Visvesvaraya Technological University, Belagavi.

PROJECT REPORT

on

"A smart image processing system for Hall Management including social distancing – "SoDisCop" "

Project Report submitted in partial fulfillment of the requirement for the award of the degree of

Bachelor of Engineering

in

Electronics and Communication Engineering

For the academic year 2019-20

Submitted by

USN	Name
1CR16EC021	ARCHANA S
1CR16EC139	S RAJARAJESHWARI
1CR16EC158	SHIVANI
1CR16EC170	SREELAKSHMI R B

Under the guidance of

Dr .K.Venkateswaran

Associate Professor

Department of ECE

CMRIT, Bengaluru

Department of Electronics and Communication Engineering

CMR Institute of Technology, Bengaluru – 560 037

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

CMR INSTITUTE OF TECHNOLOGY

CERTIFICATE

This is to Certify that the dissertation work **"A smart image processing system for Hall Management including social distancing – "SoDisCop""** carried out by Student Archana S, S Rajarajeshwari, Shivani, Sreelakshmi R B, USN: 1CR16EC021,1CR16EC139,1CR16EC158,1CR16EC170, bonafide students of **CMRIT** in partial fulfillment for the award of **Bachelor of Engineering** in **Electronics and Communication Engineering** of the **Visvesvaraya Technological University, Belagavi,** during the academic year **2019-20**. It is certified that all corrections/suggestions indicated for internal assessment have been incorporated in the report deposited in the departmental library. The project report has been approved as it satisfies the academic requirements in respect of Project work prescribed for the said degree.

Signature of Guide	Signature of HOD	Signature of Principal
________________	________________	________________
Dr .K.Venkateswaran Associate Professor Dept. of ECE., CMRIT, Bengaluru.	Dr. R. Elumalai Head of the Department, Dept. of ECE., CMRIT, Bengaluru.	Dr. Sanjay Jain Principal, CMRIT, Bengaluru.

External Viva

Name of Examiners Signature & date

1.

2

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of people who made it possible, whose consistent guidance and encouragement crowned our efforts with success.

We consider it as our privilege to express the gratitude to all those who guided in the completion of the project.

We express my gratitude to Principal, **Dr. Sanjay Jain,** for having provided me the golden opportunity to undertake this project work in their esteemed organization.

We sincerely thank Dr. R. Elumalai**,** HOD**,** Department of Electronics and Communication Engineering, CMR Institute of Technology for the immense support given to me.

We express my gratitude to our project guide Dr .K.Venkateswaran**,** Associate Professor, for their support, guidance and suggestions throughout the project work.

Last but not the least, heartful thanks to our parents and friends for their support.

Above all, We thank the Lord Almighty for His grace on us to succeed in this endeavor.

Table of Contents

List of Figures

List of Tables

Chapter 1

INTRODUCTION

1.1.OVERVIEW

The main purpose of this project is to design and employ power saving in general purpose places like Auditorium, Halls, Shopping malls, Theatres etc. The system used in this project also determines the head count of the people in the hall/room. This project describes the smart working of electrical and electronic devices like lights and fans with automatic control with the help controller .power saving is achieved by controlling fan speed and lights on and off depending on the seating arrangement in the hall also it will give alert message to authority person with the help of GSM Module when the number of people exceeds the occupancy limit of the hall. This project monitors the social distancing termed "SoDisCop" and is intentionally kept devoid of complexity target low cost implementation. The novel feature of this work also meets a critical need during the current Covid -19 pandemic.

1.2. MOTIVATION

Wastage of resources: The fans and lights are switched on when it is not required i.e when people are not occupying the hall , the electrical appliances like fans and lights are switched on. This leads to inefficient use of resources and eventually the wastage of these resources. it was found that the most electronic appliances are controlled using digital logic design, but in our project we are making use of sensors to control the electronic appliances depending upon the number of people in the hall.

Crowd Management: Crowd control is always a critical issues especially in large auditorium. In emergency situations it can result in stampede. Proactive monitoring of crowd(audience) is needed.

Social Distancing monitoring (SodisCop feature): the current Covid-19 pandemic of 1m as a critical safety guideline throughout the world by World Health Organization . social distancing has been subject of many on-going recent studies on people movement tracking and the impact of social distancing has been established in detailed.

1.3. OBJECTIVE

The objectives of this project work are as follows

- To design a system which can determines the head count of the people in the hall using digital signal processing.
- The head count should be continuously monitored to ensure that maximum limit. of the capacity of the hall is not exceeded.
- This project should provide smart control of electrical and electronic devices like lights and fans based on sector wise headcount.
- When the total head count of the hall/auditorium exceeds the capacity of the hall. the system should generate alert messages to relevant authority.
- The system should also continuously monitor any social distancing violation in the hall and will notify regarding the same to respective authorities if significant social distancing violations are observed.
- The system should be devoid of complexity and it's cost should be kept low.
- The software should be portable and light weight with respect to computation.
- The system should be adoptable for incorporating in existing video surveillance system as an added feature.

Chapter 2

LITERATURE SURVEY

2.1. Smart Hall Management and crowd control

Various researches have been performed in this field as a part of smart hall management and crowd control. The earlier works exhibit strong dependency on the sensors and advanced camera technology and hence unlike this project's low cost but effective methodology will fall under higher end of the cost spectrum [2],[3],[10]. The common method of head count is to use sensors at the entrances and detect the inflow and out of the people from the hall or vehicle [12]. The commonly used sensors for head count are

- Laser beam
- Infra-red sensor
- Thermal sensor
- video camera.

The most accurate headcount can be obtained through mechanical turnstiles but will result in constrained movement of people and hence not a suitable option [2]. Apart from complexity of consolidating the data through multiple entrances which is the norm in medium to large sized halls, the sensors-based system also requires network connectivity for data concentration and consolidation at the central system controller [7],[8]. Continuous video data through the network is a probable cause for loading the network. Occupancy or headcount-based energy management pays rich dividends through energy saved in HVAC systems [11]. Motion detection sensors play a key role in detecting the occupancy of the hall and controlling the switching of energy elements. Often these systems result in erroneous operation when the occupants are still without any movement and are mostly suited for binary operations. Headcount based energy management system overcomes these deficiencies offering linear control of the utilities.

2.2. Social distancing monitoring

Social distancing has been subject of many on-going studies involving people tracking and controlling the spread of Covid-19 [9]. These studies have been based on deep learning and convolution neural network (CNN) and utilizes advanced object detection model and statistical extrapolation to detect social distance violations

