IMPLEMENTATION OF WORKPLACE MONITORING AND SAFETY SYSTEMS USING IoT

A PROJECT REPORT

Submitted by

RANALD NIVETHAN X. N (312217106129)

SENTHIL KUMAR M (312217106148)

SURAJ VINOD (312217106170)

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

SRI SIVASUBRAMANIYA NADAR COLLEGE OF ENGINEERING
KALAVAKKAM 603 110

ANNA UNIVERSITY : CHENNAI 600 025

JUNE 2021

ANNA UNIVERSITY : CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report "**IMPLEMENTATION OF WORKPLACE MONITORING AND SAFETY SYSTEMS USING IoT**" is the bonafide work of "**Ranald Nivethan X.N. (31221706129), Senthil Kumar M. (31221706148) and Suraj Vinod (312217106170)**" who carried out the project work under my supervision.

SIGNATURE	**SIGNATURE**
Dr. S. RADHA	**Dr. C. ANNADURAI**
HEAD OF THE DEPARTMENT	**SUPERVISOR**
PROFESSOR	**ASSOCIATE PROFESSOR**
Department of Electronics and Communication Engineering, SSN College of Engineering, Kalavakkam - 603 110.	Department of Electronics and Communication Engineering, SSN College of Engineering, Kalavakkam - 603 110.

Submitted for the project viva-voce examination held on …05-08-2021……

INTERNAL EXAMINER	**EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

We thank **Dr. Shiv Nadar**, Chairman of SSN Institutions for extending a generous hand in providing the best of resources to the college. We also thank **Dr. V.E. Annamalai**, the esteemed Head of our Institution who has been a source of motivation to all the staffs and students of our college.

Our sincere thanks are due to **Dr. S. Radha**, the Head of the Department for her continuous support and motivation throughout our project.

We express our sincere thanks to the project coordinators in our panel **Dr. P. Vijayalakshmi** and **Dr. S. Hanis** for holding interactive review sessions and providing unique perspectives on our project.

We are indebted to our guide **Dr. C. Annadurai**, for his invaluable advice and guidance and for giving us innovative suggestions for the project. We express our gratitude for his patience and support.

We take this opportunity to thank all the family members, staff members, lab attendants and our peers of the Department of Electronics and Communication Engineering for their unwavering support for the completion of this project.

Ranald Nivethan X. N

Senthil Kumar M

Suraj Vinod

ABSTRACT

The continuous and rapid development in the facilities in the workplace eventually calls for safety of the workplace premises as well as improved monitoring system. For instance, an intruder alert will be sent even if a client enters the premises. To eradicate this issue, to send alert notifications only when required that is, during an intruder detection or mishap detection. The data is collected by the Raspberry Pi using the sensors interfaced to it. By employing the usage of IoT, data received from the sensors are sent to an IoT platform from where the information is passed as a notification through an email. The detected face from the video recorded by PiCam is sent to a local server using socket programming. Face detection and recognition is performed in the local server using Haar cascade in which there are three fundamental stages namely integral image calculation for extracting the features efficiently from the image, followed by executing adaboost algorithm for training weak classifiers and finally followed by cascade of weak classifiers to detect the face and followed by execution of LBPH algorithm for extracting the gray scaled histograms of the person to be recognized and then compared with existing histograms of the persons stored in the database. In case of an intruder detection, an e-mail notification is sent to the user. Similarly, when an accident or disaster is detected such as a fire accident or air pollution an alert notification is sent to the user through an e-mail and also the real time sensor data is displayed in the application.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

PPM	Parts per million
CO_2	Carbon di oxide
CO	Carbon monoxide
OpenCV	Open-source computer vision library
CNN	Convolutional neural networks
HTTP	Hypertext transfer protocol
IOT	Internet of things
API	Application programming interface
APP	Application
LAN	Local area network
SMS	Short Message service
PIR	Passive infrared sensor
RBM	Restricted Boltzmann machine
LCD	Liquid crystal display
GPIO	General purpose input/output
MOS	Metal oxide semiconductor
LPG	Liquified petroleum gas
NH_3	Ammonia
LED	Light emitting diode
SnO_2	Stannic oxide
USB	Universal serial bus
RAM	Random Access memory
HDMI	High definition multimedia interface
SSH	Secured socket shell
LBPH	Local binary patterns histogram

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

Monitoring of activities, behaviour or information which are used for influencing managing or information gathering is called surveillance. Surveillance can be achieved by intercepting electronic transmitted information or by means of observing the object or person from a long distance with the help of an electronic equipment. It can also be achieved by using other methods such as postal interception or gathering human intelligence.

In order to keep the office premises, secure it is very common for companies and owners of business to use surveillance camera. Today, there are many types of home video surveillance units to choose from which are not cost efficient and do not include automated alerts relating to workplace safety therefore the continuous and rapid development in the facilities in the workplace eventually calls for the need of safety of the workplace premises as well as monitoring system that is cost effective and efficient, so achieving a prototype to solve this issue is necessary.

1.2 CURRENT SCENARIO

In the present world the need for safety and security in workplace or an industry has become utmost importance. The word safety means integrated protection of the organization and not just merely physical possessions. Various devices have been developed with serve the rising need for safety. All the products were designed to serve a single purpose only and the majority of the devices were designed only to ensure the security of a workplace or an industry or even houses. The most widely used device currently is the surveillance camera which helps a person monitor their office premise without a need for human to be present in that place. This still remains as the largely preferred device for security but, as far as the entirety of safety is concerned an integrated safety system is the most suitable. An integrated safety system helps in efficiently safeguarding the workplace in all areas possible.

Recently, implementation of IoT in safety systems have been able to provide the user an easier access in monitoring their premises as well as getting to know the state of their establishment. Although it has been helpful in many aspects there are still areas that can be further improved making the safety system even better.

1.3 MOTIVATION AND OBJECTIVE

The system proposed here makes use of sensors such as smoke sensor, air quality sensor and pi camera. These help the user know the state of their premises that includes feature like intruder alert, smoke detection, air pollution detection. The objective of the project includes to provide a cost effective automated integrated safety management system for industries and organization, to manage security system using IoT, Computer Vision and machine learning and to additionally, to provide an accident alert system. It is also necessary to to provide instant notifications to user regarding the state and safety of their organisation via email and real time sensor data using graphs via application.

1.4 ORGANIZATION OF THE REPORT

After the introduction chapter, the rest of the report is organized as follows:

Chapter 2 introduces the reader various research works and systems developed individually for face recognition and detection systems, IoT bases surveillance systems and sensor based air quality systems. **Chapter 3** explains about the proposed system block diagram and the working description of the various parts of the system. **Chapter 4** gives the details about the hardware components and its specifications. **Chapter 5** provides the details of the various software components used in the systems and it’s details. **Chapter 6** provides the details of the various results that were achieved on the completion of the project. Finally, **chapter 7** discusses about the future work and the conclusion.

CHAPTER 2

LITERATURE SURVEY

This chapter summarises the findings from a literature review for analysing the various strategies associated with face detection and recognition algorithms, IoT based surveillance systems and sensor based systems. Based on the literature survey done we have developed a prototype that integrates safety and monitoring systems.

The scientific papers are subdivided into the following sections

• face detection and recognition systems

• IOT based surveillance systems

• IoT systems using sensors

2.1 FACE DETECTION AND RECOGNITION SYSTEMS

Paul viola et.al(2001) has proposed a face detection framework which achieves very high detection rates and the processing of images takes place extremely rapidly in this system. There are three main areas in which the system can be described. The first is called the integral image which is nothing but a new image is prepared from the input image which can be used to compute the necessary features very quickly from the image by the detector. Next is the AdaBoost learning algorithm is used to build a classifier which is extremely simple yet very efficient. The last is about building an efficient classifier which helps to remove background regions of an image and select more promising face regions on which more computation time is spent, the classifier built is using cascade of simple classifiers which in turn were built using the second stage. The features that are used are called the haar basis function. Using only constant number of basic operations the integral image can be computed. Using the integral image the haar functions can be computed in constant time at any location in the image. The training of cascade of classifiers have a trade off , on one hand more the number of features involved the better is the detection rates and false positive rate but as the number of features increase the time required to compute them also increases hence rapid detection becomes an issue.

Xiaogang Wang et.al (2013) proposed a face verification model which operates in wild environmental conditions that uses hybrid of restricted Boltzmann machine and convolutional neural network. The features that indicate similarities of person which are called relational visual features are directly learned from raw pixels of face pairs with the neural network. To extract the relational visual features from two different images of faces deep neural networks are used which jointly extract it. For extracting the global and high level features, the relational features are sent through multiple higher layers in the neural network. In order the achieve similarities of faces from different scenarios and robustness different groups of neural networks are constructed. Restricted Boltzmann machine which has two level average pooling hierarchy is now used to process the high level complementary features that were extracted from multiple groups of neural networks and give inference. For the task of face verification this hybrid deep neural network now jointly optimized by fine tuning.

R. Chellappa et.al (1995) proposed a survey on various different face recognition techniques and its applications to various sectors. The various techniques focussed on includes segmentation, feature extraction and recognition aspects of face recognition problem, using information drawn from intensity and range images of faces and profiles. Different techniques such as static matching, dynamic matching are also used. The techniques discussed also involve global transform and feature based methods using statistical, structural and neural classifiers. It also analyses the performance of listed techniques based on the parameters such as data collection, performance metrics, uniformity, feasibility of resampling, similarity measures and rank statistic comparison. The face recognition problem can be broadly described as to identify as many persons it can from a scene or image or a video from a stored set of already recognised faces. For the sake of narrowing the search race and gender which are called collateral information can also be used. The flow of face recognition is extracting the face regions from a general scene by discarding the background non face like regions from generic images then matching and identification of the face. The face recognition problem is a central issue in browsing through a database of images or electronic line up.

2.2 IOT BASED SURVEILLANCE SYSTEMS

Sonali P.Gulve et.al (2017) proposed a system which makes use of PIR sensor to make the working of a surveillance camera more efficient. It is a password protected system, once the password is entered the system is made active and if any motion is detected a SMS is sent to the client and the video captured is sent to the client through an e-mail. It uses Arduino board for normal mode and Raspberry Pi for surveillance mode. Once the password is entered the Arduino turns off all the electrical appliances by customizing coding with specific appliances. The system also monitors the activity and intimates the presence of any person in the premises.

The system is activated by a person leaving the premises by entering the password. The systems starts working by first detecting the motion of any human followed by counting the number of persons in the premises and also the presence of any unidentified person gets notified to the owner. Power consumption is also optimized by switching off any part of the system which is not in use for a long time, this is implemented apart from the security aspect hence wastage is prevented.

Ishita Gupta et.al(2016) proposed a system which is about replacing face recognition as a replacement for usage of passwords and RF ID cards for gaining access to office premises and security buildings. The system is made highly cost effective and very easy to use at the same time maintaining high performance by deploying the raspberry pi. The opening and closing of the doors is controlled by raspberry pi which makes this happen by comparing the detected face images with already stored database of names and if the name matches it is displayed in LCD and doors open. For identifying and giving a person access to the premises four stages namely capture, extraction, comparison and match or non match stage are performed. During the capture stage enrolment of a person with the system a physical or behavioural sample is taken which are also used for the verification and identification of a person. During the extraction stage a template is created by extracting unique data from the sample. Then in the comparison stage the template is then used in comparing with the existing sample. Then to finally determine if it is match or non match the comparison whether the extracted features are the same as that for the captured images of that particular person is performed.

Pravin Kumar.S et.al(2019) proposed a system that makes use of fusion of convolutional neural network and Gabor filter which are in turn used in the detection of faces. The important facial features are extracted using the Gabor filter and the model is trained using convolutional neural network. System is made cost efficient by executing the weight files in raspberry pi. For face detection the algorithm first detects the facial regions from the given input of images or video and then the detected faces are cleaned up with various filters and then further processing takes place. For face recognition the detected faces from the face detection system is given as input to the face recognition. The given input is then checked with faces that are available in the database to determine the un recognized face. The difference between face detection and recognition is that in detection, it determine whether there is a face in the input image or video, but in the face determine whose face it is.

