Automatic Door Monitoring and Smart Lighting System using WOT

A PROJECT REPORT

Submitted by

Keshav Sharma (15BCE1341)

Neeraj Kumar (15BCE1129)

Ayush Sharma (15BCE1126)

in the partial fulfilment for the award

of

Bachelor of Technology

in

Computer Science and Engineering

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

Vandalur - Kelambakkam Road, Chennai - 600 127

April - 2019

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

School of Computing Science and Engineering

DECLARATION

We hereby declare that the project entitled "Automatic Door Monitoring and Smart Lighting System using WOT" submitted by us to the School of Computing Science and Engineering, Vellore Institute of Technology, Chennai, Chennai 600127 in partial fulfilment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering is a record of bonafide work carried out by us under the supervision of Prof. Maheswari N. We further declare that the work reported in this project has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma of this institute or of any other institute or university.

Place: Chennai

Signature of Candidate (s)

Keshav Sharma (15BCE1341)

Date:

Neeraj Kumar (15BCE1129)

Ayush Sharma (15BCE1126)

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

School of Computing Science and Engineering

CERTIFICATE

This is to certify that the report entitled "Automatic Door Monitoring and Smart Lighting System using WOT" is prepared and submitted by Keshav Sharma (15BCE1341), Neeraj Kumar (15BCE1129) and Ayush Sharma (15BCE1126) to VIT Chennai, in partial fulfillment of the requirement for the award of the degree of B.Tech CSE Programme is a bona-fide record carried out under my guidance. The project fulfills the requirements as per the regulations of this University and in my opinion meets the necessary standards for submission. The contents of this report have not been submitted and will not be submitted either in part or in full, for the award of any other degree or diploma and the same is certified.

Guide	Program Chair
Name: Prof. Maheswari N.	Name: Dr. B Rajesh Kanna
Date:	Date:

External Examiner	Internal Examiner	Internal Examiner
Signature: Name: Date:	Signature: Name: Date:	Signature: Name: Date:

ACKNOWLEDGEMENT

We wish to express our sincere thanks to a number of people without whom we could not have completed the thesis successfully.

We would like to place on record our deep sense of gratitude and thanks to our guide Prof. Maheswari N, School of Computer Science and Engineering (SCSE), VIT, Chennai campus whose esteemed support and immense guidance encouraged us to complete the project successfully.

We would like to thank our Program Chair Dr. B Rajesh Kanna & Co-Chair, Dr. C Sweetlin Hemalatha, B.Tech. Computer Science and Engineering and Project Co-ordinator Dr. B V A N S S Prabhakar Rao, VIT Chennai, for their valuable support and encouragement to take up and complete the thesis.

Special mention to our Dean to Dr. Vaidehi Vijayakumar, Associate Deans Dr. Vijayakumar V and Dr. Subbulakshmi T, School of Computing Science and Engineering (SCSE), VIT Chennai campus, for spending their valuable time and efforts in sharing their knowledge.

We thank our management of VIT, Chennai for permitting us to use the library resources. We also thank all the faculty members for giving us the courage and the strength that we needed to complete our goal. This acknowledgment would be incomplete without expressing the whole hearted thanks to our family and friends who motivated us during the course of our work.

Keshav Sharma (15BCE1341)

Neeraj Kumar (15BCE1129)

Ayush Sharma (15BCE1126)

Table of Contents

List of Tables

List of Figures

List of Abbreviations

Sr. No.	Abbreviations	Full Form	Description
1.	RPI	Raspberry Pi	Name of a low powered microcontroller.
2.	IOT	Internet of things	The technology used in the project
3.	WOT	Web of Things	Improvised form of IOT
4.	IP	Image Processing	Concept behind security system
5.	OpenCV	Open Computer Vision	A tool for Image processing
6.	LDR	Light Dependent Resistors	It is a device that changes it resistance depending upon light intensity.
7.	IR	Infrared	A type of radiation
8.	API	Application Program Interface	A set of protocols and subroutines for creating applications
9.	SDLC	Software Development Life Cycle	The stages involved in the project management

ABSTRACT

The most important considerations in current technologies are automation, energy consumption and profitability. Automation aims to reduce the workforce through the use of intelligent systems. The project aims to describe a method of controlling lights and appliances with infrared sensors with minimal electrical energy consumption. When presence is detected, the streetlights are lit or off. Whenever a person crosses particular area, the lights on and off depend on the distance that is recorded by the infrared sensor.

This project has an automatic door monitoring system that uses Raspberry Pi to recognize the face with the camera. If the authorized person has come, the door is unlocked, otherwise it is locked. However, the current security system has many weaknesses, where it is simply expressed. Most doors are controlled by persons with keys, security cards, counter-signs or models for opening doors. This project is intended to help users improve door safety for sensitive areas by using face recognition. The system consists mainly, image capture, face recognition and recognition, e-mail notification, and automatic door access management. OpenCV is used for face reorganization because it uses in-face and scaled facial images without losing the essential features. The door lock can also be accessed from anywhere in the world via the Android app. The image taken by the Pi camera will be emailed to the authorized person for security reasons.

It describes the design and development of a remote-control system for appliances with Raspberry Pi and Android mobile phones. If everything is embedded or connected with Raspberry Pi one can control most of the appliances from anywhere in the world. Smart lighting is controlling the lights and appliances in the home and office, whether a person is available in the place or elsewhere the reliable can be allowed remotely inside the building by unlocking the door using android application.

1. INTRODUCTION

1.1 Problem Statement

It's very difficult for old and handicapped people to live independently. This project will help elderly or handicapped people to live a better independent life as long as possible. Existing system and proposed system have same costing all the way but making extensive use of it by bridging two microprocessors over WIFI module and running the system synchronously is a complete automated system and this is the main objective of the project. System has been trained under various situation likely to come for everyone and automating it accordingly for the best working.

The traditional works is available in different parts and the communication as it was happening with Bluetooth which can be slower if used so frequently. The existing work can't be interfaced with a different system and it has many drawbacks when it comes to this project, whole system is on WIFI and can be connected easily.

Use of the emerging technologies such as WOT (Web of things), Image processing makes it possible to exist as one system solving multiple problems including security, home automation very efficiently. A common platform for multiple modules of the project will give a different approach to keep an eye on everything using a single mobile application.

1.2 Objective

Today, in all areas of daily life, it's often said that technologies improve the efficiency of things and are easier to handle. Life itself is becoming more skillful and productive. It is getting smarter, safer and cheaper. Intelligent door monitoring and lighting system is a significant IoT-based feature in which all electrical and electronic appliances used in their own building to communicate with each other to perform functions and act in a way that is desirable to the occupants.