Chapter 3

HARDWARE

3.1.Raspberry Pi

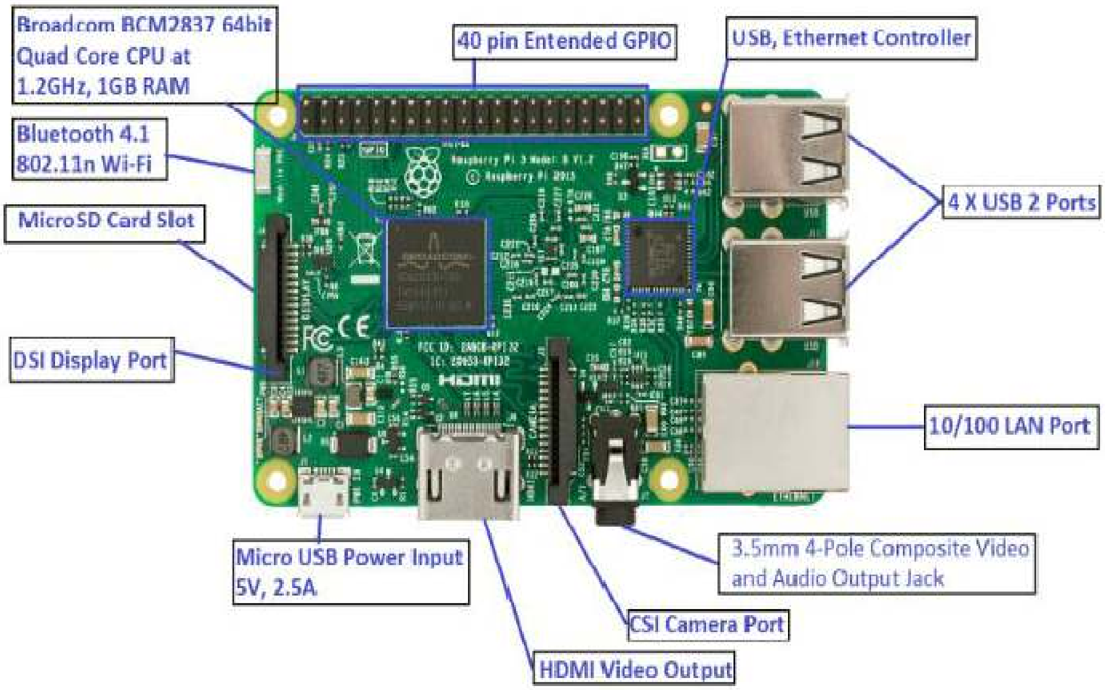


Fig 1: Raspberry Pi

The Raspberry Pi 3 Model B+ is the latest product in the Raspberry Pi 3 range, boasting a 64-bit quad core processor running at 1.4GHz, dual-band 2.4GHz and 5GHz wireless LAN, Bluetooth 4.2/BLE, faster Ethernet, and PoE capability via a separate PoE HAT The dual-band wireless LAN comes with modular compliance certification, allowing the board to be designed into end products with significantly reduced wireless LAN compliance testing, improving both cost and time to market. The Raspberry Pi 3 Model B+ maintains the same mechanical footprint as both the Raspberry Pi 2 Model B and the Raspberry Pi 3 Model B.

Table1 Technical Specifications

Processor: Broadcom BCM2837B0, quad-core A53 (ARMv8) 64-bit SoC @1.4GHz
Memory: 1GB LPDDR2 SDRAM
Connectivity: 2.4GHz and 5GHz IEEE 802.11 b/g/n/ac wireless LAN, Bluetooth 4.2, BLE. Gigabit Ethernet over USB 2.0 (maximum throughput of 300Mbps).

USB: 4 x 2.0
Expandability: Extended 40-pin GPIO header
Video and sound: 1 x full-sized HDMI port, MIPI DSI display port, MIPI CSI camera port, 4 pole stereo output and composite video port.
Multimedia: H.264, MPEG-4 decode (1080p30), H.264 encode (1080p30); OpenGL ES 1.1, 2.0 graphics
SD card support: microSD format for OS and data storage
Input power: 5V/2.5A DC via microUSB connector, 5V DC via GPIO header, Power over Ethernet (PoE)-enabled (requires separate PoE add-on).
Environment: Operating temperature 0 - 50C

3.1.1. Processor

This is a Broadcom chip used in Raspberry Pi 3A+ and 3B+ models. The ARM cores are cpable of running upto 1.4GHz, making the 3B+ about 17% faster than the original Raspberry Pi 3. The VideoCore IV runs at 400 MHz. The ARM core is 64-bit, while the VideoCore IV is 32-bit.

3.1.2. Power

The Raspberry Pi 3 is powered by a +5.1V micro USB supply. Exactly how much current (mA) the Raspberry Pi requires is dependent on what you connect to it. We have found that purchasing a 2.5A power supply from a reputable retailer will provide you with ample power to run your Raspberry Pi. If you need to connect a USB device that will take the power requirements above 1 Amp, then you must connect it to an externally-powered USB hub. The power requirements of the Raspberry Pi increase as you make use of the various interfaces on the Raspberry Pi. The GPIO pins can draw 50mA safely, distributed across all the pins; an individual GPIO pin can only safely draw 16mA. The HDMI port uses 50mA, the camera module requires 250mA, and keyboards and mice can take as little as 100mA or over 1000mA.

3.1.3. Power Pins

The Raspberry Pi can provide both 5v (pins 2 and 4) and 3.3v (pins 1 and 17) power. It also provides a ground (GND) for circuits on pins 6, 9, 14, 20, 25, 30, 34, and 39. How much current the 5v power pins can draw is reliant on what power supply you are using, and what other components you have attached to your Pi. The Raspberry Pi 3 will only draw 2.5A from its power supply, and requires around 350mA for boot up and normal headless operation. The 3.3v pins in recent Raspberry Pi 3B+ provides up to 500mA. This current is shared throughout all of the other GPIO pins too.

3.1.4. GPIO pins

These pins are capable of a 3.3v output, also referred to as setting the pin HIGH in code. When an output pin is LOW this means that it is simply providing 0v. They are also capable of taking an input of up to 3.3v, which the pin reads as HIGH.

3.1.5. Communication

- Pins 8 and 10 (GPIO 14 and 15) are UART pins, designed for communicating with the Pi using the serial port. Two I2C pins—pin 3 (GPIO 2) and pin 5 (GPIO 3). Pins 19, 21, 23, 24, 25 and 26 (GPIO 10, 9, 11, 8, GND, and GPIO 26) are used to connect to an SPI device, and they are all required for smooth operation. The SPI protocol is not enabled as standard on Raspbian, but it can be enabled in the raspi-config file, along with I2C.

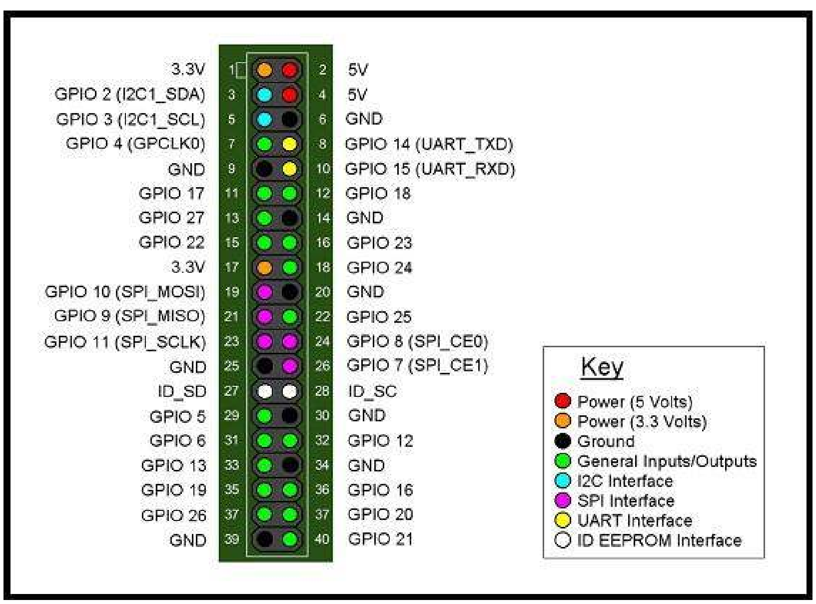


Fig 2. Raspberry Pi Pin Diagram

3.2. Pi Camera

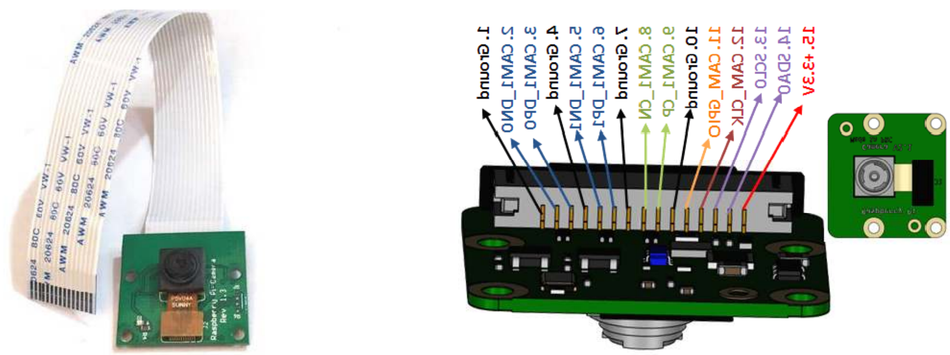


Fig 3: Pi Camera

The Raspberry Pi Camera Board plugs directly into the CSI connector on the Raspberry Pi. It's able to deliver a crystal clear 5MP resolution image, or 1080p HD video recording at 30fps Latest Version 1.3. Custom designed and manufactured by the Raspberry Pi Foundation in the UK, the Raspberry Pi Camera Board features a 5MP (2592x1944 pixels) Omnivision 5647 sensor in a fixed focus module. The module attaches to Raspberry Pi, by way of a 15 Pin Ribbon Cable, to the dedicated 15-pin MIPI Camera Serial Interface (CSI), which was designed especially for interfacing to cameras. The CSI bus is capable of extremely high data rates, and it exclusively carries pixel data to the processor. In terms of still images, the camera is capable of 2592 x 1944 pixel static images, and also supports 1080p at 30fps, 720p at 60fps and 640x480p 60/90 video recording.