2.3 IOT BASED SYSTEMS USING SENSORS

Mohammad Mohiddin et.al(2018) proposed a system about how to make a device using smart phone and Arduino to detect gases such as carbon dioxide, carbon monoxide and ammonia present in the environment. Additionally also the temperature and humidity of the environment can be measured using the device. The data from the sensors is received in smart phone and laptop, also the entire node consists of dht-11 , mq-135 and Arduino board. The total system is effective and cost efficient in monitoring, detecting and measuring the green house gases in the environment continuously.

Riteeka Nayak et.al(2017) proposed a system about how pollution has lot of harmful effects on human well being and directly affects the health of people who are exposed to it. The pollution has increased in recent times due to a lot of factors such as industrialization, urbanization and increased vehicle use and also increased population. In order to monitor this the system is developed which is an air quality monitoring system based on IOT in which the air quality is monitored over an web server using the internet and when concentration of gases such as carbon dioxide, smoke, benzene or ammonia is beyond a certain level and alarm will be sent to the user regarding this. Along with this the ppm values of the corresponding gases are sent along with the alert which is then displayed in LCD for easy monitoring purposes. System uses mq135 which is arguably the go for choice for air quality monitoring as it can detect all the gases accurately and also the amount off it. Using the system the air quality can be monitored from anywhere by the use of computer or mobile.

CHAPTER 3

SYSTEM COMPONENTS

In this chapter we describe the various system components required to design and implement the protype for monitoring and safety systems also describe the use of the components to serve the desired purpose and to integrate the stand alone components as a complete system.

The hardware components used in this project include the following:

• Smoke sensor-mq2

• Air Quality sensor- mq135

• Pi camera

• Raspberry Pi v3

• Micro Secure Digital card-32GB

3.1 SMOKE SENSOR MQ-2

MQ2 is one of the MQ series of sensors that are available and are the most widely used gas sensors. It is a MOS type sensor or metal oxide semiconductor type sensor the detection of gases happens by the change of resistance hence knowns as chemiresistors as the gas reacts with sensor material and resistance changes thus detecting the gases. Voltage divider network is used to measure the increasing resistance.

Fig. 3.1 Internal structure of mq2

MQ2 gas sensor needs 5V Dc as input and it also draws a power of 800mW. The gases that can be detected should have concentration from 200 to 10000ppm and should be LPG smoke, alcohol, propane, hydrogen, methane or carbon monoxide. A fine stainless steel mesh called anti explosion network is used to enclose the sensor. This coating is used to ensure that the internal heating element of the sensor is prevented from getting exploded as the gases are being sensed.

This coating also ensures that only gases are able to get inside the chamber by filtering out the suspended particles in the coating, additionally providing security to the sensor. The mesh is bound to the the body of the sensor using a clamping ring made of plated copper as shown in Fig. 3.1.

The analog to digital conversion is as follows:

Initially the *Rs* and *Ro* are calculated, *Ro* is the value of resistance in fresh air and Rs is the value of resistance in the presence of any detectable gas.

$$Rs = \left(\frac{Vc}{V_{Rl}} - 1\right) Rl \tag{3.1}$$

To represent in digital form Concentration of gas(in ppm)= *Rs/Ro*

3.2 AIR QUALITY SENSOR – MQ135

Mq gas series sensors are the most commonly used when comes to measuring and detecting a wide range of gases and mq135 is one of the mq series sensors. Mq 135 is commercially available as a stand alone sensor or a complete module. If the requirement is only detect the presence of gas and not the ppm values of the amount of gases detected then it is advisable to buy mq135 as module that comes along with an digital output pin and operational amplifier comparator, but if the requirement is to measure the ppm values also it is better to buy the sensor as a stand alone sensor. The MQ-135 Gas sensors are deployed in air quality control equipment and are also suitable for detecting or measuring of Benzene, Smoke, CO_2, CO NH_3, NOx, Alcohol. The mq135 sensor when bought as a module comes with the digital pin which helps sensor operate in the absence of a microcontroller and can be used when we only have to detect the gases. Analog pin is needed along with the sensor if we have to detect and measure the ppm of the gases as shown in Fig. 3.2. The analog pin works based on the TTL logic and needs 5V input so that it can be used with any microcontroller.

The analog and the digital pin both can be used for this purpose. When powered up the module with 5V the power LED begins to glow and if the gases are not detected the output LED remains turned off which means the digital pin will be at 0V. Before the sensor actually starts detecting the gases the sensors have to be on for a some time called the pre heating time. After pre heating the sensor now if the gases are measured the output digital pin goes high and also the output LED glows also make use of potentiometer till the output pin becomes high. Thus whenever the gas is introduced to the particular gas at the desired concentration the digital pin should go high otherwise it will remain the same.

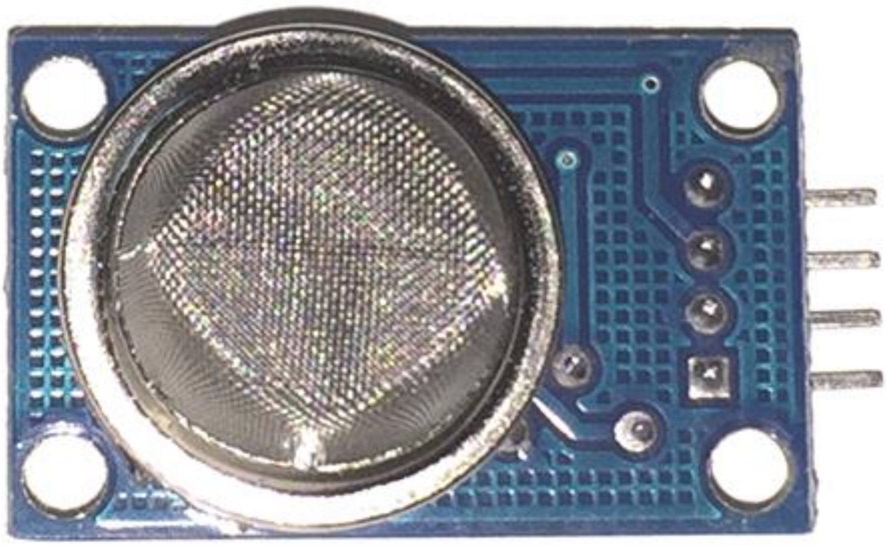

Fig. 3.2 Internal Structure of mq135

The gas sensing material used in mq135 is SnO_2 which has very high resistance in the presence of clean air. As the polluting gases are detected the gas sensing material resistance decreases. The Rs/Ro vs ppm graph gives the measure of the ppm values of the gases detected using mq135 sensor. Resistance of sensor is *Rs*:

$$Rs = \left(\frac{Vc}{V_{Rl}} - 1\right) Rl \qquad (3.2)$$

3.3 PI CAMERA

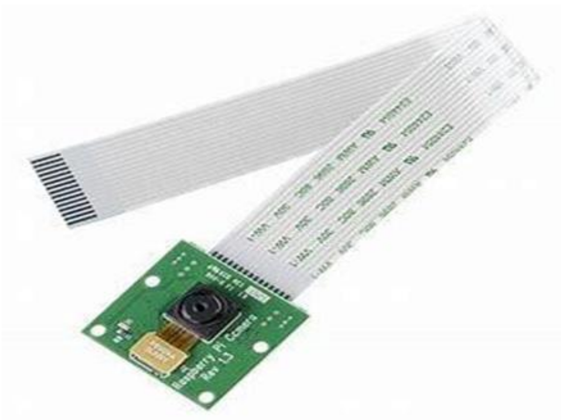

Fig. 3.3 Pi camera

Raspberry Pi supports a portable camera module which is called pi camera. MIPI is camera serial interface protocol using which raspberry Pi communicates with the camera as shown in Fig. 3.3. This can be used image processing, surveillance or machine learning projects. payload of camera is very less in surveillance drones hence more commonly used in this. Even normal USB cams that are with computer can also used as pi modules.

3.4 RASPBERRY PI V3

The Raspberry Pi is a single board computer that plugs into a computer monitor or TV and uses a standard keyboard and mouse. Raspberry Pi's key features are

Table 3.1 Pin Description

Pin Number	Pin Name	Description
1,4,7,10	Ground	System Ground
2,3	CAM1_DN0, CAM1_DP0	MIPI Data Positive and MIPI Data Negative for data lane 0
5,6	CAM1_DN1, CAM1_DP1	MIPI Data Positive and MIPI Data Negative for data lane 1
8,9	CAM1_CN, CAM1_CP	These pins provide the clock pulses for MIPI data lanes
11	CAM_GPIO	GPIO pin used optionally
12	CAM_CLK	Optional clock pin
13,14	SCL0, SDA0	Used for I2C communication
15	+3.3V	Power pin

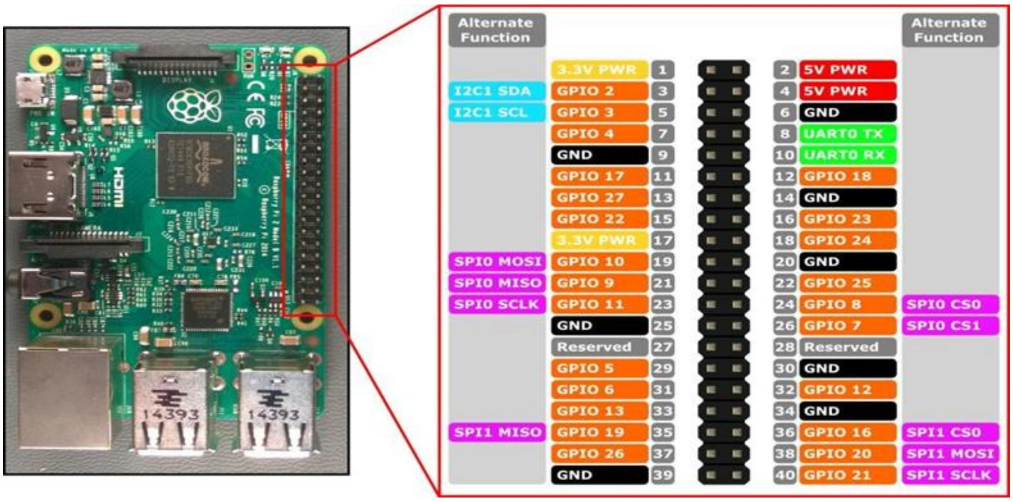

Fig. 3.4 Raspberry Pi v3 and its pin configuration

a high-performance dual-display support at resolutions up to 4K via a pair of micro-HDMI ports, 64-bit quad-core processor , dual-band 2.4/5.0 GHz wireless LAN, Bluetooth 5.0, Gigabit Ethernet, USB 3.0, and PoE capability (via a separate PoE HAT add-on). hardware video decode at up to 4Kp60, up to 4GB of RAM as shown in Fig. 3.4.

3.4.1 TECHNICAL SPECIFICATION

• Broadcom BCM2387 chipset
• 1.2GHz Quad-Core ARM Cortex-A53
• 802.11 bgn Wireless LAN and Bluetooth 4.1 (Bluetooth Classic and LE)
• 1GB RAM
• 64 Bit CPU
• 4 x USB ports
• 4 pole Stereo output and Composite video port
• Full size HDMI
• 10/100 BaseT Ethernet socketbr
• CSI camera port for connecting the Raspberry Pi camera
• DSI display port for connecting the Raspberry Pi touch screen display
• Micro SD port for loading your operating system and storing data
• Micro USB power source

3.4.2 FEATURES

• Now 10x Faster - Broadcom BCM2387 ARM Cortex-A53 Quad Core Processor powered Single Board Computer running at 1.2GHz!
• 1GB RAM so you can now run bigger and more powerful applications
• Fully HAT compatible
• 40pin extended GPIO to enhance your "real world" projects.
• Connect a Raspberry Pi camera and touch screen display (each sold separately)

• Stream and watch Hi-definition video output at 1080

• Micro SD slot for storing information and loading your operating systems.