The aim of the proposed work is to set up such a system that can reduce the workload and energy loss, lead a comfortable life, improve the standard of living and elder and disabled people can remotely control their devices. This project aims to introduce a new security system based on Raspberry Pi. The main goal is to develop a prototype that can simulate wireless tasks,

including monitoring and controlling the digital door lock. Such a system would have the ability to provide safe and controlled electrical appliances.

The system can be used in many places, e.g. In banks, laboratories, hospitals and other sophisticated automated systems, reducing the risk of unauthorized access. There is a need for a simple, efficient and accessible lighting and door monitoring system that operates intelligently according to predefined user-defined parameters. This can turn a simple home into a true smart home. The main reason for developing this system is to save time and manpower as well as safety and comfort.

1.3 Motivation

In recent years, it is important to have a reliable security system that can secure the assets and protect the privacy. However, the current security system has many weaknesses, where it is simply expressed. Most doors are controlled by persons with keys, security cards, counter-signs or models for opening doors. The goals of this project are to help users improve door safety in sensitive areas by using face recognition.

Smart lighting has been proved energy efficient, convenient and economic so this system comes up with an improvised lighting system which can be in action with both manual and automatic manner. It is a good step to install a system like this which give an ease to operate electrical and to save energy.

2. PROJECT DESCRIPTION AND GOALS

2.1 Project Description

The system works with 3.3 and 5 V DC. The components connected to the Raspberry PI 3 are an infrared sensor and an IC ULN2803APG relay driver.

The project consists of two components: a door monitoring system with Raspberry Pi and 5V with a small transformer; the other smart lighting system running on a NodeMCU connected to the 3.3V transformer

Electrical appliances and the door motor installed in the house can be controlled with an Android application.

The relay driver is connected to three electrical devices such as lights and a lock. When the infrared sensor is detected by a person, the webcam captures the image and sends it via e-mail via raspberry Pi.

When a person stands in front of camera, it recognizes the image after being compared with the images stored in the database when it is validated, the door motor turns, however, when a strangely comes, system sends an image to the E -Mail. Once the image has been authenticated by the user who has the application in his smartphone, can then send that person via smartphone.

The third case is a unknown person with a known person, then the dc motor will rotate and door will be open. Once the door is open, the user sends the start signal of the light and the fan using the mobile application managed in smartphone.

2.2 Goals

- Making a user-friendly home automation system for old aged and disabled people.
- Reducing the energy consumption rate having a remote access over all appliances of the house.
- Alert the house owners of approaching unknown persons.
- Safety from any kind of stealth action and spying activities.
- To interface the two different microcontrollers on a common platform interconnected with each other.
- Automatic motion-based lighting that can also be controlled with app.

3. LITERATURE SURVEY

Today, global security at a fast and ever-changing pace is one of the basic needs of life. Use of Security and automation technology plays an important role in improving safety and reducing workload.

- Y. Januzaj, Luna A. 2015 [1] *Real-time access control proposal for face recognition with Raspberry Pi instead of GSM and relay services. The limitation of the job was that he could not control the situation of the backlight and the conditions of the ambient light.*
- M. Carikci, Ogen F. 2012 [2] *have proposed work on a facial recognition system based on the self-facial method and using the self-recognition method as well as the Euclidean distance method to compare the subject's image with the images in the database. It was a very efficient and fast method, but also very accurate.*
- H.Lwin, Khaing A., Tun H. 2015 [3] *proposed a door lock access system consisting of three subsystems: facial recognition, face recognition and automated access control. Face recognition is updated with PCA (Principal Component Analysis). The door opens for the known leader of the microcontroller and caution is required for the unknown. The downside to this system is that input images are continuously captured by a webcam until you press the Stop Camera button. Someone on the site must review the images or system status with unauthorized persons and take appropriate action. The microcontroller is connected to a personal computer (PC). The entire system does not work if the PC hangs or does not work.*
- G.senthilkumar.et.al., Gopalakrishnan K. 2014 [4] *proposed a work on the embedded image capture system using Raspberry Pi. In this work, they captured the image and compared it to the database, but the system could not function properly in ambient light conditions.*
- U. Sowmiya., Shafiq Mansoor K. 2015 [5] *Developed to connect any door with internet. In this system, the user has also implemented a sensor and a PIR camera. PIR sensor used to detect a person and camera used to capture the video of the person coming to the door. The video was sent via a 3g dongle to an authorized person. They had also discussed the benefits of this system. They had concluded the use of this system in banks, hospitals, etc., but their proposed model did not allow sending messages to authorized persons.*

- J. Kartik, Srimadhavan V. 2013 [6] *Have proposed two systems are proposed, one is based on GSM technology and others uses a webcam to detect the intruder. The first security system uses a web camera installed in the premises of the house, managed by software installed on the PC and using the Internet for communication. The camera identifies the movement of any intruder before the measurements or the range of the camera. The product communicates to the customer scheduled via the Internet an arrangement and in the meantime it triggers an audible alarm. The second security system is based on SMS and uses GSM innovation to send the SMS to the owner*

- Shrikanthan N. Tan F Karande A. 2009 [7] *Have proposed a system for home automation but it has some limitations you do not have quality devices for setting up your smart home via Bluetooth, you can't at least need to set everything up on yourself. It's worth taking the pain. To configure a Bluetooth home automation system, it must have a custom microcontroller, such as the 8051 microcontroller, as well as electromechanical relays capable of interfacing with the microcontroller, such as the power relay, the contactor, the delay relay, etc. communication between the micro-controller and your mobile phone requires a Bluetooth module.*

4. TECHNICAL SPECIFICATION

4. 1 Hardware Specification

4.1.1 NodeMCU ESP8266:

The ESP8266 is a low-cost WiFi module chip that can be connected to the Internet for the Internet of Things (IoT). They do not have the built-in configuration to do this. You can install the ESP8266 with this device and do incredible things. Control, monitoring, analysis and much more.

Main features of NodeMCU:

It can be programmed directly via the USB port with LUA or Arduino IDE programming. By simply programming, It can establish a WiFi connection and define input / output pins according to your needs, just like Arduino. This will make a web server and much more. It provides support for bidirectional telemetry and flight control using the MAVLink protocol.

10 GPIO D0-D10, PWM functionality, IIC and SPI communication, 1-Wire and ADC A0, etc. All in one card.

Wi-Fi network (can be used as an access point and / or station, host a web server), connect to the Internet to retrieve or download data.

4.1.2 Raspberry Pi Quad Core Cortex A53:

The Raspberry Pi 3 Model B is a computer the size of a credit card. Just add a keyboard, a mouse, a monitor, a power adapter and a microSD card with Linux distribution installed, and you have a full-fledged computer that can run applications ranging from word processor software and spreadsheets to applications.