The Raspberry Pi Camera Board Features:

- Fully Compatible with Both the Model A and Model B Raspberry Pi
- 5MP Omnivision 5647 Camera Module
- Still Picture Resolution: 2592 x 1944
- Video: Supports 1080p at 30fps, 720p at 60fps and 640x480p 60/90 Recording
- 15-pin MIPI Camera Serial Interface - Plugs Directly into the Raspberry Pi Board
- Size: 20 x 25 x 9mm

- Weight 3g
- Fully Compatible with many Raspberry Pi cases

Table 2 -Pi Camera Pin Description

Pin Number	**Pin Name**	**Description**
1	Ground	System Ground
2,3	CAM1_DN0, CAM1_DP0	MIPI Data Positive and MIPI Data Negative for data lane 0
4	Ground	System Ground
5,6	CAM1_DN1, CAM1_DP1	MIPI Data Positive and MIPI Data Negative for data lane 1
7	Ground	System Ground
8,9	CAM1_CN, CAM1_CP	These pins provide the clock pulses for MIPI data lanes
10	Ground	System Ground
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

3.3.GSM MODULE

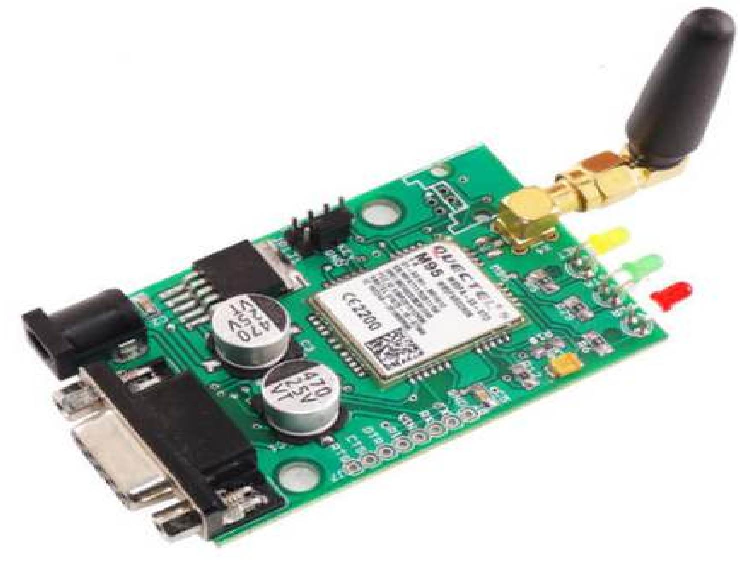


Fig 4: GSM Modem

A GSM modem is a device which can be either a mobile phone or a modem device which can be used to make a computer or any other processor communicate over a network. A GSM modem requires a SIM card to be operated and operates over a network range subscribed by the network operator. It can be connected to a computer through serial, USB or Bluetooth connection.

A GSM modem can also be a standard GSM mobile phone with the appropriate cable and software driver to connect to a serial port or USB port on your computer. GSM modem is usually preferable to a GSM mobile phone. The GSM modem has wide range of applications in transaction terminals, supply chain management, security applications, weather stations and GPRS mode remote data logging.

Features of GSM Module:

- Improved spectrum efficiency
- International roaming
- Compatibility with integrated services digital network (ISDN)
- Support for new services.
- SIM phonebook management
- Fixed dialing number (FDN)
- Real time clock with alarm management

- High-quality speech
- Uses encryption to make phone calls more secure
- Short message service (SMS)

3.4.LED

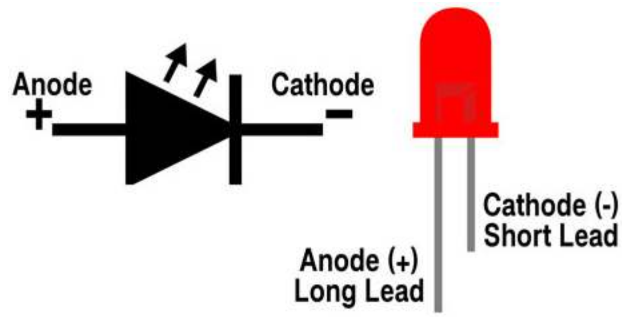


Fig 5: Ordinary LED

A light-emitting diode (LED) is a semiconductor light source that emits light when current flows through it. Electrons in the semiconductor recombine with electron holes, releasing energy in the form of photons. The color of the light (corresponding to the energy of the photons) is determined by the energy required for electrons to cross the band gap of the semiconductor.[5] White light is obtained by using multiple semiconductors or a layer of light-emitting phosphor on the semiconductor device.

An LED has a positive (Anode) lead and a negative (Cathode) lead. The schematic symbol of the LED is similar to the diode except for two arrows pointing outwards.

The Anode (+) is marked with a triangle, and the Cathode (-) is marked with a line. The longer lead of an LED is generally the positive (Anode), while the shorter lead is the negative (cathode).

3.5.SERVO MOTOR

Fig 6: Servo Motor

Tiny and lightweight with high output power. Servo can rotate approximately 180 degrees (90 in each direction), and works just like the standard kinds but smaller. You can use any servo code, hardware or library to control these servos. Good for beginners who want to make stuff move without building a motor controller with feedback & gear box, especially since it will fit in small places. It comes with a 3 horns (arms) and hardware.

Table 3 – Servo Motor Wire Configuration

Wire Number	Wire Colour	Description
1	Brown	Ground wire connected to the ground of system
2	Red	Powers the motor typically +5V is used
3	Orange	PWM signal is given in through this wire to drive the motor

3.6. Connection Diagrams

The hardware set up of this project is shown in fig below. The detailed connection diagrams are shown in the following figure. The serial interface between Pi3 and GSM with level shifting circuit is shown separately in fig