• 10/100 BaseT Ethernet socket to quickly connect the Raspberry Pi to the Internet

3.5 CIRCUIT DIAGRAM

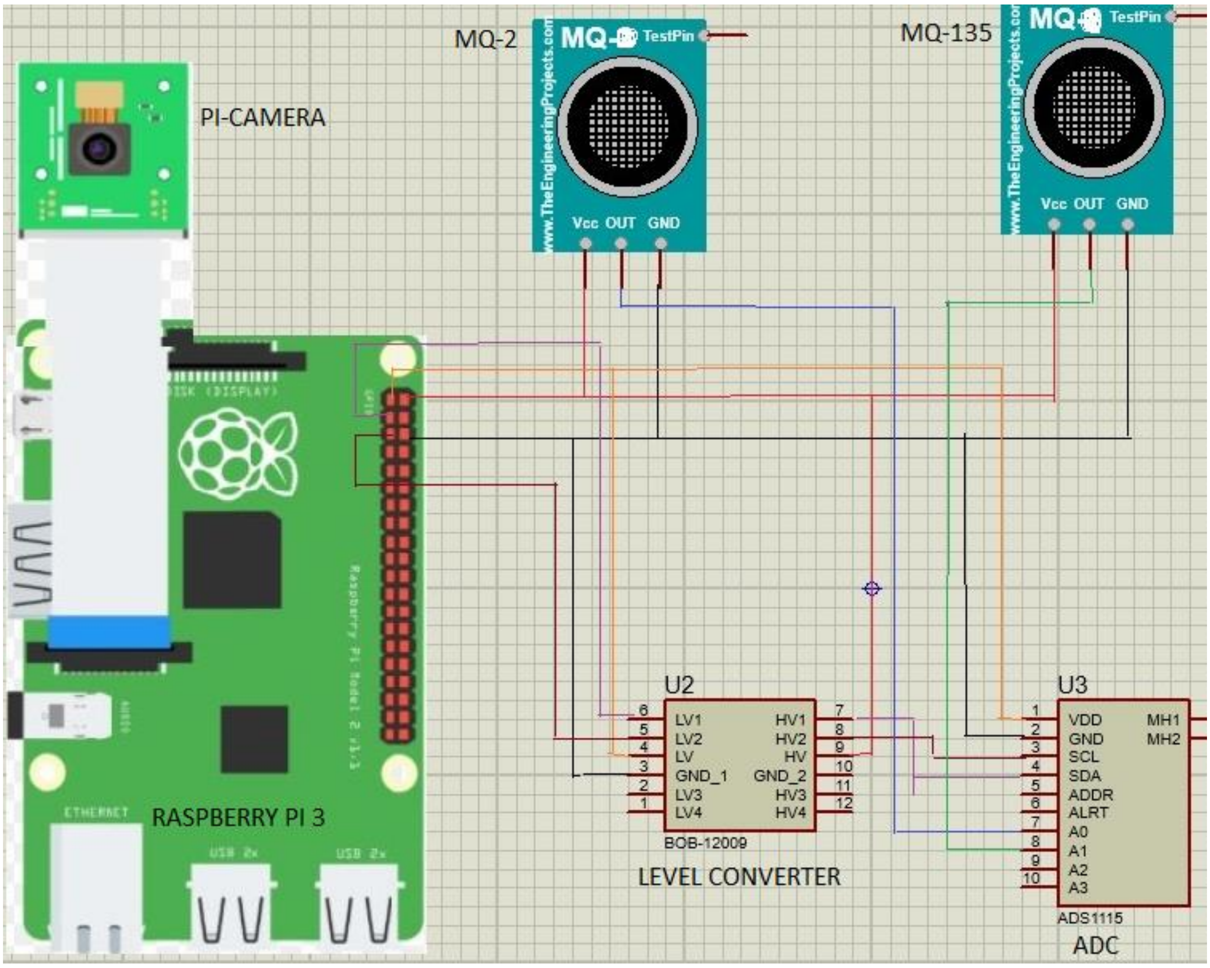

Fig 3.5 Circuit diagram of the system

Fig 3.5 shows how 5V input from the raspberry Pi is provided to the inputs of the level convertor and sensors mq2 and mq135 respectively and 3.3V input is given to ADS1115. ADS1115 is an analog to digital convertor and is used because the outputs obtained from the sensors namely mq2 and mq135 are raw analog values therefore ADS1115 converts this output to digital outputs. Outputs of the sensors is provided to ADS1115 analog to digital convertor pins A0 and A1. SDA and SCL pin outputs are 5V which needs to be down converted to 3.3 V as the GPIO pins of raspberry Pi v3 gets damaged with voltages above 3.3 V so the SDA and SCL outputs is given to 4 channel logic level convertor and then provided to the raspberry Pi.

CHAPTER 4
SOFTWARE PACKAGES

In this chapter we describe the various software packages required to design and implement the protype for monitoring and safety systems. The packages we use can be grouped into software packages for face recognition and computer vision, for creating the application which shows live graphical representation of the sensor data and then software packages for operating the raspberry Pi.

The software utilized in this project are as follows:

- Raspbian
- PuTTY
- Open CV
- MIT App Inventor
- Dlib
- Face recognition

4.1 RASPBIAN DESKTOP

It is a Debian-based (32 bit) computer operating system for Raspberry-Pi. There are several versions of Raspbian including Raspbian Buster and Raspbian Stretch. Raspbian is highly optimized for the Raspberry Pi line's low-performance ARM CPUs. Raspbian uses PIXEL, Pi Improved X-Window Environment, Lightweight as its main desktop environment as of the latest update.

It is composed of a modified LXDE desktop environment and the Open box stacking window manager with a new theme and few other changes. The distribution is shipped with a copy of computer algebra program Mathematica and a version of Minecraft called Minecraft Pi as well as a lightweight version of Chromium as of the latest version**.** Fig. 4.1 shows how a fully loaded Raspbian desktop looks.

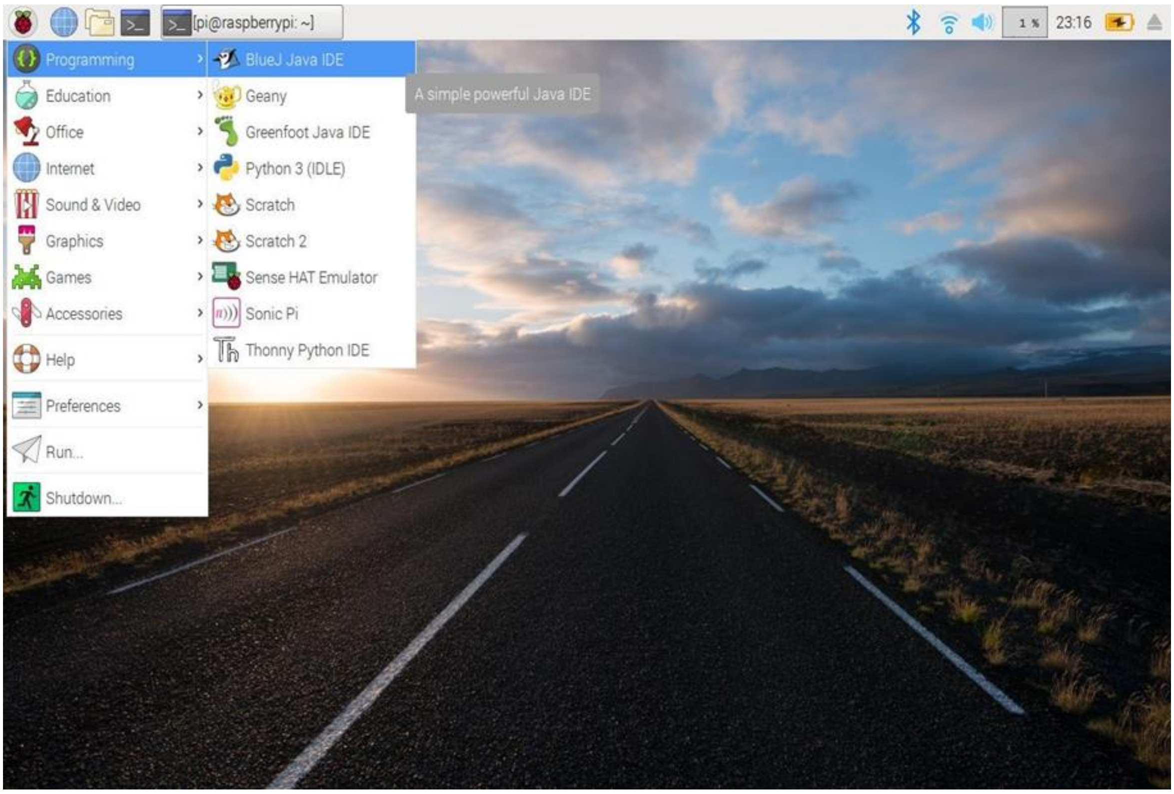

Fig. 4.1 Rasbian desktop

4.2 PUTTY

PuTTY is a serial console and file transfer application that is also an open-source terminal emulator. It supports serial port connection and also other network protocols which includes SCP, SSH, Telnet, Login, and raw socket connection.

PuTTY was ported to various other operating systems, even though it was originally written for Microsoft Windows. A few platforms like UNIX are available on official ports, and also include Classic mac OS and mac OS, and unofficial ports that are contributed to platforms like Symbian Windows Mobile and Windows Phone.

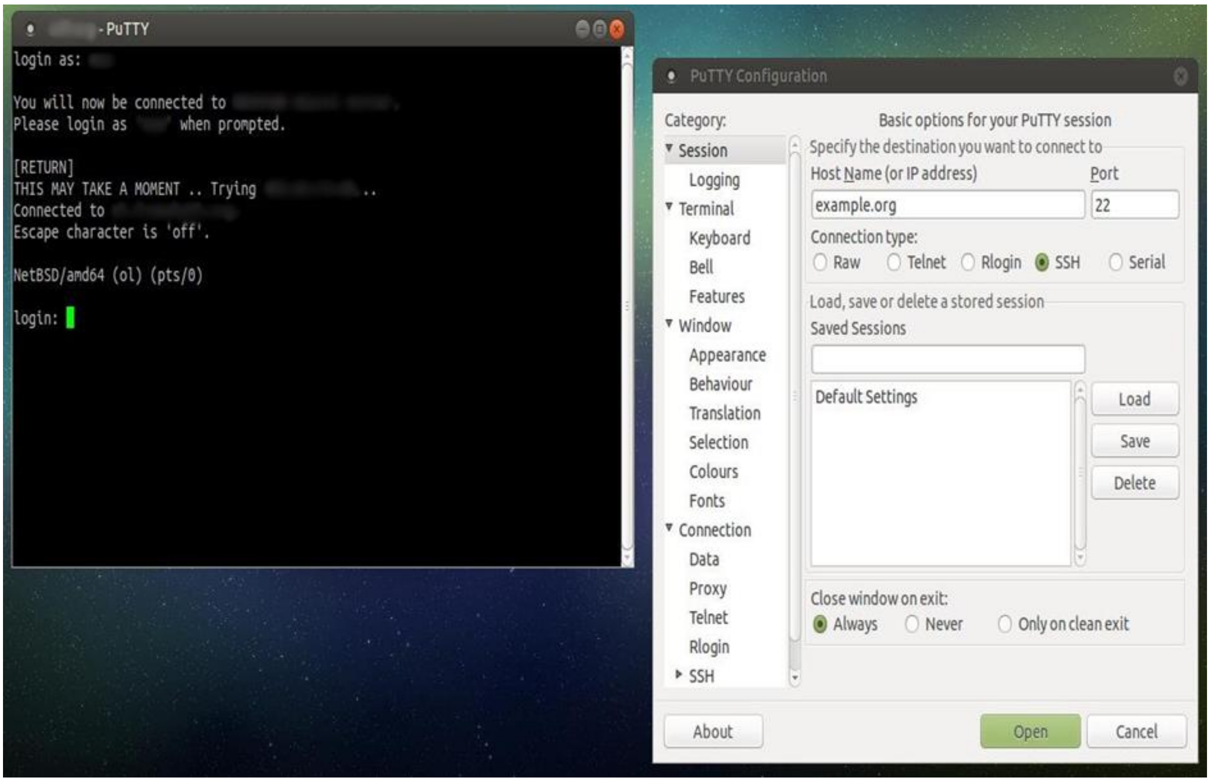

Fig. 4.2 PuTTY running under Ubuntu server

It provides user control over the SSH encryption and supports variations in the secure remote terminal as shown in Fig. 4.2, which also included alternate cyphers like RC4, DES, 3DES, Blowfish and Public-key authentication as user control over SSH encryption key and protocol version. PuTTY can also allow local, remote or dynamic port forwarding with SSH and emulate control sequences from xterm, VT220, VT102 or ECMA-48 emulation. It includes user provided GSSAPI DLLs and supports SSO through GSSAPI. The network communication layer which supports IPv6, and the SSH protocol supports the zlib@openssh.com delayed compression scheme and can be used with local serial port connections.