Since the Raspberry Pi 3 supports HD video, it can even create a Media Center. The Raspberry Pi 3 Model B is the first Raspberry Pi to be open source from the start and should be the de facto embedded Linux card in all forums.

4.1.3 Raspberry PI 5MP Camera Board Module

The Raspberry Pi Camera v2 is the new official camera card released by the Raspberry Pi Foundation, a high quality 8 megapixel Sony IMX219 expansion card for the Raspberry Pi, designed specifically for the image sensor. One of the small sockets is located on the top of the card and uses the dedicated Csi interface, which was specially developed for connection to the cameras.

4.1.4 DC Motor:

A DC motor is part of a class of rotary electric machines that convert DC electrical energy into mechanical energy. The most common types are based on the forces generated by magnetic fields. Nearly all types of DC motors have an internal mechanism (electromechanical or electronic) to periodically change the current direction in a part of the motor.

4.1.5 Two-Channel Relay Switch:

The Arduino Relay Module enables a variety of microcontrollers such as Arduino, AVR, PIC, ARM with digital outputs to control larger loads, and devices such as AC or DC motors, solenoids, solenoids, and incandescent bulbs. This module can be integrated with 2 relays that can control 2 relays. The relay shield uses a high quality QIANJI JQC-3F relay rated at 7A / 240V AC, 10A / 125V AC and 10A / 28V DC. The relay output is indicated individually by a light emitting diode. Figure 15: ESC 4 x 30 amps.

4.2 Software Specification

Sr. No	Software	Version	Use case
1.	VNC Viewer		For accessing the Raspberry PI and PI Camera module.
2.	OpenCV		For face detection and recognition with the image stored in database.
3.	Arduino IDE	v1.8.8	To flash code in NodeMCU in LUA programming language
4.	Python IDLE	2.7	The primary scripting language used for backend
5.	Connectify		To create a personal hotspot connection.
6.	Apache Server		Used to get live streaming from Pi Camera.