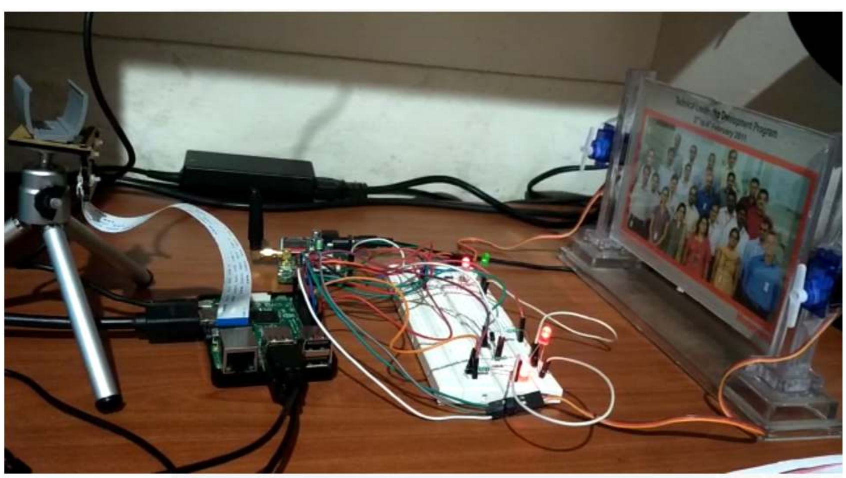


Fig 6A Hardware set up

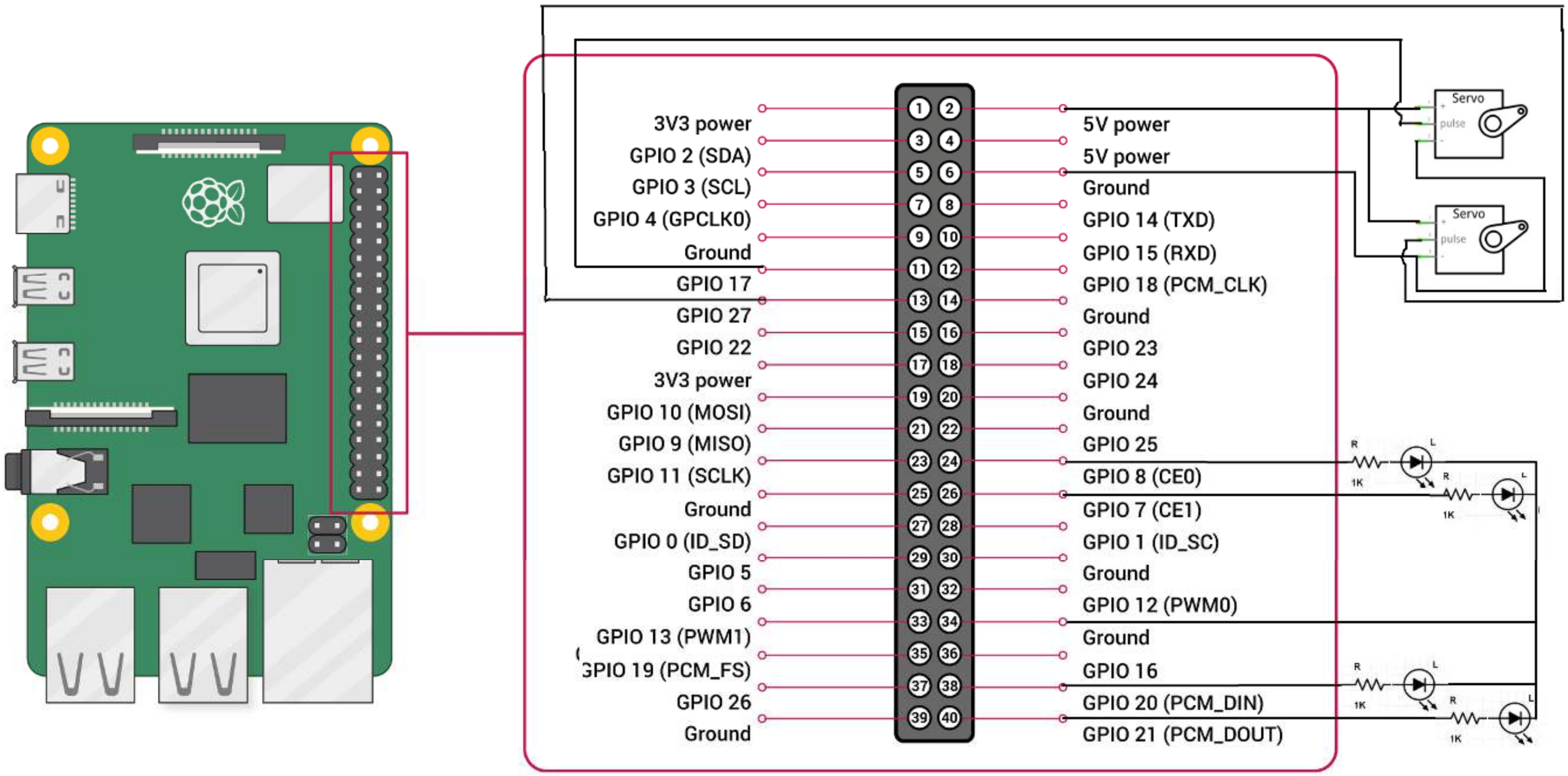


Fig 7 - Connection Diagram for Energy Management

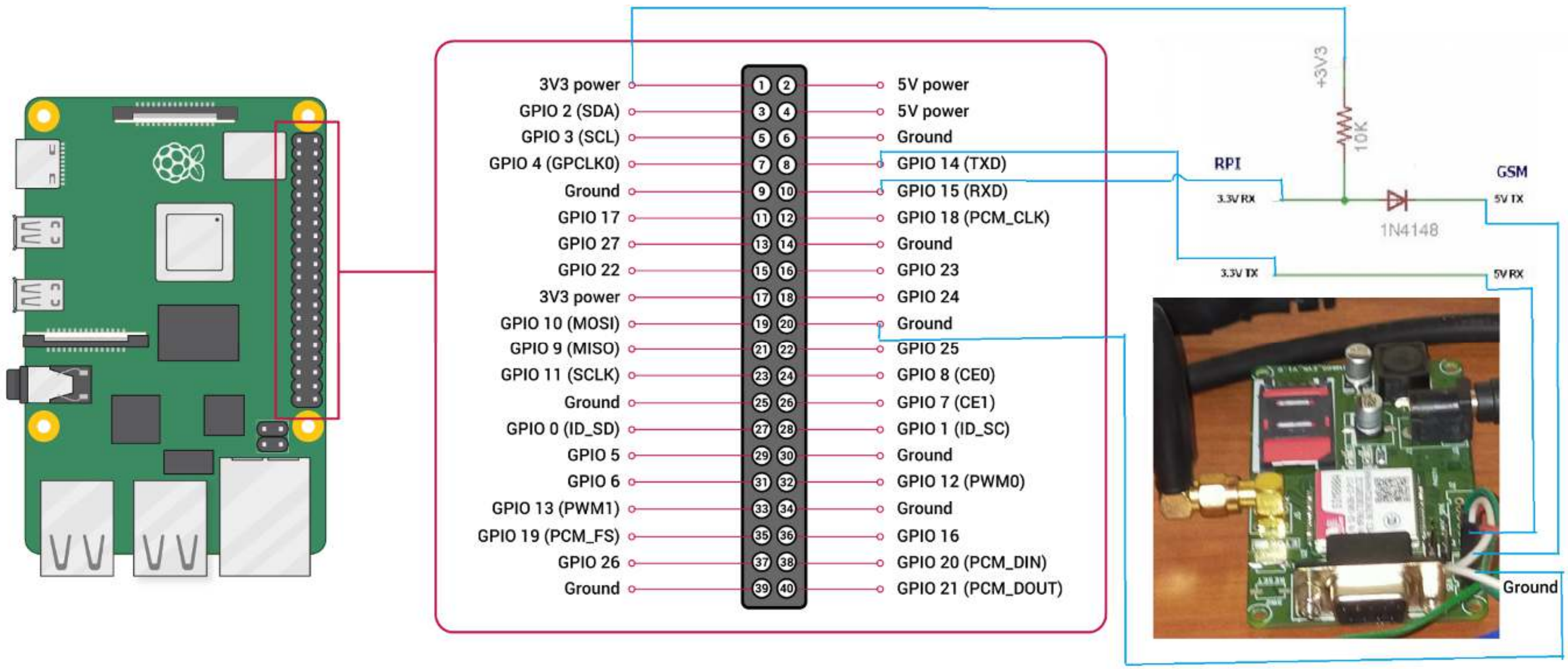


Fig 8 - GSM Interface

Chapter 4

SOFTWARE

4.1. Libraries used

4.1.1. OpenCV

OpenCV-Python is a library of Python bindings designed to solve computer vision problems.OpenCV-Python makes use of Numpy, which is a highly optimized library for numerical operations with a MATLAB-style syntax. All the OpenCV array structures are converted to and from Numpy arrays. This also makes it easier to integrate with other libraries that use Numpy such as SciPy and Matplotlib.

4.1.2. NumPy

NumPy is the fundamental package for scientific computing with Python. It contains among other things:

- a powerful N-dimensional array object
- sophisticated (broadcasting) functions
- tools for integrating C/C++ and Fortran code
- useful linear algebra, Fourier transform, and random number capabilities

Besides its obvious scientific uses, NumPy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined. This allows NumPy to seamlessly and speedily integrate with a wide variety of databases.

4.1.3. RPi. GPIO

RPi.GPIO is a popular Python library used on Raspberry Pi platforms to control GPIO pins.In addition to GPIO control, it is also used by many other libraries to query the Raspberry Pi hardware version as header pin layouts differed between certain versions. And, currently, it also provides some useful software PWM functionality on all GPIO pins.

4.1.4. I/O

I/O (or, informally, io or IO) is the communication between an Information processing system, such as a computer, and the outside world, possibly a human or another information processing system. I/O devices are the pieces of hardware used by a human (or other system) to communicate with a computer. For instance, a keyboard or computer mouse is an input device for a computer, while monitors and printers are output devices.