4.3 OPENCV

OpenCV (Open Source Computer Vision Library) is part of the open source computer vision and machine learning software library that is freely available to any person on the internet including the source code. It provides a common framework for machine learning perception and computer vision applications in the commercial products. OpenCV is licenced by BSD (Berkeley software distribution), so it easy to change to code and utilize it for business purposes.

There are around 2500 computer vision and machine learning algorithms that are state of the art and classical are present in the library. The algorithms can be used for a lot of different purposes such as remove red eyes from images taken using flash, extract 3D models of objects, stitch images together to produce a high resolution image of an entire scene, follow eye movements, track moving objects , classify human actions in videos, recognize scenery and also detect faces and combine it with augmented reality. More and more recent algorithms are being added to the library daily.

There are API's for various languages such as java, python, MATLAB and C++ it can also be used in different operating systems android, windows , macOS and Linux it also takes advantage of MMX and SSE instructions when available.

CUDA and OpenCL which are interfaces for parallel programming which are also being actively developed for OpenCV. There are more functionalities than the algorithms that are present in in the library. OpenCV is mainly maintained and developed in C++ and it is made to work very efficiently with the containers STL.

4.4 MIT APP INVENTOR

Anyone can leverage to solve real-world problems using MIT App Inventor which is an online app development platform. A web-based editor is provided by App Inventor for building mobile phone applications targeted at Android and iOS operating systems. It was inspired by languages like StarLogo TNG (Begel & Klopfer, 2007) and Scratch (Resnick et al., 2009; Maloney, Resnick, Rusk, Silverman, & Eastmond, 2010), and uses a block-based programming language built using Google Blockly (Fraser, 2013) and Scratch, empowering anyone to build a mobile phone app to meet a need. App Inventor has helped over 6.8 million people in over 190 countries to build over 24 million apps thus far. The interface supports many languages from all around the globe. App Inventor has been used by people around the world to provide efficient solutions to real problems in their environment, community and the world. It has been adapted to serve the requirements of specific populations, like building apps for emergency/first responders (Jain et al., 2015) and robotics (Papadakis & Orfanakis, 2016).

The MIT App Inventor consists of two main editors in its user interface which are the design editor and the blocks editor. The design editor, or designer, which is a drag and drop interface which helps in laying out the elements of the application's user interface (UI). App inventors can visually lay out the logic of their apps using color-coded blocks using the blocks editor's environment which snap together like puzzle pieces to describe the program. It also provides a mobile app called the App Inventor Companion (or just "the Companion") which developers use to test and help in the development of the app by adjusting the behaviour of their apps in real time as shown in Fig. 4.3. This feature helps in quickly building a mobile app and also helps in the testing phase of the application.

In MIT App Inventor, a block-based programming language is used by the app developers to code the applications behaviour. The blocks used in App Inventor are of two types which are built-in blocks and component blocks. The basic atoms and operations generally available in other programming languages, like Booleans, strings, numbers, lists, mathematical operators, comparison operators, and control flow operators are provided by the built-in blocks' library as shown in Fig. 4.4. Component blocks are used to respond to system and user events, interact with device hardware, and adjust the visual and behavioural aspects of components and consists of the properties, methods and events used while building the app. App Inventor limits the runtime creation of new entities unlike

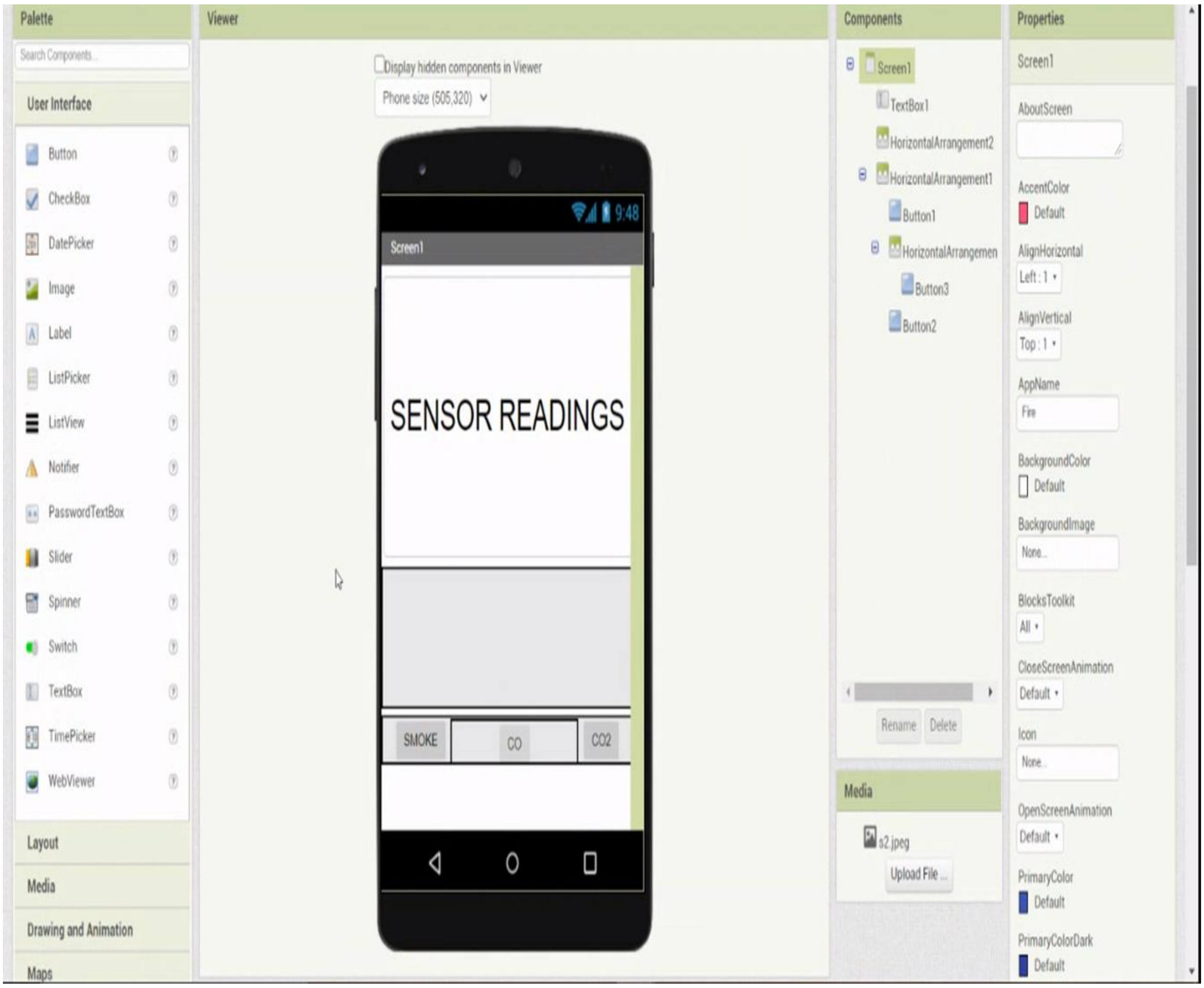

Fig. 4.3 Designer editor

many other programming languages which provides multiple benefits. First, the user can visualize the app more clearly by explicitly positioning app the components in the app rather than having to reason about things that will not exist until a future time. Second, reduces the chances of app to run out of memory as the chance of users introducing cyclic memory dependencies in the user interface is reduced which encourages app inventors to appropriately structure their applications and reuse components to prevent overloading of the system or their end users.

The app developers can visualize how the app appears on the device screen and adjust the form factor of the visualized device (e.g., phone or tablet) according to their requirements using the design editor for App Inventor.

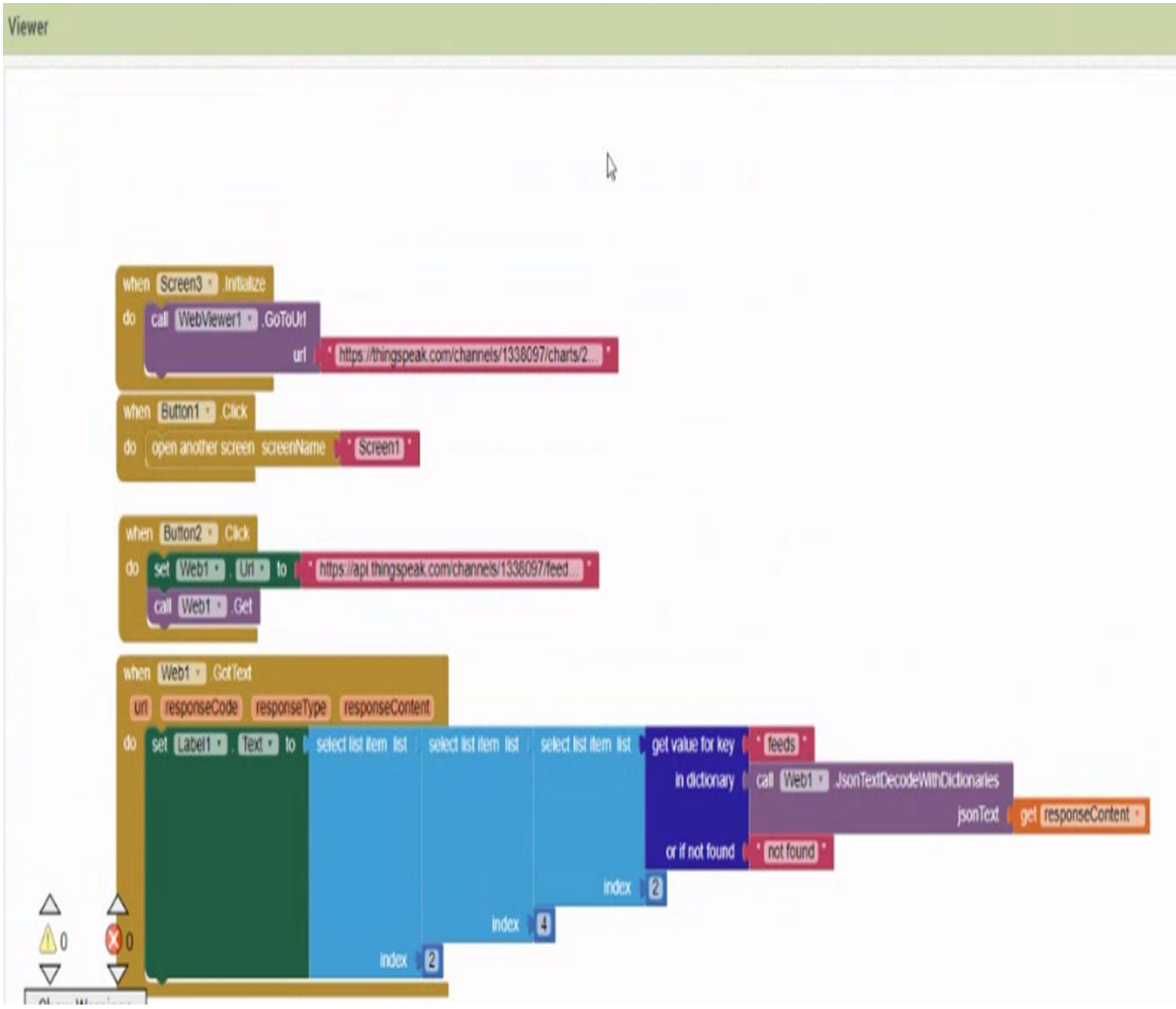

Fig.4.4 Blocks Editor

The adjustments made to the properties of the visual components, for example, background colour and size, are reflected in real time. Using the App Inventor Companion, apps can also be run in a live development mode.