4.3 Block Diagram

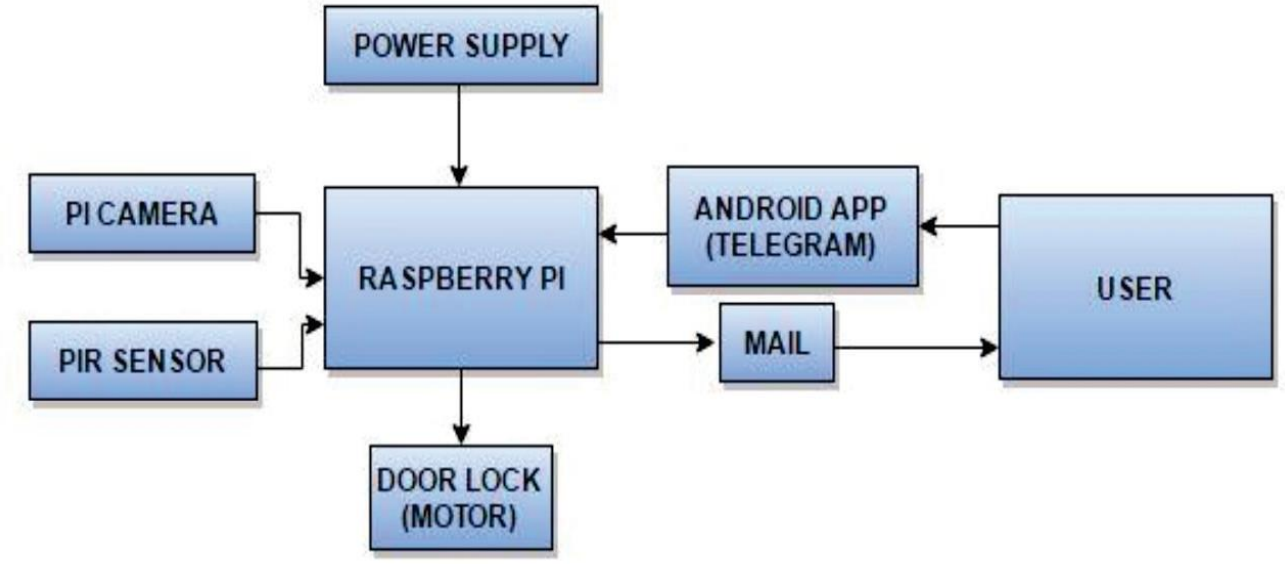


Figure 1: Block diagram of Door Monitoring System

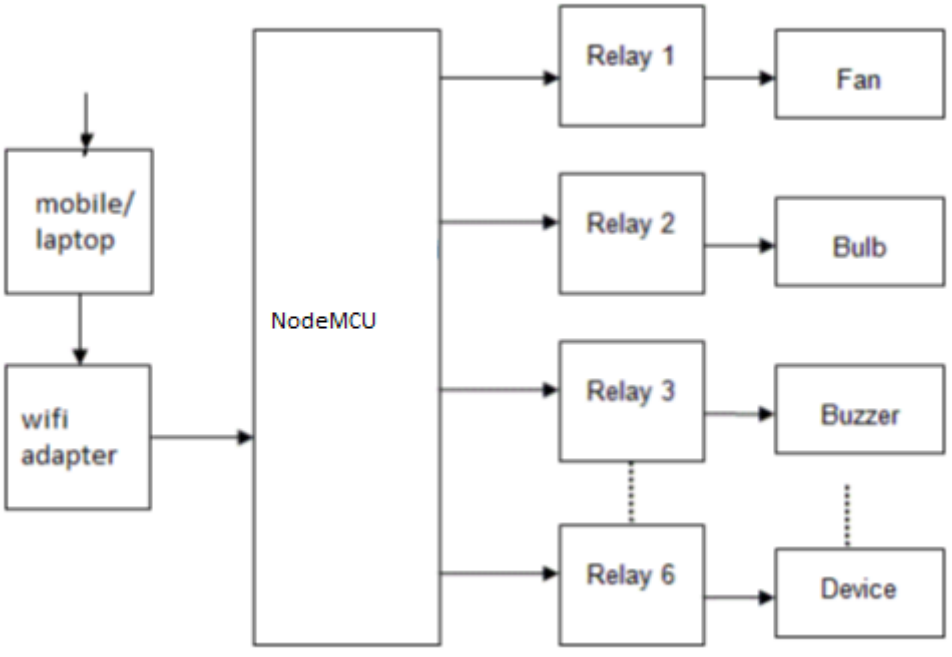


Figure 2: Block diagram of Smart Lighting system

5. DESIGN PLAN

5.1 Engineering Design

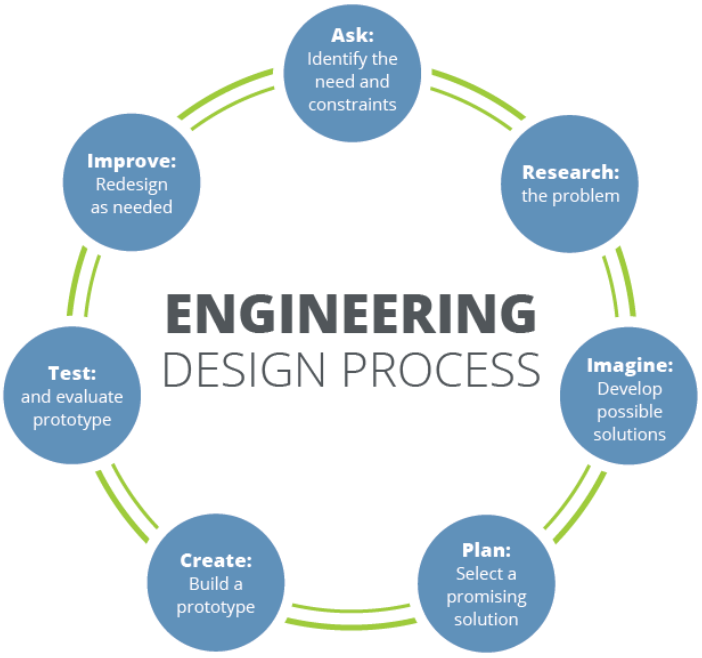


Figure 3: Engineering Design

The basic steps taken into account for the engineering design are as follows:

1. Problem Definition: The aim of the project scope was to find the relevant and efficient use of technology.
2. Background Research: Exploration of various IoT and trends in technology that can be implemented such as Image Processing, Raspberry Pi and wearable devices.
3. Requirement Specification: A appropriate budgeted project which involves the latest trends and brings new aspects of technology utility in the light.
4. Solutions/Proposed Solution: A novel security, lighting system embedded together as one with automation.
5. Selected solution: Designing the complete system using Web of Things.
6. Procedure/ Development Work: Development of modules, assembly of hardware for the system.

7. Prototype: Constructing the prototype by integrating the modules and hardware.
8. Testing: Unit testing of the individual modules, and integration testing of the system.
9. Make Observations: Minor modifications and additions based on the review feedback by the panel.
10. Redesign: Introduction of the changes suggested by the panel in the reviews for a better and improved prototype.

5.2 Design Approach

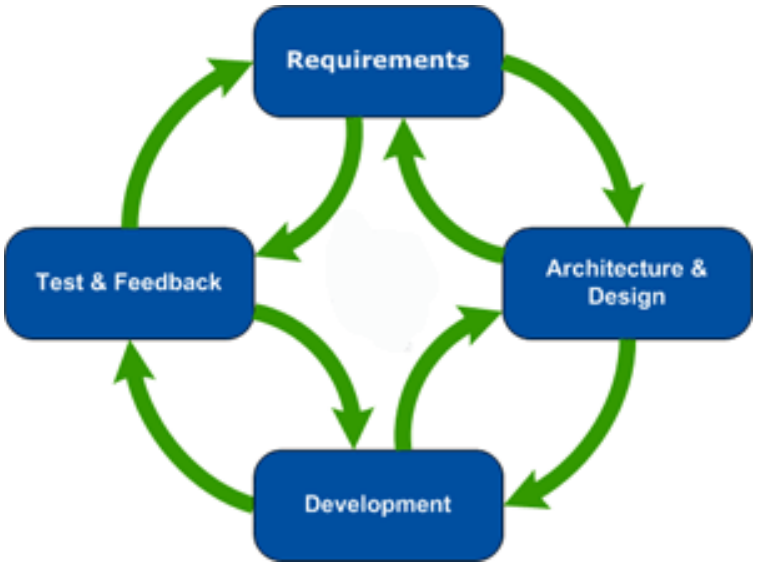


Figure 4: Software development process used- Agile Development

The SDLC process used was the Agile process. The Agile process was chosen to continually update the workflow and create the best possible design based on the advice of guide and the panel members. While the general idea and problem definition were defined, more complex details had to be defined based on the information provided by the physicians and mutual discussions. Despite the time and space restrictions, the project members were committed and ready to work.

Benefits of using the Agile process for project:

1. To incorporate all the suggestions given by guide and panel members.
2. Make major design changes according to the suggestion of the subject matter experts.

3. Complete the project on time and also achieve the stretch goals of project.
4. Integrate the different modules developed under a single umbrella providing a comprehensive and scalable solution.
5. The rapid development of prototypes for testing purpose could be completed within the timelines and could be tested and verified.
6. The flexibility provided to work.

6. PROJECT MODULES

6.1. Internet of Things (IOT):

The Internet of Things (IoT) is an ecosystem of connected physical objects that can be accessed via the Internet. The "thing" in the IoT may be a person with a heart monitor or an automobile with built-in sensors, ie objects that have been assigned an IP address and that have the ability to collect and transmit data in a network without any help or manual procedure.

Including all the component (things) are interconnected with each other in some or the other way like DC Motor is getting signal from both Raspberry Pi and NodeMCU RPI is used to send signal after processing the image and recognizing it on the other hand NodeMCU has a remote access over internet through an app and both can communicate with each other. Also 2 LEDs(things) embedded in the system can turned off and on using mobile over internet.

Advantages of implementation:

1. Effective use of resources

2. Reduction of human effort

3. Reduces costs and increases productivity

4. Real-time marketing

5. Decision analysis

6. Best customer experience

7. High quality data

6.2 Image Processing

Image processing is a method of converting an image into a digital format and performing certain operations to obtain an enhanced image or to extract useful information therefrom. This is a type of signal distribution in which the input is an image, such as a video image or a photograph, and the output can be an image or features associated with that image. Typically, the image processing system is to process the images as two-dimensional signals while applying already defined signal processing techniques.

It is one of today's fast growing technologies with applications in different areas of a business.

Image processing is also a critical area of research in engineering and computer science.

The image processing is divided into 5 groups:

1. Visualization - Observe invisible objects.

2. Sharpen and restore images - create a better image.

3. Image Recovery - Look for the image that interests

4. Pattern Measurement - Measures different objects in a picture.

5. Image Recognition - Distinguish objects in an image.

Image types:

•Binary image - The binary image, as the name implies, contains only two pixel elements: 0 and 1, where 0 is black and 1 is white. This picture is also called monochrome.

• Black and White Image - The image, which consists only of black and white colors, is called a black and white image.

• 8-bit color format - This is the most popular picture format. It contains 256 shades and is commonly referred to as a grayscale image. In this format, 0 stands for black and 255 white and 127 gray.

• 16-bit color format - This is a color picture format. It contains 65,536 different colors. It is also called High Color Format. In this format, the color distribution is not the same.