4.1.5. Picamera

picamera, which provides a pure Python interface to the camera module. And best of all, to use picamera to capture images in OpenCV format. The standard picamera module provides methods to interface with the camera, we need the (optional) array sub-module so that we can utilize OpenCV. Remember, when using Python bindings, OpenCV represents images as NumPy arrays — and the array sub-module allows us to obtain NumPy arrays from the Raspberry Pi camera module.

4.1.6. Serial

This module encapsulates the access for the **serial** port. It provides backends for **Python** running on Windows, OSX, Linux, BSD (possibly any POSIX compliant system) and IronPython. The module named “**serial**” automatically selects the appropriate backend.

4.1.7. sys

import sys in python is loading the module named sys into the current namespace so that we can access the functions and anything else defined within the module using the module name. One of the most common items is the list of arguments created when the program was called.

4.1.8. Time

This module provides various time-related functions. It returns a **floating point value** that represents the number of seconds that have passed since the **epoch**. The epoch is a a **platform-dependent** point where the time starts.

4.2. Software Algorithms

4.2.1. Head count and crowd control.

Face detection is accomplished in software through frontalface cascade routine[1] of OpenCV . This routine is based on pretrained Haar cascade models for faces. The Raspberry-Face-Recognition master/haar cascade_frontal face mmodels are downloaded in to the controller memory and specified as classifier for cascade detection. The main advantage of the cascade used in detection is rapid detection of faces and the quick discard of portions in image unlikely to contain faces.

The current implementation uses frontal faces detection methodology and is captured in the fig 8. The detection runs without any degradation in performance in the form of either delay or missing reading of frame) with frame rate of 15 frames per sec at the highest resolution of Pi camera (1920x1080). The headcount determined for the hall is then checked against the hall capacity for triggering the alarm for crowd control through GSM interface to relevant authority.

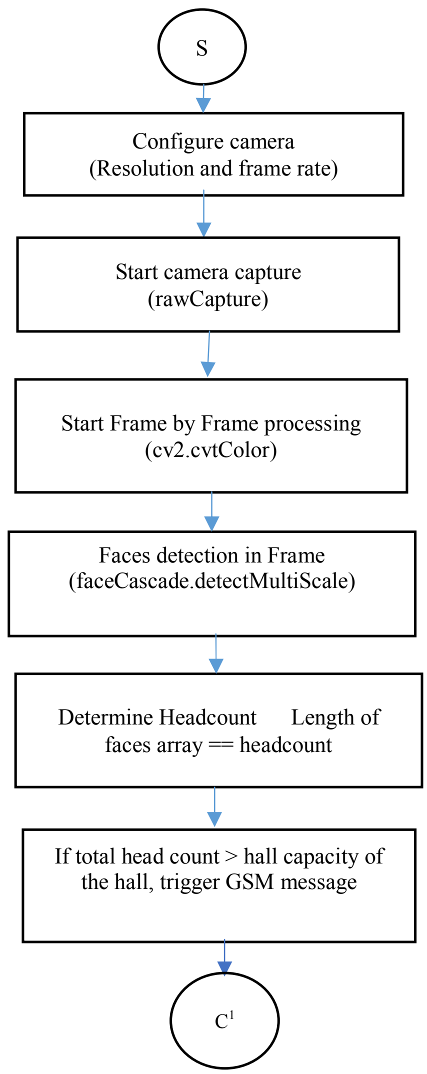


Fig. 8. Flow diagram of Head count and Crowd Control

4.2.2. Sector wise head count and Energy savings Scheme

The sector wise head count is computed by evaluating the x and y co-ordinates of each of the faces detected and determining the sector where the face lies. For this determination the left top corner of the detected face indicated by the x, y coordinates is used as the criteria and hence duplicate counting for the faces overlapping with multiple sectors is eliminated. Based on the sector wise head count being greater than zero, the light for the specific sectors represented by LED is turned ON .

Based on any of top and bottom sectors in specific half being greater than zero , fan for the specific sectors represented by servomotor is turned ON . This scheme is used as a sample for implementation . However this algorithm can accommodate any energy management scheme suitable for all halls. Extending the coverage of energy efficiency in terms of the number of sectors or adding utility items is easily handled in software. With a wifi capable hardware such as Raspberry Pi , the switching on and off commands can be extended to wifi-compatible switches. The software flowchart for sector wise head count and energy saving is shown in fig 9.

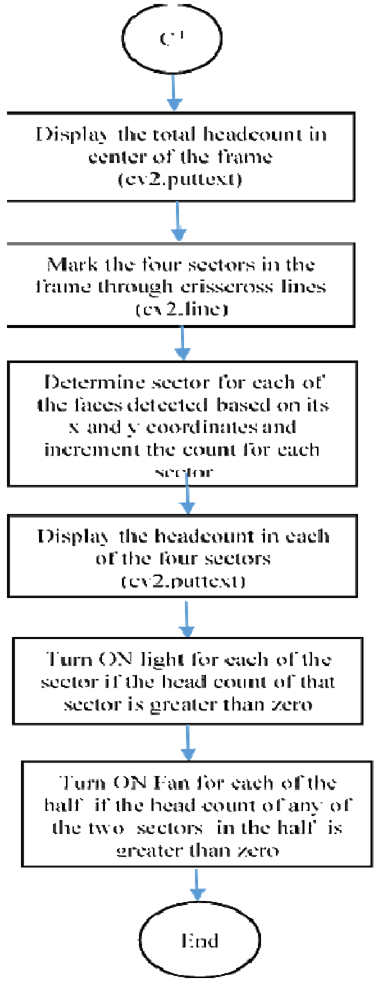


Fig. 9. Flow diagram of Sectorwise count and energy svaing

4.2.3. Social Distancing monitoring (SodisCop feature).

The current Covid-19 pandemic has mandated social distancing of 1metre as a critical safety guideline throughout the world by World Health Organization [6]. In this project this project a novel way of determining the social distancing is implemented . The algorithm has been captured as a flow diagram in fig. 10.

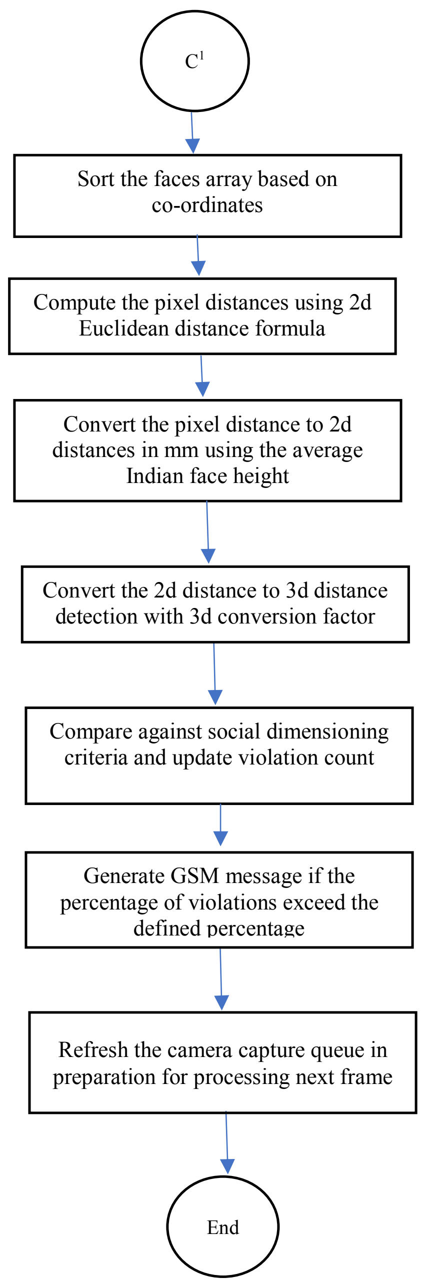


Fig 10 – Flow diagram of Social distancing Monitoring

The frame with the faces detected as part of the headcount forms the basis of determining the distances between each person in the hall. The x, y co-ordinates in pixels of the left top corner (left zygion) of each face captured by face cascade routine represents the location of the faces and are used in computing the distances between the faces and other faces using Euclidean distance formula1.These distances in pixels is converted to physical distances in mm using a scaling factor based on the average facial height of Indians [4]. To improve the accuracy of distance computation, the average of heights is computed for the two faces for which distance measurement is being computed.