The capability of creating map-based applications has been recently into the App Inventor by the development team. With the help of this functionality, app inventors can develop content-rich apps using drag, drop, and edit markers, lines, polygons, rectangles, and circles in their maps, as well as integrate web-based data from geographic information systems (GIS). Thus, the content can be moved around easily by the user to achieve great result without needing to provide most of the logic for this in code.

4.5 DLIB

It is a general-purpose software library written in the programming language C++ which supports cross-platform use. Ideas from design by contract and component-based software engineering have heavily influenced the design of Dlib. Thus, it is considered to be a set of independent software components. It is an open-source software released along with a Boost Software License.

The development of Dlib began in 2002 and since then, a wide variety of tools have been added into it which includes software components for dealing with networking, threads, graphical user interfaces, data structures, linear algebra, image processing, data mining, XML and text parsing, numerical optimization, Bayesian networks, machine learning, and many other tasks.

4.6 FACE_RECOGNITION

Face_recognition is a Python library which comprises of various techniques used for face recognition. It works in a few steps:

- Identify a face in a given image
- Identify specific features in the face

Based on this encoding, we measure the similarities between two face images based on the features which helps us figure out whether the two images belong to the same person or not. First the input image is loaded and then the facial features are marked using the face_locations command.

The next involves detecting each component on the face in order to run the comparison for this face_encodings is employed. It is a face for us. But, in the algorithm that we use, it is only an array of RGB values that matches a pattern that it has learnt based on the facial patterns which where provide based on the data samples we provided to it.

The face recognition algorithm notes certain important measurements on the face such as the colour and size and slant of eyes, the gap between eyebrows, etc and these features put together defines the face encoding, the information obtained out of the image which is used to identify the particular face. The output of the face_encodings in the form of arrays that represent each orthogonal component. The final step is to compare the detected image with the known database. This is done by compare_faces command. The output shows the recognized image and shows the name of the person if known otherwise shows a tag as unknown.

CHAPTER 5

METHODOLOGY OF IMPLEMENTATION

Having discussed the software packages and system components prominently present in our system prototype, we move on to deal with the methodology of implementation. The system block diagram is shown in Fig. 5.1. represents flow of implementation of the system. Also we describe the necessary algorithms that we have used in our system for face detection and recognition.

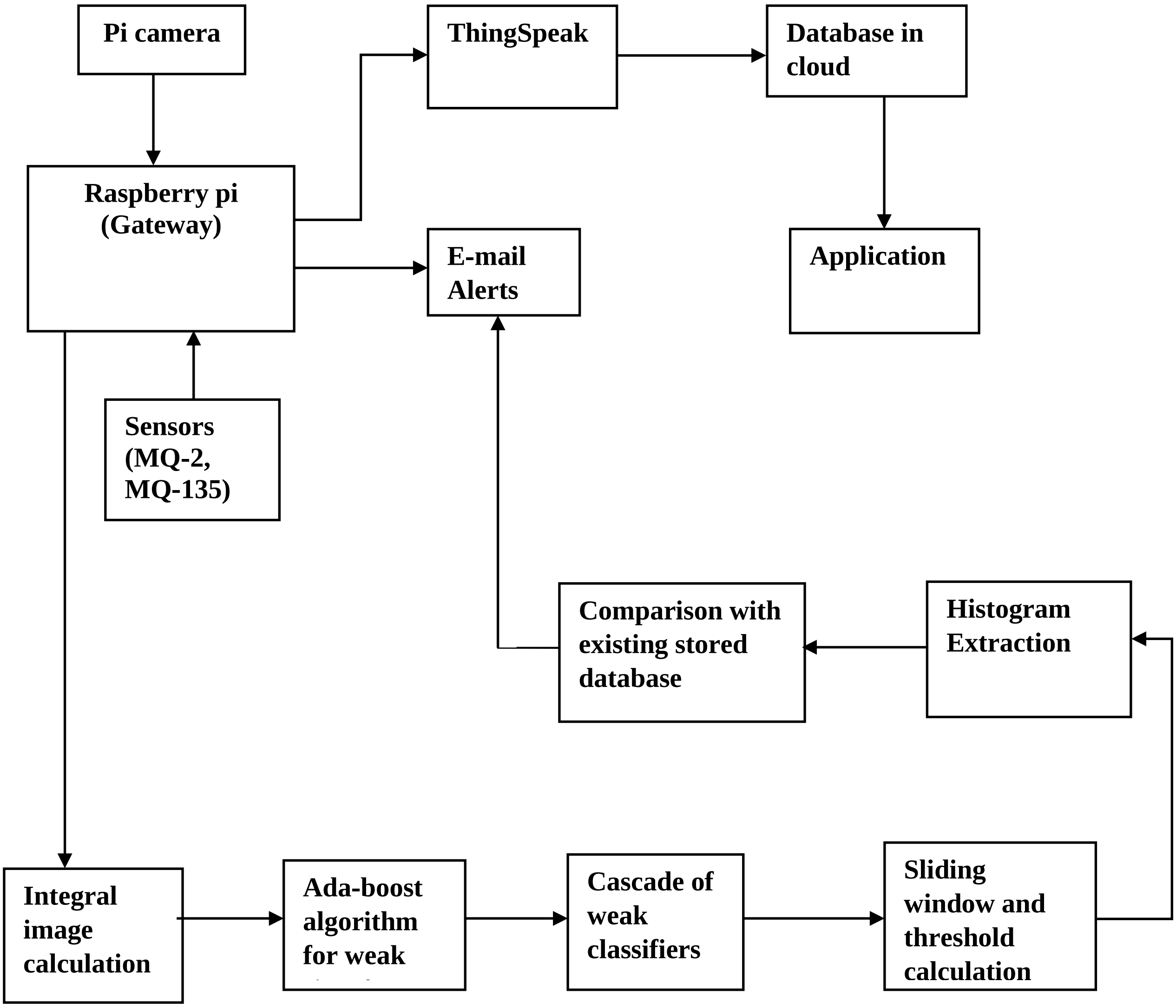

Fig 5.1 Block Diagram of system

5.1 SYSTEM DESCRIPTION

The system description as shown in Fig. 5.1 is described in this section. Using the pi camera we capture the image of the person entering the premises. This captured image is then processed through haar cascade and LBPH algorithm for providing intruder alerts through email and parallelly the environmental data is gathered by the sensors and sent to cloud platform called “ThingSpeak” through raspberry Pi.

5.1.1 RASPBERRY PI (GATEWAY)

The Raspberry Pi is a single board computer that is used as a gateway for processing the raw data received from the sensors and the PiCamera and to send the data to the user through web services.

The Raspberry Pi is fed with a program for processing data received from the sensors and a face detection program both written in Python. The Raspberry Pi is interfaced with multiple sensors such as smoke sensor(mq-2), air quality sensor(mq-2). It is also interfaced with a PiCamera. The Raspberry Pi GPIO pin doesn’t handle voltages above 3.3V so 5V to 3.3V bidirectional 4 channel logic level convertor is used.

5.1.2 SENSORS

The usage of sensors is critical in the system as it provide the raw data which is further utilized for processing. The sensors used for smoke detection air quality monitoring, are analog type sensor and hence an analog to digital converter is used as Raspberry Pi can only interpret a digital signal. ADS1115 , an analog to digital converter by Adafruit is used for the conversion . The analog sensors are connected to the channels of the ADS1115 which based on the input analog signal received produces an equivalent digital output which is used for further processing by the Raspberry Pi.

5.1.3 PI CAMERA

The PiCamera is interfaced to the Raspberry Pi by a short ribbon cable and is used to video record the desired premises. The recorded video is utilized by the Raspberry Pi to perform face detection.

5.1.4 LOCAL SERVER

The purpose of the local server is to perform face recognition on the detected face sent by the Raspberry Pi. The local server is fed with a face recognition code that utilizes dlib, face_recognition, numpy and other required library packages for face recognition; the code is written in Python.

If the detected image is of a known person there is no action taken whereas, if the detected image is of an unknown person an e-mail is sent to the user. By this method the user is intimated instantly only when an unknown person enters the premises.

5.1.5 THINGSPEAK

The cloud storage is used to store the information regarding the sensors' outputs. Thingspeak- an IoT platform service is used to save the data received from the Raspberry Pi. A channel is created in thingspeak which has multiple fields where in each field the data of each sensor is stored as shown in Fig. 5.1. For every 15 seconds the channel is made available to receive data from the Raspberry Pi. The stored data is represented in the form of live graphs with information not only the raw data but also the time and date of updation.

The above figure shows the graph used in ThingSpeak, as shown in the figure the graph displays the information with the time and data(x-axis) and the data received at that instant(y-axis).

Fig 5.1 Graphical representation in ThingSpeak

5.1.6 USER(APP AND EMAIL)

The user is intimated regarding any accident or presence of an unknown person within the premises through an E-mail. As far as the statistical data are concerned, an android application is created using the MIT App Inventor which is used to display the data in the form of the live graphs. These graphs are the ones used to represent data in the ThingSpeak channel. The data from the graphs can be utilized for desired analysis. The air quality indicator is also displayed in the mobile application.

5.2 IMPLEMENTATION DESCRIPTION

The Raspberry Pi receives the output of the interfaced sensors and PiCamera and with the help of the program fed to it performs the required operations. Along with the assistance of a local server and an IoT platform the information is sent to the user via an e-mail and a mobile application.

5.2.1 FACE DETECTION

Object Detection using Haar feature-based cascade classifiers is a constructive object detection method proposed by Paul Viola and Michael Jones in their paper, "Rapid Object Detection using a Boosted Cascade of Simple Features" in

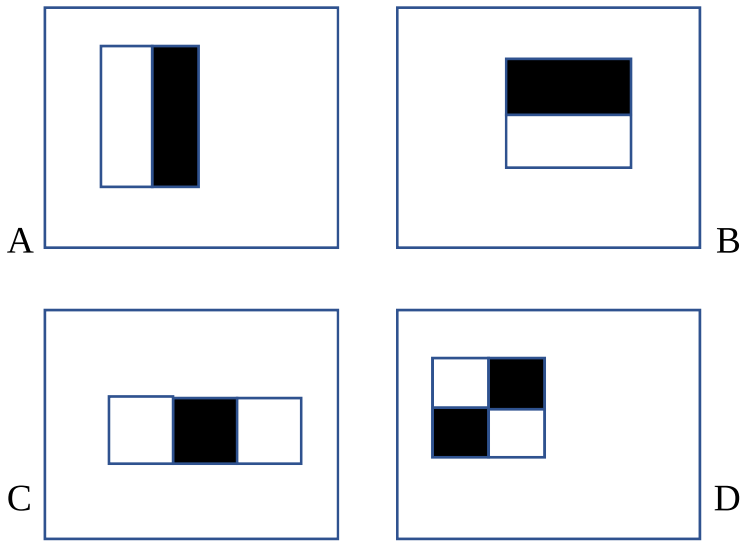

Fig. 5.2 Haar features

2001.Based on positive and negative images the cascade function will be trained. It is then used to identify objects in other images.

For obtaining a feature subtract the sum of pixels of the white rectangle from the pixel sums from black rectangle as shown in Fig. 5.2.

$$ii(x,y) = \sum_{x' \leq x, y' \leq y} i(x', y') \tag{5.1}$$

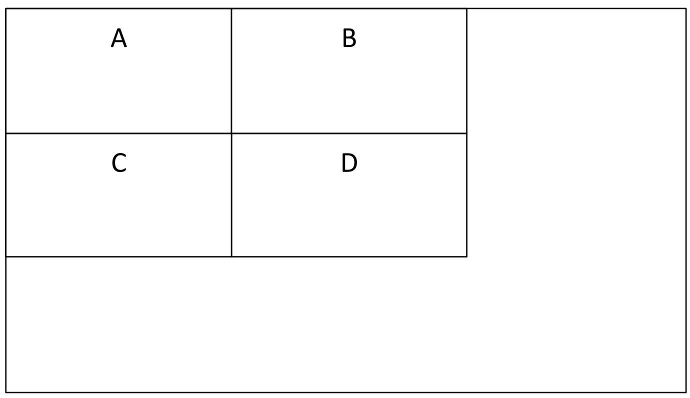

Fig. 5.3 Integral image rectangle

$$s(x,y) = s(x,y-1) + i(x,y) \tag{5.2}$$

$$ii(x, y) = ii(x - 1, y) + s(x, y) \tag{5.3}$$

At first, the algorithm should be given a lot of positive images (images of faces) and negative images (images without faces) to train the classifier. Based on the training features are extracted from it.