Image as a Matrix-

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & f(0,2) & \dots & f(0,N-1) \\ f(1,0) & f(1,1) & f(1,2) & \dots & f(1,N-1) \\ . & . & . & & . \\ . & . & . & & . \\ . & . & . & & . \\ f(M-1,0) & f(M-1,1) & f(M-1,2) & \dots & f(M-1,N-1) \end{bmatrix}$$

Figure 5: Matrix Distribution of an Image

Images are represented in rows and columns the following syntax in which images are represented:

The right side of this equation is digital image by definition. Every element of this matrix is called image element, picture element, or pixel.

6.3 Web of Things

Connecting every object to the Internet and assigning it an IP address is only the first step towards the Internet of Things. Things could easily exchange data with each other, but they do not necessarily understand what that data means. This is what Web protocols such as HTTP bring to the Internet: a universal way of describing images, text, and other multimedia so that machines can "understand" each. The Web of Things - or WoT - is just the next step in this evolution: using and customizing web logs to connect everything that exists in the physical world and make it available on the World Wide Web.

The Web of Things is a high-level application protocol designed to maximize Internet of Things interoperability. We hope that this brief introduction will give an idea of the potential.
Web technologies are very popular and provide the flexibility and functionality required by the majority of future IoT applications, including real-time discovery, security, and messaging.

6.4 Module Description

The project comprises of two modules in the project one is door monitoring on RPI and smart lighting system over NodeMCU but the curious thing here is both have a common platform to access the things connected with them. DC motor is communicating with both RPI after processing the image and NodeMCU through the android application. Similarly, more appliances can be controlled using the multiple channel relay switch. Both the microcontrollers have been powered using a 5v transformer to reduce the delay in processing. Based on the pins of microcontroller connected appliances can be controlled form anywhere around the world if both user’s device and both the microcontrollers are connected over the internet. The application displays the intensity of light received in LDR sensor also for how long devices are in active state. This refers to a complete system carried over an app this high level abstraction refers to web of things. It has heterogeneous protocols running at a time like HTTP, SMTP, etc.

7. CODES AND STANDARDS

7.1 IEEE 802.11 WIFI

This is true for wireless LANs and provides 1 or 2 Mbps transmission in the 2.4 GHz band using frequency hopped spread spectrum (FHSS) or direct sequence spread spectrum (DSSS). All 802.11 specifications use the Ethernet protocol and CSMA / CA (CSMA / CA) for carrier-dependent contra-access multiple access. The initial modulation used in the 802.11 standard was Phase Shift Keying (PSK). However, some of the newest specifications use other schemes, such as Complementary Encoding Coding (CCK). New modulation methods enable higher transmission rates and a lower susceptibility to interference.

7.2 USB

USB, or Universal Serial Bus (USB), is an industry standard developed for the design of cables, connectors, and protocols for connecting, communicating, and feeding PCs and their peripherals.

The USB is designed to standardize the connection of computer peripherals to personal computers, both for communication and power. It has largely replaced various previous interfaces, such as serial ports and parallel interfaces, as well as separate portable device chargers - and has become commonplace on a variety of devices.

7.3 SMTP

Email is becoming one of the most valuable Internet services today. Most Internet systems use the SMTP protocol to relay mail from one user to another. SMTP is a push protocol that is used to send e-mails, while the Post Office Protocol (POP) or Internet Message Access Protocol (IMAP) retrieves these messages on the recipient side.

SMTP is part of the application layer of the TCP / IP protocol. With a process called "Save and Forward", SMTP transmits your emails to and over networks. It works closely with the so-called Mail Transfer Agent (MTA) to email your communications to the right computer and inbox.

SMTP specifies and specifies how your e-mail is transferred from the MTA of your computer to an MTA on another computer and even on multiple computers. The above Save & Transfer feature allows the message to be progressively moved from your computer to the destination. At each step, the Simple Mail Transfer Protocol does its work. Fortunately, everything happens behind the scenes.

7.4 HTTP

HTTP (HyperText Transfer Protocol) is an application-level protocol used primarily on the Web. HTTP uses a client-server model in which the Web browser is the client and communicates with the Web server hosting the Web site.
The browser uses HTTP transmitted over TCP / IP to communicate with the server and retrieve the user's web content. HTTP is a widely used protocol and was quickly adopted on the Internet because of its simplicity. It is a stateless and connectionless protocol.

7.5 IPv4

A protocol that defines these addresses is called an Internet Protocol. These addresses are referred to as the Internet Protocol (IP) addresses of the sender and the recipient.

Other IP addresses have two versions:
Internet Protocol Version 4 (Ipv4): 32 binary digits (0 or 1) are used to represent the address.

8. CONSTRAINTS AND ALTERNATIVES

8.1 Design Constraints

The system was built considering the following design limitations:

- The system uses a 220V AC power supply.
- The best technologies should be used to store information.
- The availability of the internet connection.
- Differences in the needs of different users.
- Elderly and disabled persons.
- Optimal use of existing resources as well as minimal resources and additional costs for the affected users.

8.2 Components Constraints

The NodeMCU cab be used instead of Arduino because it uses integrated WIFI. It can be easily connected NodeMCU to the Internet, compared to connecting UNO to the Internet. Best is the compatibility of NodeMCU with Arduino can do all sorts of tasks with the UNO. It can also be programmed NodeMCU with the Arduino IDE directly in the language 'C'

Comparison of the NodeMCU and Arduino.

This comes with 128 KB of RAM and UNO, there are only 2 KB of RAM, which NodeMCU has more space.
It has 4 MB ROM (Flash) and UNO only 32 KB. NodeMCU can store more code than UNO.
It comes with a micro USB port and UNO comes with a USB Type B port. A micro USB cable is available compared to a USB type B.
The development board is smaller than that of Arduino UNO.
The prices for NodeMCU and UNO are almost identical, so we chose NodeMCU.

8.3 Budget Constraints

Due to the budget constraints, we decided to proceed with the low-cost pi camera of 600 Rs rather going for a high definition home surveillance camera of 5000Rs because the aim is make a real time prototype. After executing over it we can analogically say that it fits on almost all type of cameras.

9. PROJECT DEMONSTRATION

9.1 Project Views

A few screen captures of the project to provide a broader understanding of the project and a holistic view of the functionalities have been given below.