This distance which lies in 2d plane is compensated for the depth introduced by the 3d space for which camera frame is captured. In this algorithm, the variation of the heights of the faces caused by the depths is used as a means of compensating the 2d distance and compute the 3d distance. The algorithm is designed to accommodate the fact that 3d distance is always greater than the 2d distance by maintaining the compensation factor greater than or equal to unity

The distances computed is compared with World Health Organization (WHO) guideline of 1metre [6] to trigger the count of social distancing of violations. The overall social distancing compliance status in the hall is measured as percentage of violations to the total number of distances measured. When the percentage of violations exceeds 10%, a message is generated through GSM to the relevant authority. The 10% percent violations limit will also avoid any intermittent erroneous detection and annunciation of social distancing violations. Any single violation as long as headcount is less than 10 will trigger the social distancing violation message. The limit also serves to avoid any spurious/borderline violation triggering in case of larger head count complying with social distancing most of the times.

The algorithm without computational complexity does not result in any significant delay in responding to social distance violation. Current social distance measurement between left zygions of faces can be easily switched to left eye to eliminate any impact of masks.

Chapter 5

RESULTS

The results are summarized in for each of the below major functions of the project.

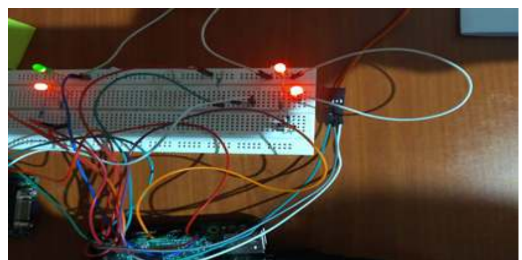
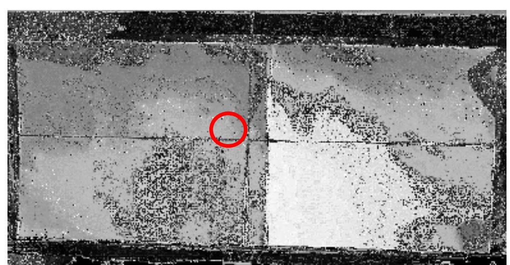
1. Head count and Crowd control
2. Energy Management
3. Social distancing

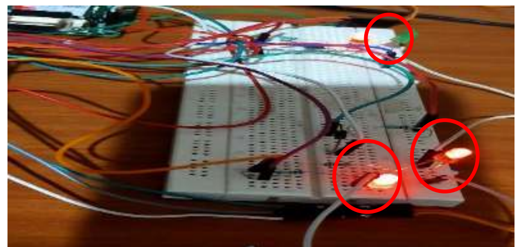
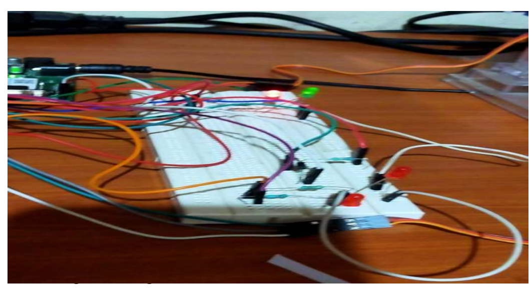
5.1. Head count and Crowd control

Sl. #	Description	Results
1	**HEAD COUNT AND CROWD CONTROL** **Test case 1** The frame is divided into four sectors and the headcount detected of each sector is shown in Fig 11**.** Head count algorithm based on face detection algorithm was able to capture the headcount with 100% accuracy for all the four sectors of the frame. **Test Case 2** As a corner case, the headcount was tested with a crowd image of more than 60 people. The algorithm detected 60 faces accurately as shown in Fig 12**.** The detection accuracy is dependent on the resolution of the camera. **Test case 3** This test is for crowd management. Once the headcount in the hall exceeds its capacity, a message is sent to the respective in charge person that "Hall Capacity is exceeded" so that immediate action can be taken to regulate the crowd. Fig 13 captures the result of this case including GSM message sent to the cell phone.	  Fig 12 – Head count detection for a crowd > 60 people  Fig 13 – Crowd Management

5.2. Energy Management.

Energy devices control based on sector wise head count was demonstrated through LEDs and Servomotors representing lights and fans respectively. While four LEDs represented each of the four sectors , fans in each half (left and right) of the hall were simulated through two servo motors

Sl. #	Description	Results
2	**ENERGY MANAGEMENT** **Test Case 1**: This case is for all the four sectors of the frame having headcount greater than zero as shown in fig 14	 Fig 14 – Energy Management with all four sectors occupied
	The lights corresponding to each of the four sectors represented by four LEDs are switched ON as shown in Fig 15	 Fig 15 – Lights (LEDs) for all four sectors ON
	Test Case 2 Zero head count in hall is simulated by focusing the camera on to a blank screen (blanked out with a white paper). The head count is detected as zero for all the four sectors as indicated in Fig 16.	 Fig 16 – Zero Head count in all four sectors
	All the four sectors' lights are now commanded off by the controller as shown in Fig 17	 Fig 17– All lights are switched OFF

Sl. #	Description	Results
2	**ENERGY MANAGEMENT (contd.)** **Test Case 3**: Left top sector headcount made zero (blanked out). As shown in Fig 18, the headcount for that sector is correctly computed by the algorithm as zero. The lights corresponding to Left top sector is switched OFF in line with energy management objective as shown in Fig 19 **Test Case 4** This test case is for Fans control portion of the energy management. The fans are simulated by two servo motors one for each half of the auditorium. The right half of the auditorium is blanked out as shown in Fig 20. Servo motors and LEDs corresponding to the right half of the auditorium are controlled OFF as part of energy management. The right servo motor at zero degrees is shown in fig 15 and the LEDS in fig 21	 Fig 18 – Head count detected zero for the blanked sector  Fig 19 – Light for zero headcount sector is OFF  

5.3. Social distancing

Sl. #	Description	Results
3	**Social distancing** The test involved three-member audience in a living room (fig 22). the green lines indicate that social distancing criteria > 1metre being met while the pink lines indicate the social distancing is violated (< 1metre). With only one of the three distance measurements meeting the social distancing criteria, a GSM message is generated indicating social distancing violations and sent to the cell phone for relevant authority as shown in screen capture in Fig 23.	 Fig 22– social distancing monitoring  Fig 23– social distancing violation message

5.4. Validation of the social distancing algorithm through computation

A sample study of 24 occupants in a frame (Fig 25) indicates healthy congruence between the distances computed from the algorithm used in this project and computed values from the physical positioning of audience. The error exceedances marked in red were analyzed and found to be due to variations in facial dimensions (one of them is due to a foreigner being part of the group). Social distancing violations accuracy is 100% and includes one safer detection as a violation of a border line computed value. The results of this validation are captured in table 4 below.

TABLE 4 – Social distancing algorithm Validation

x	y	h	2d distance as per Euclidian formula (mm)	3d adjusted as per algorithm used in this project (mm)	Depth(mm)	computed value (based on physical position) in mm	Percentage Error
871	710	71	0				
1072	705	72	337.4470144	342.1997893	0	337.4470144	1.4084507
1310	687	75	722.6335772	763.345328	0	722.6335772	5.6338028
693	676	71	306.2841138	306.2841138	0	306.2841138	0
570	633	74	514.2501092	535.978987	150	535.6801049	0.0557949
468	621	78	664.7686749	730.3092485	150	681.4817614	7.1649001
1129	620	74	452.2711297	471.3811775	200	494.5191349	-4.6788801
1546	615	90	1016.127807	1288.049333	200	1035.623349	24.374304
931	601	70	211.7832625	214.8087377	200	291.2939242	-26.257048
748	592	73	284.0823433	292.0846628	200	347.423053	-15.928244
1486	590	70	1066.549714	1081.786139	200	1085.139758	-0.3090495
1332	572	67	836.8904548	886.854064	200	860.4566423	3.0678387
1232	537	70	681.3827711	691.1168107	250	725.7978236	-4.7783297
480	518	77	706.3739855	766.0675618	300	767.4400351	-0.1788379
847	500	64	375.763519	416.8626539	300	480.8307626	-13.303664
1039	500	72	451.3530931	457.7101789	300	541.9590526	-15.545247
667	463	74	530.2367497	552.6411194	400	664.1919984	-16.794975
1430	461	77	992.3502881	1076.210876	400	1069.934154	0.5866456
1197	452	67	723.0266149	766.1923829	400	826.297456	-7.2740237
1362	434	70	958.7331312	972.4293188	400	1038.830697	-6.3919345
854	405	64	543.0638277	602.4614339	450	705.2788959	-14.57827
1026	405	61	622.0467416	724.0216173	450	767.7513587	-5.6958208
701	384	60	673.5808655	797.0706908	450	810.0686282	-1.6045477
1172	347	72	791.4314463	802.5783681	500	936.1430095	-14.267547