A variety of sizes and locations of each kernel are used to evaluate and capture lots of features. Using the integral image, the sum of pixels under black and white rectangle can be calculated for obtaining the features as shown in Fig. 5.3.

The catch here is that, among all these features calculated, most of them are irrelevant. So, among the available features the best features are selected using Adaboost algorithm.

On all the training images apply all the available features then there exists one feature that strongly separates the positive images and negative images, this feature alone is selected. Among the selected features there will be errors and misclassifications, so a feature which has the least error rate which classify whether there is a face or no face is present is selected for use.

By taking weighted sum of the classifiers which were selected from the previous iterations we form a strong classifier. The classifier selected in previous stage alone cannot detect the images but by taking weighted average they can. Most of the picture given does not contain a face, so by using a check method classify if a window consists of facial region or not and discard it in first time processing of the window and do not process it further. The rest of the region consists of facial region this can be focussed on instead of the entire image using classifiers in cascade.

All the features selected are not applied in a single window, they instead are grouped into various stages so if some window does not successfully pass through one stage it is discarded and it is not processed in further layers. But if some window passes in one stage then the next stage of features is applied. If some window passes through all stages then it is considered as a face region**.**

5.2.2 FACE RECOGNITION

Local binary pattern designates a particular pixel of an image by using the binary number that we get from the threshold levels of the neighbouring pixels of the current pixel. This is a very simple strategy but very efficient. Local binary pattern uses four parameters for thresholding the neighbourhood pixels, they are: radius, neighbours, grid x and grid y. Radius is used to represent the area around the chosen pixel and subsequently used to construct the binary pattern. Neighbours is the count of the sample points required to build the binary pattern corresponding to the radius chosen, more the count of the sample points obtained the computational resources needed is also higher. The grid x parameter is used to find the number of cells present in the horizontal direction. The feature vector's dimension depends on this parameter. More the dimension more fine is the grid. The grid y parameter is used to find the number of cells present in the vertical direction. The feature vector's dimension increases as the grid becomes more fine.

The algorithm is trained using the data of the people who will be recognized using their face images. Each person's image is given an id which the algorithm uses to give output corresponding to input image. All images of a particular person are given the same id. Using the sliding window algorithm that takes neighbours and radius parameters into account the algorithm creates an intermediate image that highlights the facial characters and represents the given image in a better way. Convert the image given in grayscale and get a portion of this image as a window of 3 x 3 pixels. This is represented in a matrix that has intensity of each pixels. Using the value that is middle element in the matrix as threshold using which the neighbours are given a binary value depending on whether they are greater than or less than the threshold. This matrix is assigned a new binary value by concatenating each binary value from each position, the value obtained is then converted to a decimal value and set to the central value of the matrix which is a new pixel value from the original image. Thus, the obtained image represents the characteristics of the original image in a better way.

Grid X and grid Y parameters are used to divide the image into multiple grids each represented as histogram containing 256 positions representing the occurrences of each pixel intensity. A Final histogram representing the characteristics of the original image is obtained by concatenating the individual histograms. Now to perform Face recognition a new image is captured and the same steps above is performed. Now the obtained histograms are compared with existing histograms using Euclidean distance therefore the algorithm output is the ID of the most matching histogram.

5.2.4 SENSOR DATA AND IT'S PROCESSING

The source program that is fed into the Raspberry Pi also deals with the operations required to send the data obtained from the sensors to the web server. The sensors used along with their purpose is as follows:

1. MQ-2 - Smoke detection
2. MQ-135 – Air quality detection

MQ-2 is an analog sensor that is used for both smoke detection by setting two different threshold levels in the program based on the calculation. When the levels of the smoke sensor exceeds normal an alert email is sent to the user by the Raspberry Pi via internet. MQ-135 is an analog sensor that is used for both smoke air quality by setting two different threshold levels in the program based on the calculation. When the levels of the smoke sensor exceed normal an alert email is sent to the user by the Raspberry Pi via internet.

The data received from the sensors are converted to ppm, then utilized by the Raspberry Pi for processing and after processing the outputs of the air quality sensors are sent to an IoT platform called ThingSpeak. In ThingSpeak, the received data from the Raspberry Pi is plotted as graphs and this graphical representation is sent to the mobile application. Based on the output of the smoke sensor an alert notification is sent to the user via an e-mail.

5.2.4.1 CONVERTING RAW SENSOR DATA TO PPM

The datasheet of MQ-2 sensor shows the sensitivity characteristics of the sensor and is used to calculate the ppm level of the required particles.

The data sheet is as shown in Fig. 5.4 and Fig. 5.5 shows data in logarithmic form

.

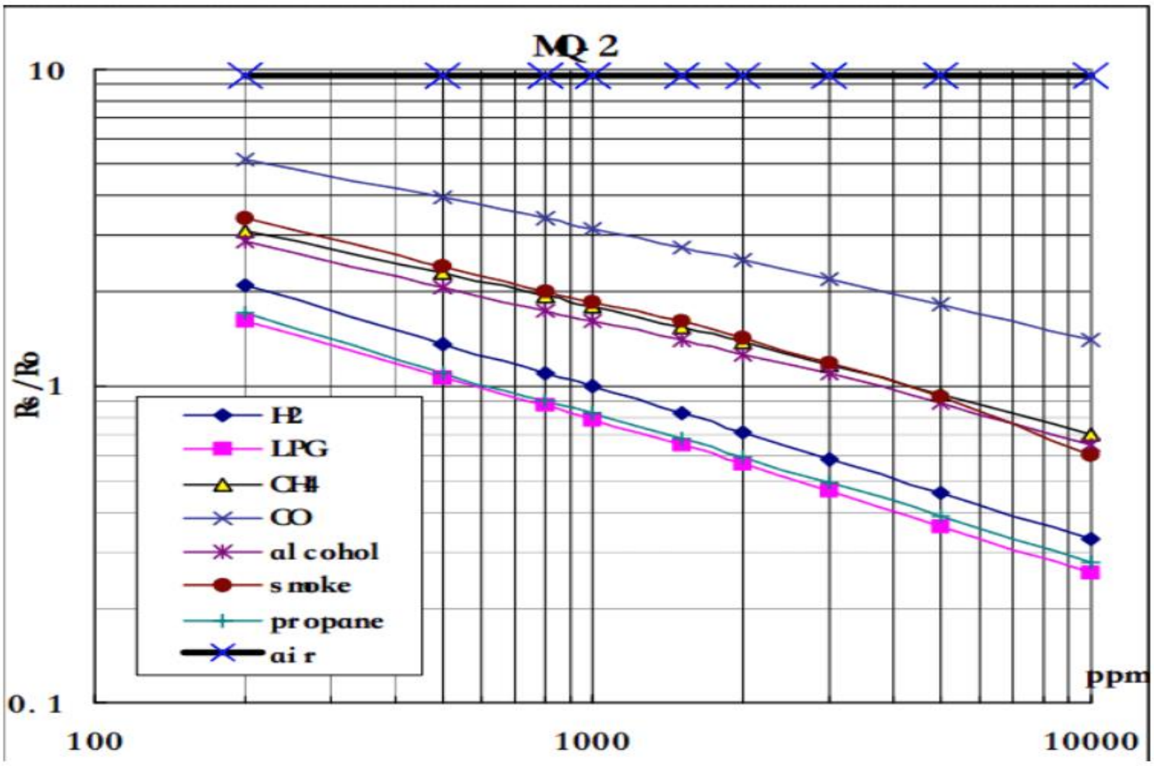

Fig. 5.4 Data sheet MQ-2 sensor

The graph tells us the concentration of a gas in part per million (ppm) according to the resistance ratio of the sensor (R_s/R_o). R_s is the resistance of the sensor that changes depending on the concentration of gas.

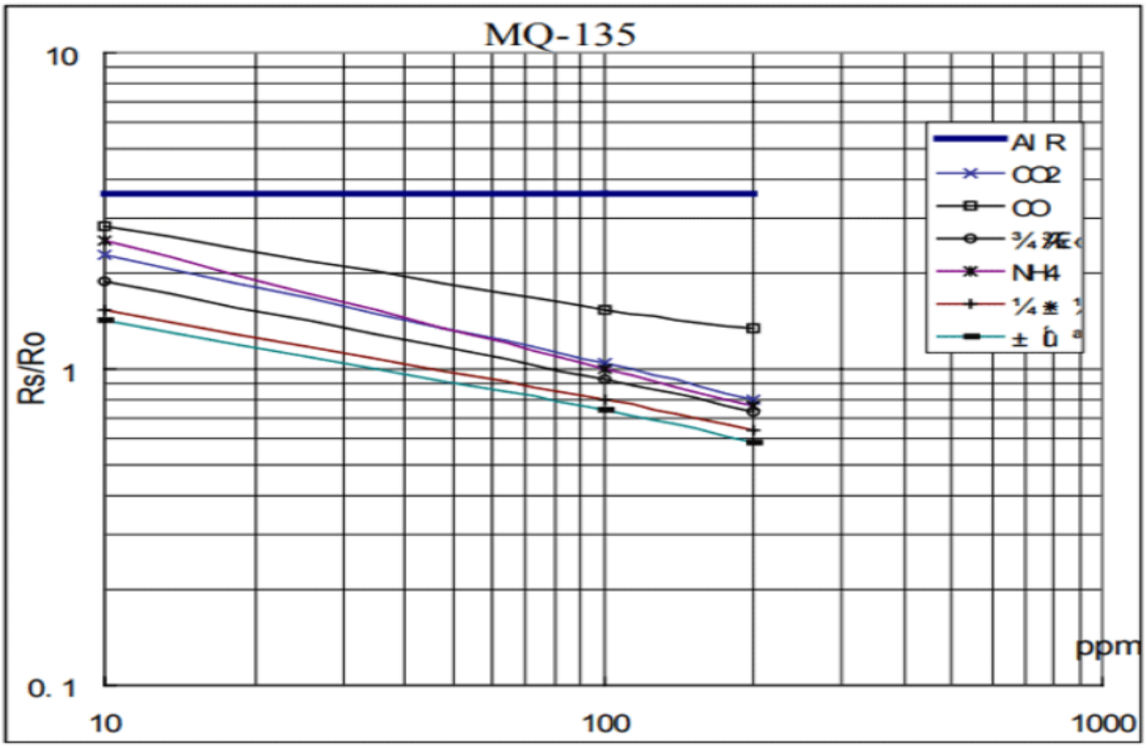

Fig. 5.5 Data Sheet MQ-135 sensor

R_o is the resistance of the sensor at a known concentration without the presence of other gases, or in the fresh air.