Screen Captures:

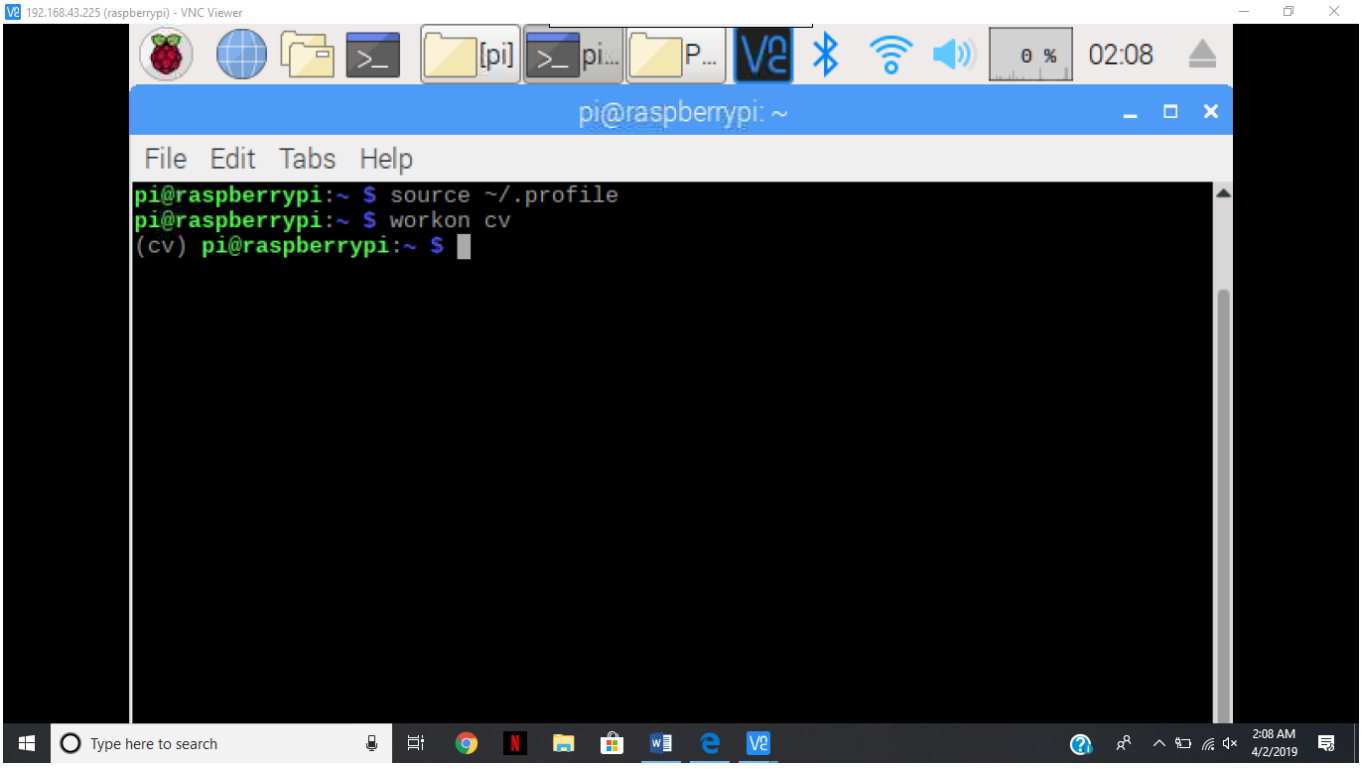


Figure 6: Loading Profile and Starting OpenCV

Figure 6 shows that profile has been loaded and OpenCV has been started to use its library and features.

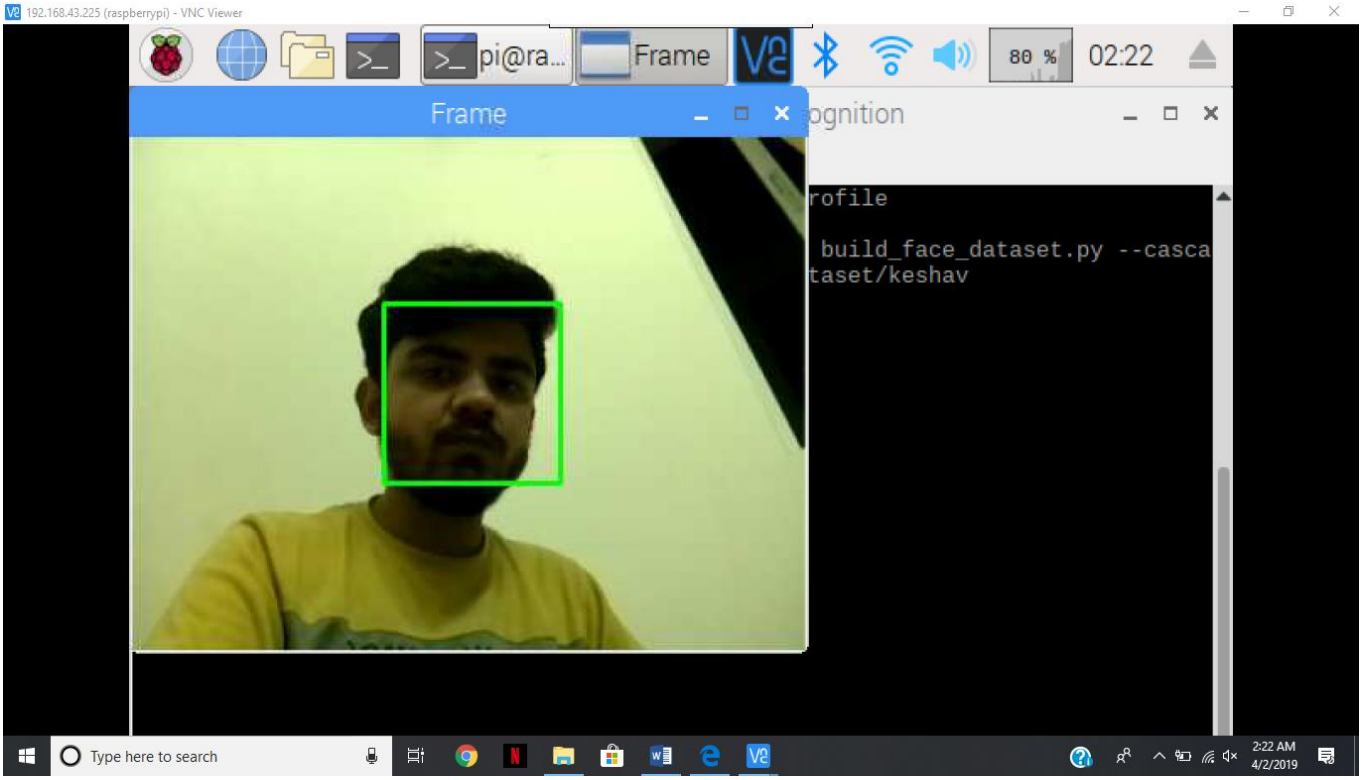


Figure 7: Capturing the face to store in database

Figure 7 shows the first phase to capture known faces for face recognition taking the photo to process it further.

