configuration for an ESP32-based camera module (AI Thinker) and connects it to a WiFi network. The setup function initializes the camera module, sets up WiFi connection, and starts a camera server.The loop function does nothing as the camera server functionality is handled in another task.

CHAPTER:5

OUTPUT:





CHAPTER:6

FUTURE ENHANCEMENT & CONCLUSION:

In the future, advancements in image processing for fire detection are likely to focus on real-time analysis, enhanced accuracy, and expanded capabilities. Utilizing deep learning algorithms and convolutional neural networks (CNNs), these systems will be able to detect fires with greater precision, even in challenging environments such as low-light conditions or areas with high levels of smoke. Integration with IoT devices and drones could enable proactive monitoring of large areas, allowing for early detection and swift response to potential fire hazards.

Additionally, the integration of thermal imaging technology and multispectral analysis may further improve detection accuracy by identifying heat signatures and distinguishing between different types of fires.

Overall, future image processing technologies for fire detection are poised to revolutionize fire safety measures, offering more reliable and efficient methods for early detection and prevention.