For fresh air,

$$\frac{Rs}{Ro} = 9.8\ for\ MQ2\ gas\ sensor \tag{5.4}$$

$$Rs = \frac{Vin - Vout}{Vout} \tag{5.5}$$

$$Ro = \frac{Rs}{9.8} \tag{5.6}$$

$$y = mx + b \tag{5.7}$$

y: X value

x: X value

m: Slope of the line

b: Y intercept

$$\log(y) = m * \log(x) + b \tag{5.8}$$

In our case, we chose the points (x_0,y_0) and (x,y) from the smoke line.
The formula to calculate m is the following:

$$m = \frac{[\log(y) - \log(y_0)]}{[\log(x) - \log(x_0)]} \tag{5.9}$$

Now that we have m, we can calculate the y-intercept. To do so, we need to choose one point from the graph (once again from the smoke line).

$$\log(y) = m * \log(x) + b \tag{5.10}$$

$$b = \log(y) - m * \log\,(x) \tag{5.11}$$

In order to get the real value of the gas concentration according to the log-log plot we need to find the inverse log of x:

$$x = 10^{\left(\frac{[\log(y)-b]}{m}\right)} \text{ in ppm} \tag{5.12}$$

In our implementation the calculated values are tabulated as follows:

Table 5.1 Calculated Values

Name of substance	Rs	Ro	m	b
Smoke	8.708	0.8886	-0.4434	1.617
CO2	23.8	6.43	-0.36664	0.7746
CO	23.8	6.63	-0.21966	0.6233

5.2.5 MOBILE APPLICATION

The mobile application is built using MIT App Inventor and is designed for Android. The application is designed to display the levels of smoke, CO_2 and CO in the form of a graph that shows the data along with the date and time. The application also has an indicator that shows the level of safety in air quality**.**

CHAPTER 6

RESULTS

The output of the sensors and the face recognition module is displayed with the assistance of a mobile application and e-mail. The following depicts the representation of each output. The following graphs represents the amount of particulate matter for their respective type of particles. The before graphs show very less variation in ppm as time increases, because of absence of smoke, CO_2 or CO. As the level of particulate matter in the surrounding environment increases the graph shows increase in ppm value above the threshold for safe levels. This data can be used to provide alarm and notification to the user.

6.1 SENSOR DATA – MQ2

Fig. 6.1 Before smoke

Oxygen gets adsorbed on the surface of sensing material when it is heated in air at high temperature. Then donor electrons present in tin oxide are attracted towards this oxygen, thus preventing the current flow, hence the output analog values are low therefore the ppm values are also normal as shown in Fig. 6.1.

Fig. 6.2 Increasing during smoke

When smoke is present, these oxygen atoms react with the smoke thereby decreasing the surface density of the adsorbed oxygen. Now current can flow through the sensor, which generates high analog voltage values hence ppm values are also high as shown in Fig. 6.2 it also shows that there is a possibility of fire could be present in the premises.

Fig. 6.3 Decreasing after smoke

The ppm values decrease as the smoke reduces over time as the current flow ceases as shown in Fig. 6.3.

6.2 SENSOR DATA – Mq135 – CO_2

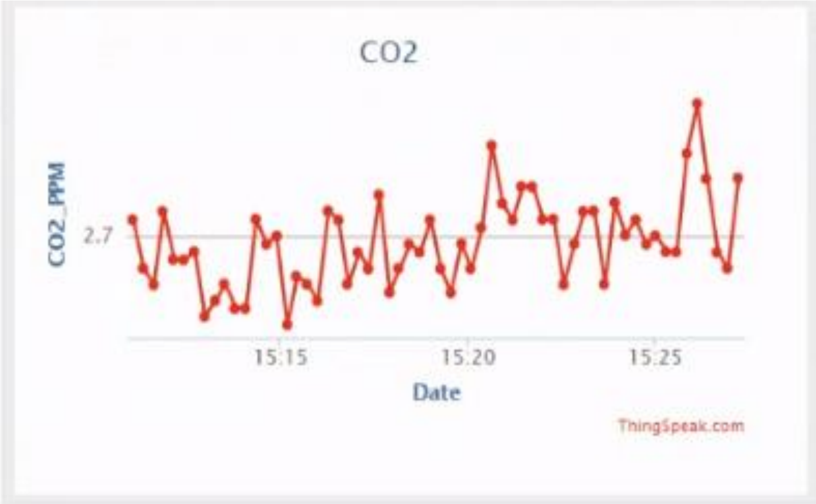

Fig. 6.4 Before excess CO_2

MQ-135 gas sensor applies SnO_2 which has a higher resistance in the clear air as a gas-sensing material. Hence the ppm values are not high and normal as shown in Fig. 6.4.

Fig. 6.5 Increasing during excess CO_2

When there is an increase in carbon dioxide(CO_2), the resistance of the gas sensor decreases along with that as the stannic oxide reacts with gas to decrease the resistance hence ppm values are high as shown in Fig. 6.5.

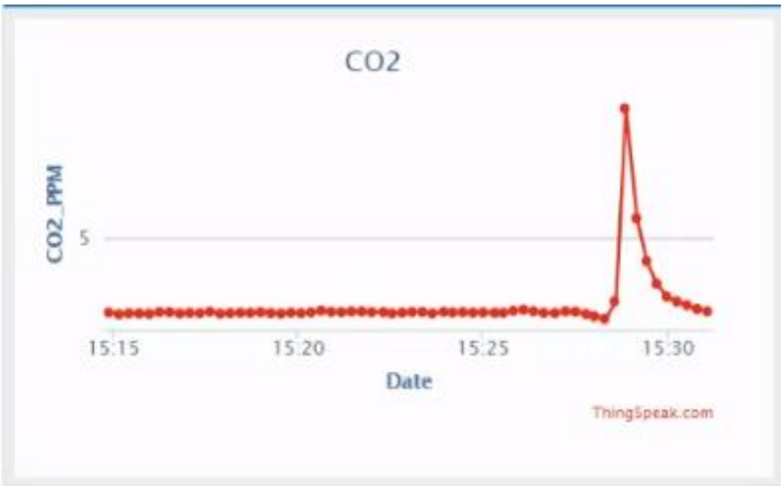

Fig. 6.6 Stabilizing after excess CO_2

As the gases cease to exist the stannic oxide has higher resistance and hence the graphs start stabilizing as shown in Fig. 6.6.

6.3 SENSOR DATA – MQ135 -CO

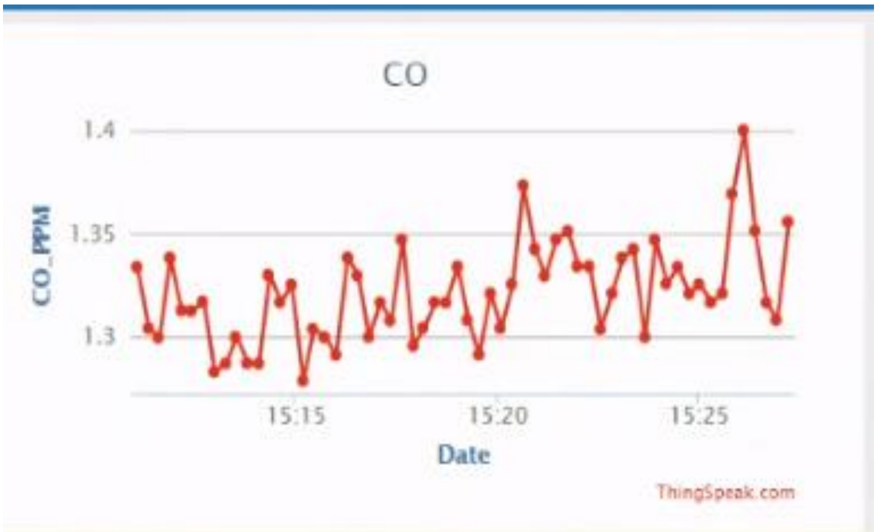

Fig. 6.7 Before increase in CO ppm

MQ-135 gas sensor applies SnO_2 which has a higher resistance in the clear air as a gas-sensing material. Hence the ppm values are not high and normal as shown in Fig. 6.7.

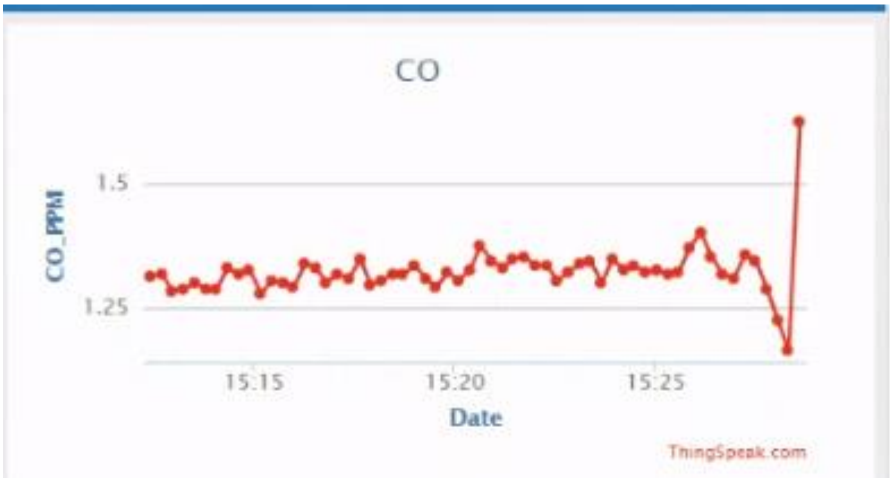

Fig. 6.8 Increasing

When there is an increase in carbon dioxide(CO_2), the resistance of the gas sensor decreases along with that as the stannic oxide reacts with gas to decrease the resistance hence ppm values are high as shown in Fig. 6.8.

Fig. 6.9 Stabilizing after increase in CO ppm

As the gases cease to exist the stannic oxide has higher resistance and hence the graphs start stabilizing as shown in Fig. 6.9.

6.4 FACE DETECTION

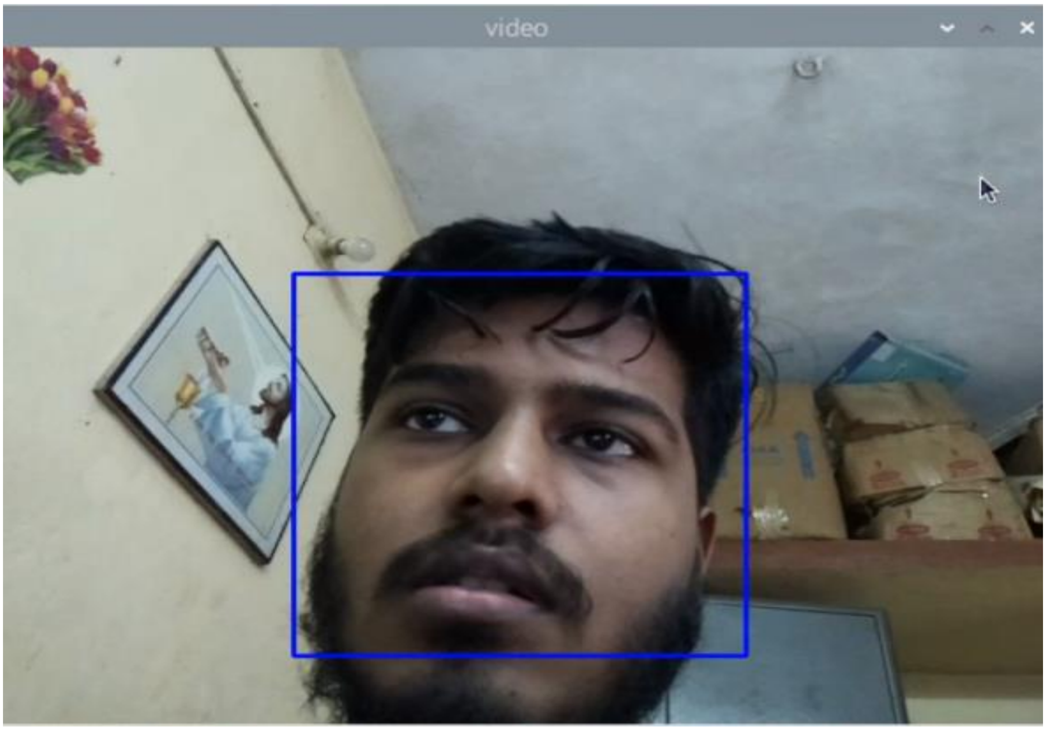

Fig. 6.10 Face Detection

So first stage of haar cascade is integral image which is used to extract the features efficiently from the image, then followed by the execution of adaboost algorithm which is used to select few features from a large set of features which classifies the images as face regions and non facial regions. Then cascade of classifiers is used which further computes on windows which were not rejected in the previous stages of the classifiers. So the face detection happens rapidly. Fig. 6.10 shows the face detected using haar cascade. It can be seen as a blue colour bounding box bounding the face.

6.5 FACE RECOGNITION

The detected faces are recognized as seen by the green bounding box with the names of the respective people recognized using LBPH algorithm.