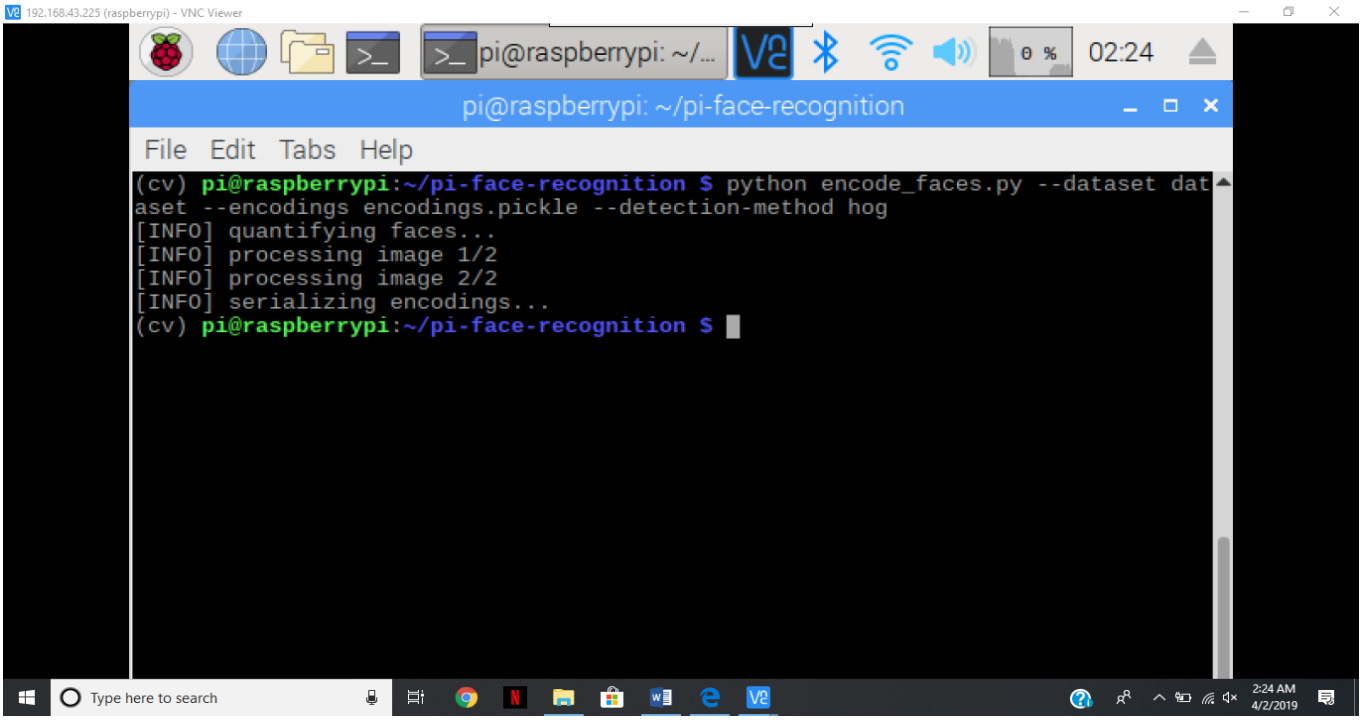


Figure 8: Encoding the images into a matrix

Figure 8 shows the second phase in which encoding of images is been done so as to convert each and every image into the matrices.

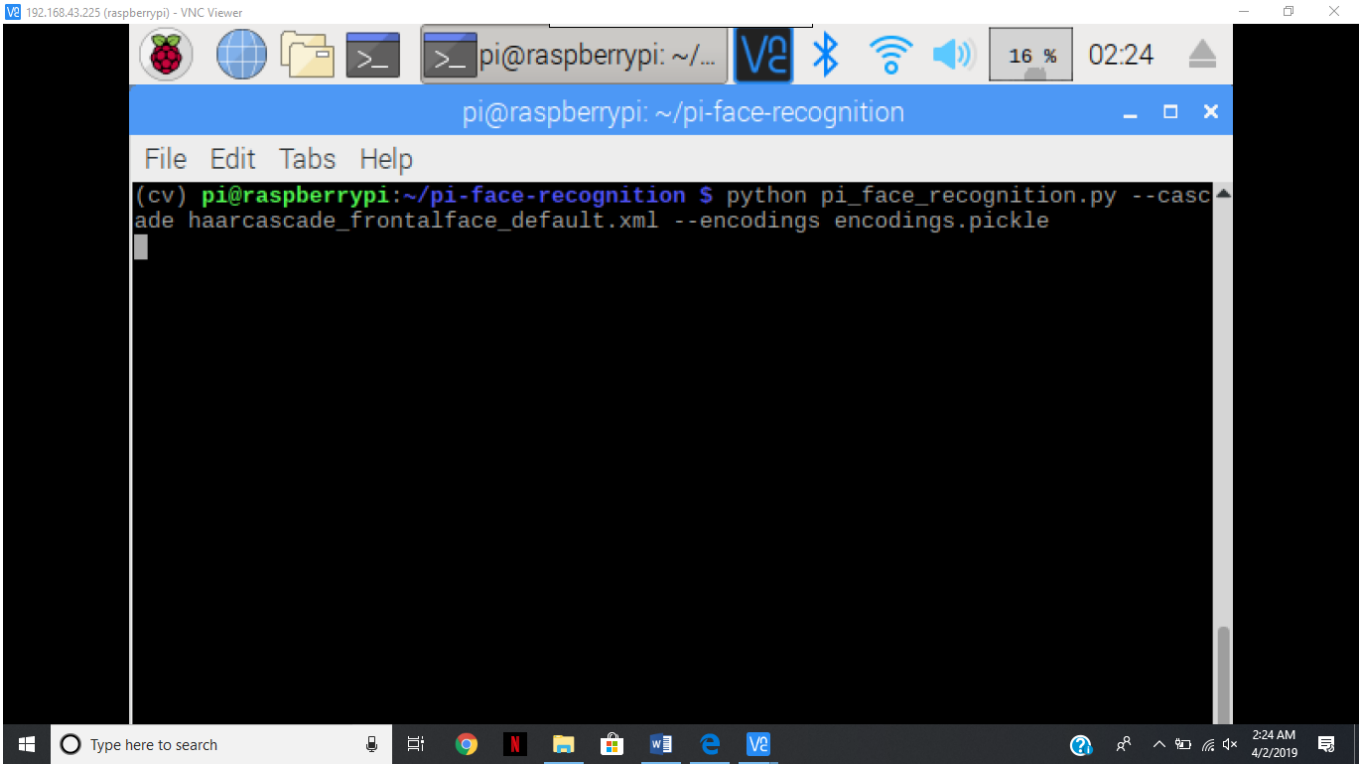


Figure 9: Face recognition for door monitoring

Figure 9 is third and the last phase to recognize whether the captured person is known or unknown.