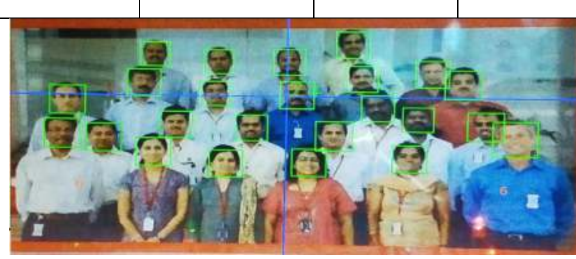


Fig 25 – sample frame used for validation

Chapter 6

APPLICATIONS AND ADVANTAGES

6.1. ADVANTAGES OF THE PROJECT

First and foremost, effective crowd management helps to ensure the safety of those at an event happening in the hall/ auditorium, from the guests to the staff, and the performers. When an event is taking place, everyone in the hall should be able to enjoy themselves without worrying about their safety.

The consequences of a poorly managed crowd can be disastrous; people can be injured and lives can be lost. Even if an event in the hall is poorly managed but nobody is hurt, it may not bode well for future events; people may have less trust in a hall management, and sanctions can be imposed for poor event management.

Everything in the hall/ auditorium needs to be managed properly. If anything is poorly managed, that can lead to frustration of audience. This frustration can lead to tempers fraying and altercations happening, so effective management can help minimise the risk of these occurrences.

The effective management of power equipment in the hall/ auditorium is another advantage where it can ensure no loss in power by efficiently managing lights and fans in the hall.

Maintenance of required social distance between people in the hall/ auditorium is another self-advantage, where the audience need not worry about their safety and can enjoy the event happening in the hall/ auditorium.

6.2. APPLICATIONS OF THE PROJECT

The project finds its application in many places like:

Schools and college auditoriums and seminar halls.

Movie theatres.

Entertainment centers.

Company's seminar halls, conference halls and auditoriums.

Open air theatres.

Function halls or conventional halls.

In public places.

Chapter 7

CONCLUSIONS AND SCOPE FOR FUTURE WORK

7.1. CONCLUSION

This project addresses the critical challenges having useful application in current environment energy savings, crowd control and last but not least social distancing, a very important aspect of Covid-19 pandemic. The algorithms used in this project are easy to implement, devoid of complexity and has the advantage of scalability to address halls with larger physical dimensions. The software is designed with portability in mind and is based on open source. The cost of this system is low and with the right positioning of a camera with moderate features to focus on audience alone in full frame, the method is found to deliver accurate results meeting the objectives of this project. The software's portability provides another valuable opportunity of integrating in to video surveillance applications with existing CCTV in auditoriums/public places.

7.2. SCOPE FOR FUTURE WORK

This project is based on frontal face detection and hence there is scope for future work to cover lateral face detection as well as human contour detection. This project uses average facial height-based distance estimation in an image and is independent of the focal length camera used. The accuracy of the distance measurement can be enhanced by inclusion of the focal length of the camera as future work.

REFERENCES

Paul Viola, Michael, "Rapid Object Detection using a Boosted Cascade of SimpleFeatures",conference on computer vision and pattern recognition, 2001.

'Damien LEFLOCH' , "Real-Time People Counting system using Video Camera", pp.26, 2009) .

Sani Md. Ismail, Shaikh Mohammad Fahim, Mahmood Reaz, "A Simple Real-Time People Counter with Device Management System Using Digital Logic Design", International Journal of Scientific & Engineering Research Volume 3, Issue 8, ISSN 2229-5518, 2012.

Prasanna LC1, Bhosale S2, D'Souza AS 3, Mamatha H4, Thomas RH, Sachin KS, "Facial indices of North and South Indian Adults: Reliability in Stature Estimation and Sexual Dimorphism", DOI: 10.7860/JCDR/2013/5497.3204,2013.

Face Detection using Haar Cascades https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_objdetect/py_face_detection/py_face_detection.html.

World Health organization United nations "Protecting yourself and others from the spread COVID-19", https://www.who.int/emergencies/diseases/novel-coronavirus-2019/advice-for-public.

A. C. Davies, J. H. Yin, and S. A. Velastin, "Crowd monitoring using image processing," Electronics and Communications Engineering Journal, vol. 7, pp. 37-47, 1995.

Hongquan Songa, Xuejun Liua, Xingguo Zhang, Jiapei Hua," Real-time Monitoring for Crowd Counting usingVideo Surveillance and GIS", 10.1109/RSETE.2012.6260673-June 2012.

Narinder Singh Punn, Sanjay Kumar Sonbhadra and Sonali Agarwal, "Monitoring COVID-19 social distancing with person detection and tracking via fine-tuned YOLO v3 and Deepsort techniques", arXiv:2005.01385v2 [cs.CV], 6 May 2020.

Hashimoto, K., Morinaka, K., Yoshiike, N, Kawaguchi, C. And Mat-sueda, S., "People Count System Using Multi-Sensing Application", International Conference On Solid State Sensors And Actuators, TRANSDUCERS '97 , Chicago, 2, JUN 1997, pp.1291-1294.

Konstantinos T. Papakostas, Ioannis Tiganitis, and Agis M. Papadopoulos, "Energy And Economic Analysis Of An Auditorium's Air Conditioning System With Heat Recovery In Various Climatic Zones", Jan' 2018.

A. Albiol I. Mora ; V. Naranjo, "Real-time high density people counter using morphological tools", IEEE Transactions on Intelligent Transportation Systems, Volume: 2 , Issue: 4 , Dec 2001.