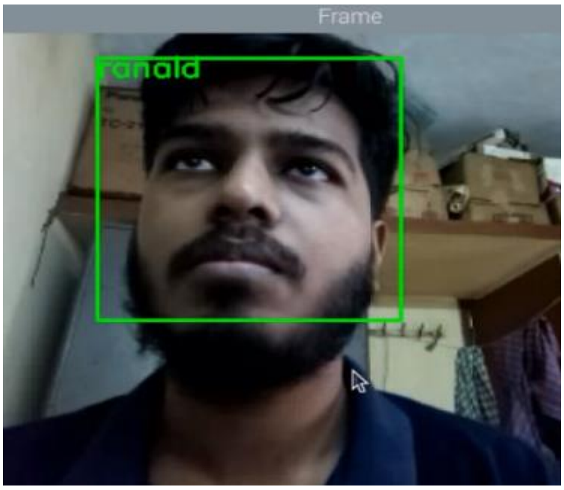

Fig. 6.11 Face Recognition1

Fig. 6.12 Face Recognition2

Fig. 6.13 Face Recognition3

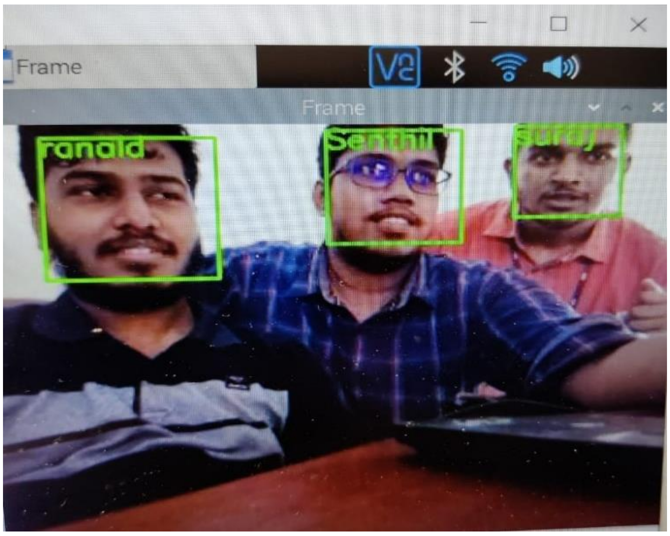

Fig. 6.14 Face Recognition4

This detected face is labelled as unknown as shown in Fig. 6.13 which happens because of no matching with the trained face samples of the known person. Multiple people can also be recognized in a frame as shown in Fig. 6.14 which makes the algorithm applicable for commercial use. The faces to be recognized are stored as histograms of the person with the corresponding ID's of the person.

6.6 ALERT EMAILS

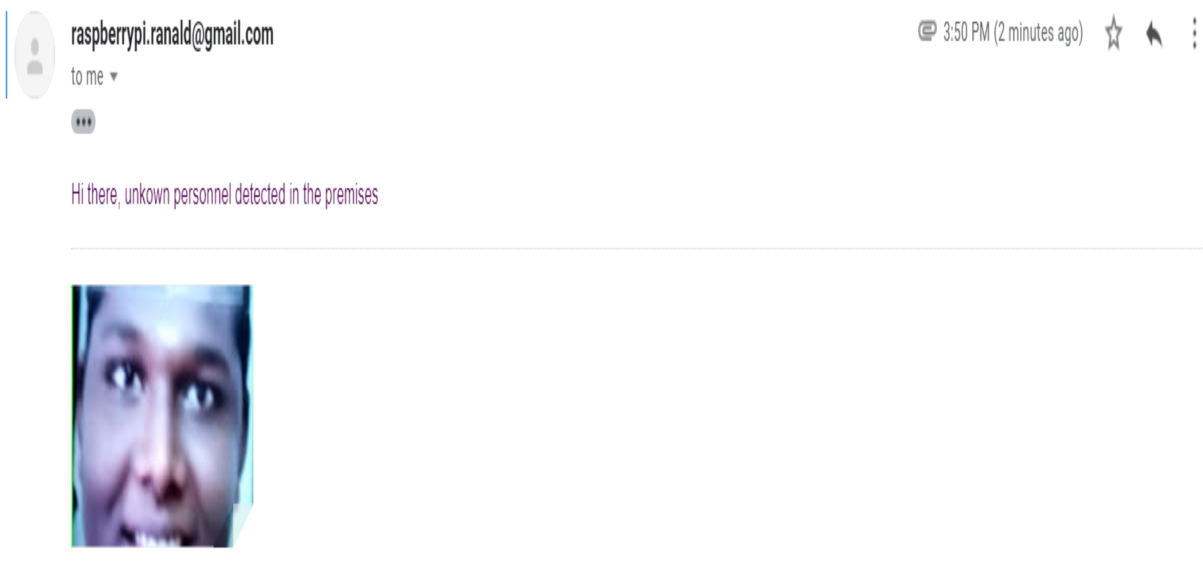

Fig. 6.15 Alert for unknown person

Fig. 6.16 Alert for fire detected

CHAPTER – 7

CONCLUSION AND FUTURE WORK

The proposed system is designed in such a way that it integrates multiple features such as surveillance and accident management and statistical data management into a single system. The compactness and simplicity of the system makes it highly easier to use for the user. The mobile application helps maintain a chronological data for further analysis by the user by representing the amount of pollutants in our case smoke, carbon dioxide and carbon monoxide in the form of live graphs that helps to provide a visual representation of the data with respect to time. As far as accident management is considered, the system sends an alert notification in case of any mishap which helps the user be able to be aware of the status of the premises even when not present. The surveillance system helps to identify any intruder and sends immediate alerts to the user via email. Thus, the proposed system is hugely useful and easily operatable which increases the comfortability of the user.

We recognized that the proposed system could include additional functionalities such as disaster management system. The face recognition algorithm can be made more robust and efficient which could increase the accuracy of intruder detection. Better quality camera other than the pi cam can be used for greater range and accuracy. Further the face recognition database can be stored in cloud so that it is scalable and large quantities of data can be stored which might be a necessity for big corporations. App also can also be designed using latest flutter technology so that a single design can be used for IOS, android and web apps.

REFERENCES

[1] Prajakta Pardeshi Sonali P. Gulve, Suchitra A. Khoje,(2017) ‘Implementation of IoT-based Smart Video Surveillance System’, Computational Intelligence in Data Mining, Vol. 556, pp. 771-780.

[2] Sun, Xiaogang Wang, Xiaoou Tang, Yi,(2013) ‘Hybrid Deep Learning for Face Verification’, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 38, pp. 10- 20.

[3] R. Chellappa, C.L. Wilson and S. Sirohey,(1995) ‘Human and machine recognition of faces: A survey ’, Proceedings of IEEE , Vol 83, No. 5, pp. 705-740.

[4] Michael Jones, Paul Viola, (2001) ‘Rapid Object Detection using a Boosted Cascade of Simple Features’, IEEE computer society conference on computer vision and pattern recognition, Vol. 1, pp. 1-9.

[5] Chaitali Kadam, Ishita Gupta, Shreya Dumbre ,Varsha Patil, (2016) ‘Face detection and Recognition using Raspberry pi’, IEEE International WIE Conference 185 on Electrical and Computer Engineering (WIECON-ECE), Vol. 2, pp 83–86.

[6] Bhuvaneswari Balachander , Pravin Kumar.S, (2019) ‘Fast and Cost Efficient Face Detection System with CNN Using Raspberry PI ‘, International Journal of Engineering and Advanced Technology (IJEAT) , Vol. 8, No. 6, pp. 3037-3039.

[7] Abbas Ali, Moin Moghal ,Zishan Khan, (2014) ‘IoT based Air Pollution using NodeMCU and Thingspeak’, International Journal of Scientific and Engineering Research, Vol. 9, No. 2, pp. 62-66.

[8] Mohan Joshi,(2017) ‘Research Paper on IoT based Air and Sound Pollution monitoring system’, International journal of computer applications Journal, Vol. 178, No.7, pp. 11-17.

[9] K. Sung and T. Poggio.(1998) ‘Example-based learning for viewbased face detection.’ In IEEE Pattern Analysis Mach Intelligence, Vol. 20, pp. 39–51.

[10] C. Papageorgiou, M. Oren, and T. Poggio. (1998) 'A general framework for object detection.' International Conference on Computer Vision, Vol. 63, pp. 555-562.

[11] A. Ahmed, J. Guo, F. Ali, F. Deeba and A. Ahmed, (2018) 'LBPH based improved face recognition at low resolution,' International Conference on Artificial Intelligence and Big Data (ICAIBD), Chengdu, China, pp. 144-147.

[12] Mohammad Mohiddin, V. Sesha Sai Kumar (2018) 'A Low Cost Air Quality Monitoring Sensor System with Arduino Uno Micro Controller Board on Smart Phone and Laptop' Advances in Industrial Engineering and Management, Vol. 7, No 2, pp. 22-29.

[13] Riteeka Nayak, Malaya Ranjan Panigrahy, Vivek Kumar Rai, T Appa Rao,(2017) 'IoT Based Air Pollution Monitoring System' Imperial journey of interdisciplinary research, Vol. 3, No. 4, pp. 571-575.

[14] Chaitanya Baweja, Dr. Simmarpreet, Supriya Chopra, Vandana Kalra (2016), 'Influence of Temperature and Humidity on the Output Resistance Ratio of the MQ-135 Sensor' , International Journal of Advanced Research in Computer Science and Software Engineering, Vol. 6, No. 4, pp. 423-429.

[15] Brawner Brian L. Heyasa , Van Ryan Kristopher R. Galarpe , (2017) 'Initial Development and Testing of Microcontroller-MQ2 Gas Sensorfor University Air Quality Monitoring' , IOSR Journal of Electrical and Electronics Engineering, Vol. 12, No. 3, pp 47-53.

senthil kumar <senthil17148@ece.ssn.edu.in>

Fwd: UG Project Report for approval - reg.

Annadurai C <annaduraic@ssn.edu.in> Sun, Jun 13, 2021 at 10:24 AM
To: senthil kumar <senthil17148@ece.ssn.edu.in>, suraj17170@ece.ssn.edu.in, ranaldn@gmail.com, ranald17129@ece.ssn.edu.in

Forwarded Conversation
Subject: UG Project Report for approval - reg.

From: **Annadurai C** <annaduraic@ssn.edu.in>
Date: Fri, Jun 11, 2021 at 9:58 PM
To: Radha S. <radhas@ssn.edu.in>

Respected Mam,
Herewith I have attached my UG students Project Report for your approval Mam.
I kindly request you to accept the same Mam.

Kindly ignore my previous mail Mam.

--
Dr. C.Annadurai
Associate Professor / ECE
SSN College of Engineering
Old Mahabalipuram Road
Kalavakkam -603 110.
Mobile : 9444270709 and 9094370709

From: **Radha S.** <radhas@ssn.edu.in>
Date: Fri, Jun 11, 2021 at 10:19 PM
To: Annadurai C <annaduraic@ssn.edu.in>

Approved

Regards,
Radha

From: **Annadurai C** <annaduraic@ssn.edu.in>
Date: Sat, Jun 12, 2021 at 7:30 PM
To: Vijayalakshmi P <vijayalakshmip@ssn.edu.in>

Dear Mam,

Herewith I have forwarded the approved mail which was received from HOD for UG students project report.

Thanks & Regards,

--
Dr. C.Annadurai
Associate Professor / ECE
SSN College of Engineering
Old Mahabalipuram Road
Kalavakkam -603 110.
Mobile : 9444270709 and 9094370709

::DISCLAIMER::

The contents of this e-mail and any attachment(s) are confidential and intended for the named recipient(s) only. Views or opinions, if any, presented in this email are solely those of the author and may not necessarily reflect the views or opinions of SSN Institutions (SSN) or its affiliates. Any form of reproduction, dissemination, copying, disclosure, modification, distribution and / or publication of this message without the prior written consent of authorized representative of SSN is strictly prohibited. If you have received this email in error please delete it and notify the sender immediately.

Header of this mail should have a valid DKIM signature for the domain ssn.edu.in

Annadurai_Latest.pdf
1465K