APPENDIX A

Code – Head count , crowd control and Energy Management

```
#import the necessary packages
from picamera.array import PiRGBArray
from picamera import PiCamera
import time
import cv2
import RPi.GPIO as GPIO # Import Raspberry Pi GPIO library
from time import sleep # Import the sleep function from the time module

import serial
import time,sys
topleftcount=toprightcount=bottomleftcount=bottomrightcount=0




cascadePath='/home/pi/Downloads/Raspberry-Face-Recognition-master/haarcascade_frontalface_default.xml'
faceCascade=cv2.CascadeClassifier(cascadePath)

# initialize the camera and grab a reference to the raw camera capture
camera = PiCamera()
camera.resolution = (1920, 1080)
camera.framerate = 15
rawCapture = PiRGBArray(camera, size=(1920, 1080))

leftservo_pin = 17 # Initializing the GPIO 17 for leftservo motor
rightservo_pin = 27 # Initializing the GPIO 27 for rightservo motor
GPIO.setmode(GPIO.BCM) # We are using the BCM pin numbering

sleep(1)
#LEDIO Ports
GPIO.setwarnings(False) # Ignore warning for now
GPIO.setup(20, GPIO.OUT, initial=GPIO.LOW) # Set pin 38 to be an output pin and set initial value to low (off)
GPIO.setup(21, GPIO.OUT, initial=GPIO.LOW) # Set pin 40 to be an output pin and set initial value to low (off)
GPIO.setup(7, GPIO.OUT, initial=GPIO.LOW) # Set pin 26 to be an output pin and set initial value to low (off)
GPIO.setup(8, GPIO.OUT, initial=GPIO.LOW) # Set pin 24 to be an output pin and set initial value to low (off)
#servo motors setup
GPIO.setup(leftservo_pin, GPIO.OUT) # Declaring GPIO 17 as output pin
GPIO.setup(rightservo_pin, GPIO.OUT) # Declaring GPIO 27 as output pin
lpwm = GPIO.PWM(leftservo_pin, 50) # Created PWM channel at 50Hz frequency
rpwm = GPIO.PWM(rightservo_pin, 50) # Created PWM channel at 50Hz frequency
lpwm.start(2.5)
rpwm.start(2.5)

SERIAL_PORT="/dev/ttyS0"
ser=serial.Serial(SERIAL_PORT,baudrate=9600,timeout=5)


# allow the camera to warmup
```

```
time.sleep(0.1)
# capture frames from the camera
for frame in camera.capture_continuous(rawCapture, format="bgr", use_video_port=True):
# grab the raw NumPy array representing the image, then initialize the timestamp
# and occupied/unoccupied text
image = frame.array
gray = cv2.cvtColor(image,cv2.COLOR_BGR2GRAY)
#GPIO.output(20, GPIO.LOW)
# Get all face from the video frame
faces = faceCascade.detectMultiScale(gray, 1.2,5)
print("Total headcount",(len(faces)))
font = cv2.FONT_HERSHEY_DUPLEX
cv2.putText(image, str(len(faces)), (960, 540), font, 2.0, (0,255,0), 1)
if(len(faces)>30):
ser.write("AT+CMGF=1\r")
print("Text mode enabled....")
time.sleep(0.5)
ser.write('AT+CMGS="9886008420"\r')
msg="Hall Capacity Exceeded...."
time.sleep(0.5)
ser.write(msg+chr(26))
time.sleep(0.5)
print("message sent")
# Draw the four sector boundaries
cv2.line(image, (0,540), (1920,540), (255,0,0), 2)
cv2.line(image, (960,0), (960,1080), (255,0,0), 2)
for(x,y,w,h) in faces:
# Create rectangle around the face
cv2.rectangle(image, (x-5,y-5), (x+w+5,y+h+5), (0,255,0), 2)
if(x<=960):
if(y<=540):
topleftcount=topleftcount+1
else:
bottomleftcount=bottomleftcount+1
else:
if(y<=540):
toprightcount=toprightcount+1
else:
bottomrightcount=bottomrightcount+1
print("TopLeftcount=",topleftcount)
print("TopRightcount=",toprightcount)
print("BottomLeftcount=",bottomleftcount)
print("BottomRightcount=",bottomrightcount)
cv2.putText(image, str(topleftcount), (479, 269), font, 1.0, (0,0,255), 1)
cv2.putText(image, str(toprightcount), (1439, 269), font, 1.0, (0,0,255), 1)
cv2.putText(image, str(bottomleftcount), (479, 809), font, 1.0, (0,0,255), 1)
cv2.putText(image, str(bottomrightcount), (1439, 809), font, 1.0, (0,0,255), 1)
# show the frame
cv2.imshow("Frame", image)
#Turn on lights based on headcount
if(topleftcount>0):
GPIO.output(20, GPIO.HIGH)
else:
GPIO.output(20, GPIO.LOW)
```

```
if(bottomleftcount>0):
GPIO.output(21, GPIO.HIGH)
else:
GPIO.output(21, GPIO.LOW)
if(toprightcount>0):
GPIO.output(7, GPIO.HIGH)
else:
GPIO.output(7, GPIO.LOW)
if(bottomrightcount>0):
GPIO.output(8, GPIO.HIGH)
else:
GPIO.output(8, GPIO.LOW)
if(topleftcount>0) or (bottomleftcount>0):
lpwm.ChangeDutyCycle(2.5) # Move servo to 0 degrees
sleep(0.1) # Delay of 1 sec
lpwm.ChangeDutyCycle(7.5) # Move servo to 90 degrees
sleep(0.1)
lpwm.ChangeDutyCycle(12.5) # Move servo to 180 degrees
sleep(0.1)
lpwm.ChangeDutyCycle(7.5)
sleep(0.1)
lpwm.ChangeDutyCycle(2.5)
sleep(0.1)
if(toprightcount>0) or (bottomrightcount>0):
rpwm.ChangeDutyCycle(2.5) # Move servo to 0 degrees
sleep(0.1) # Delay of 1 sec
rpwm.ChangeDutyCycle(7.5) # Move servo to 90 degrees
sleep(0.1)
rpwm.ChangeDutyCycle(12.5) # Move servo to 180 degrees
sleep(0.1)
rpwm.ChangeDutyCycle(7.5)
sleep(0.1)
rpwm.ChangeDutyCycle(2.5)
sleep(0.1)
key = cv2.waitKey(1) & 0xFF

# clear the stream in preparation for the next frame
rawCapture.truncate(0)
topleftcount=0
toprightcount=0
bottomleftcount=0
bottomrightcount=0
# if the `q` key was pressed, break from the loop
if key == ord("q"):
break
```

Code -Social distance monitoring

```
# import the necessary packages
from picamera.array import PiRGBArray
from picamera import PiCamera
import time
import cv2
import numpy as np
import array as arr
import serial
from time import sleep # Import the sleep function from the time module
import RPi.GPIO as GPIO # Import Raspberry Pi GPIO library
cascadePath='/home/pi/Downloads/Raspberry-Face-Recognition-master/haarcascade_frontalface_default.xml'
faceCascade=cv2.CascadeClassifier(cascadePath)
SERIAL_PORT="/dev/ttyS0"
ser=serial.Serial(SERIAL_PORT,baudrate=9600,timeout=5)
sumofheight=0
averageheight=0.0
heightcount = 0
refx=0
refy=0
refh=0
completedpass=0
pixeldistance=0.0
twod_distance=0.0
threed_distance=0.0
indianfaceheight=119.5
socialdistance =1000.0
okcount=0
notokcount=0
percentageviolations=0.0
threedcorrectionfactor=1.0
# initialize the camera and grab a reference to the raw camera capture
camera = PiCamera()
camera.resolution = (1920, 1080)
camera.framerate = 15
rawCapture = PiRGBArray(camera, size=(1920, 1080))
def sortkey(n):
return n[0]
# allow the camera to warmup
time.sleep(0.1)
# capture frames from the camera
for frame in camera.capture_continuous(rawCapture, format="bgr", use_video_port=True):
# grab the raw NumPy array representing the image, then initialize the timestamp
# and occupied/unoccupied text
image = frame.array
gray = cv2.cvtColor(image,cv2.COLOR_BGR2GRAY)
# Get all face from the video frame
faces = faceCascade.detectMultiScale(gray, 1.2,5)
sortedfaces = sorted(faces,key = sortkey)
if len(sortedfaces) >0:
for passcount in range(0,len(sortedfaces)):
for(x,y,w,h) in sortedfaces:
# Create rectangle around the face
```

```

cv2.rectangle(image, (x-20,y-20), (x+w+20,y+h+20), (0,255,0), 4)

pixeldistance = np.sqrt([((x-refx)**2+(y-refy)**2)])

twod_distance = (((pixeldistance)*2)/(refh+h))*indianfaceheight
if (h>refh):
threedcorrectionfactor =h/refh
else:
threedcorrectionfactor =refh/h
threed_distance = twod_distance*threedcorrectionfactor
#print(twod_distance,threed_distance)
if (threed_distance < socialdistance):
cv2.line(image, (x,y), (refx,refy), (255,0,255), 2)
notokcount = notokcount+1
else:
cv2.line(image, (x,y), (refx,refy), (0,255,0), 2)
okcount = okcount+1
refx = sortedfaces[passcount][0]
refy = sortedfaces[passcount][1]
refh = sortedfaces[passcount][2]

percentageviolations=(((notokcount)*100)/(okcount+notokcount))
print(okcount, notokcount)
print(percentageviolations)
if(percentageviolations>10.0):
ser.write("AT+CMGF=1\r")
print("Text mode enabled....")
time.sleep(0.5)
ser.write('AT+CMGS="9886008420"\r')
msg="social distancing violated"
time.sleep(0.5)
ser.write(msg+chr(26))
time.sleep(0.5)
print("message sent")

# show the frame
cv2.imshow("Frame", image)
okcount=0
notokcount=0
key = cv2.waitKey(1) & 0xFF

# clear the stream in preparation for the next frame
rawCapture.truncate(0)

# if the `q` key was pressed, break from the loop
if key == ord("q"):
break
```