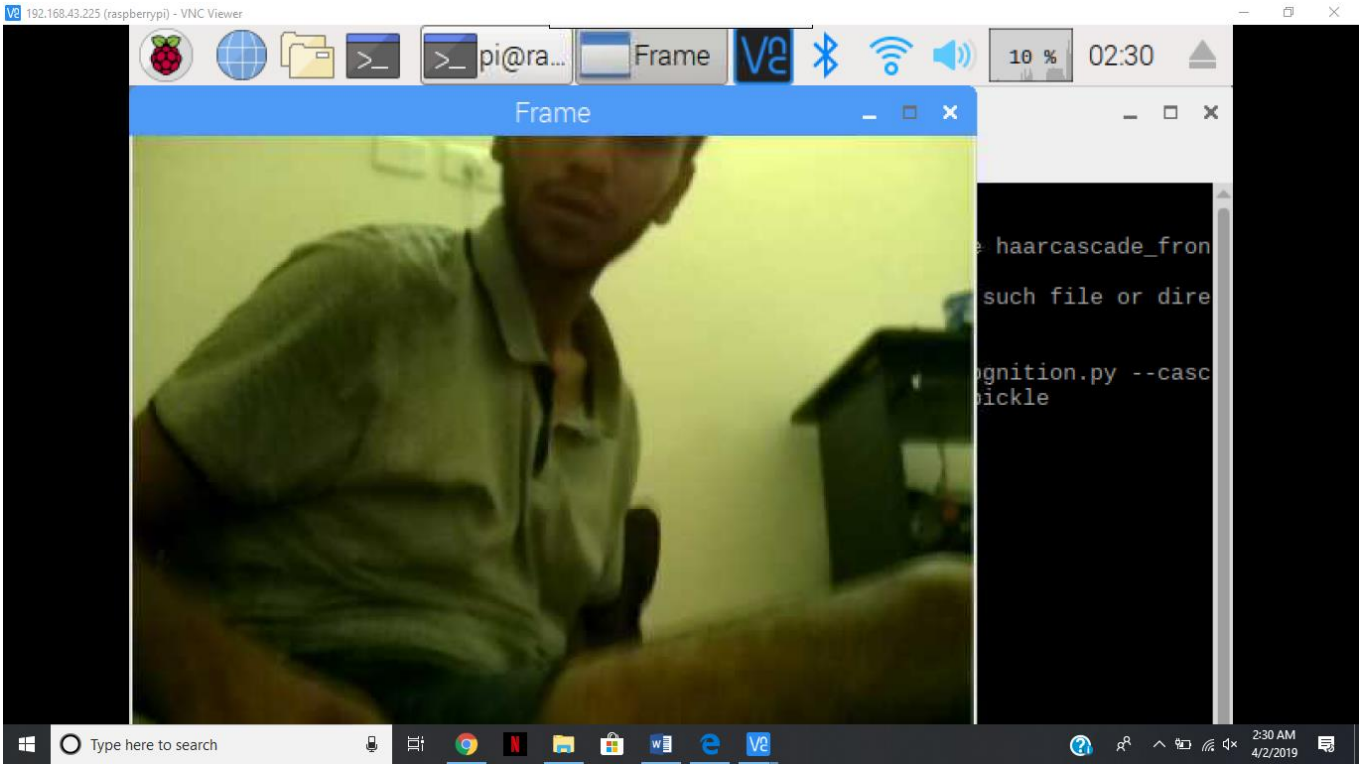


Figure 10: Unknown Face Detected

After executing the python file for face recognition the streaming starts and detects an unknown face in the camera and responds to it.

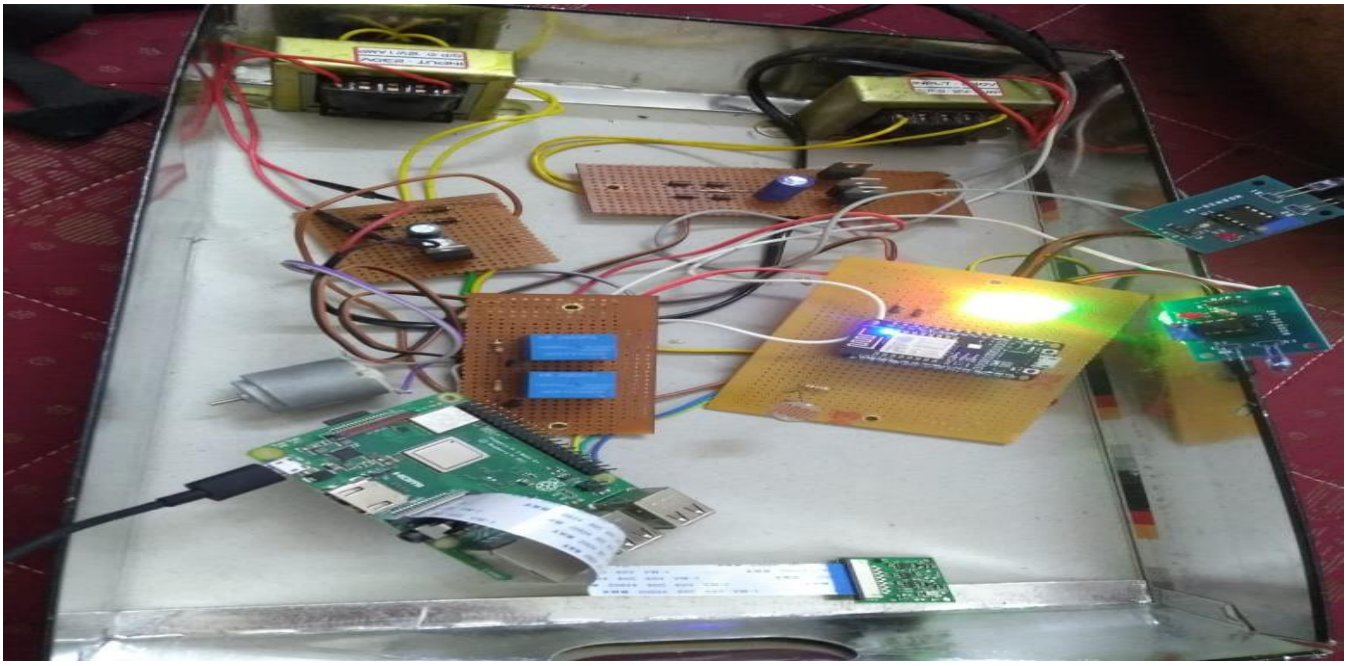


Figure 11: Both LEDs together

In Figure 11 both the IR sensors have been triggered due to availability of obstacle in the range of both sensors.

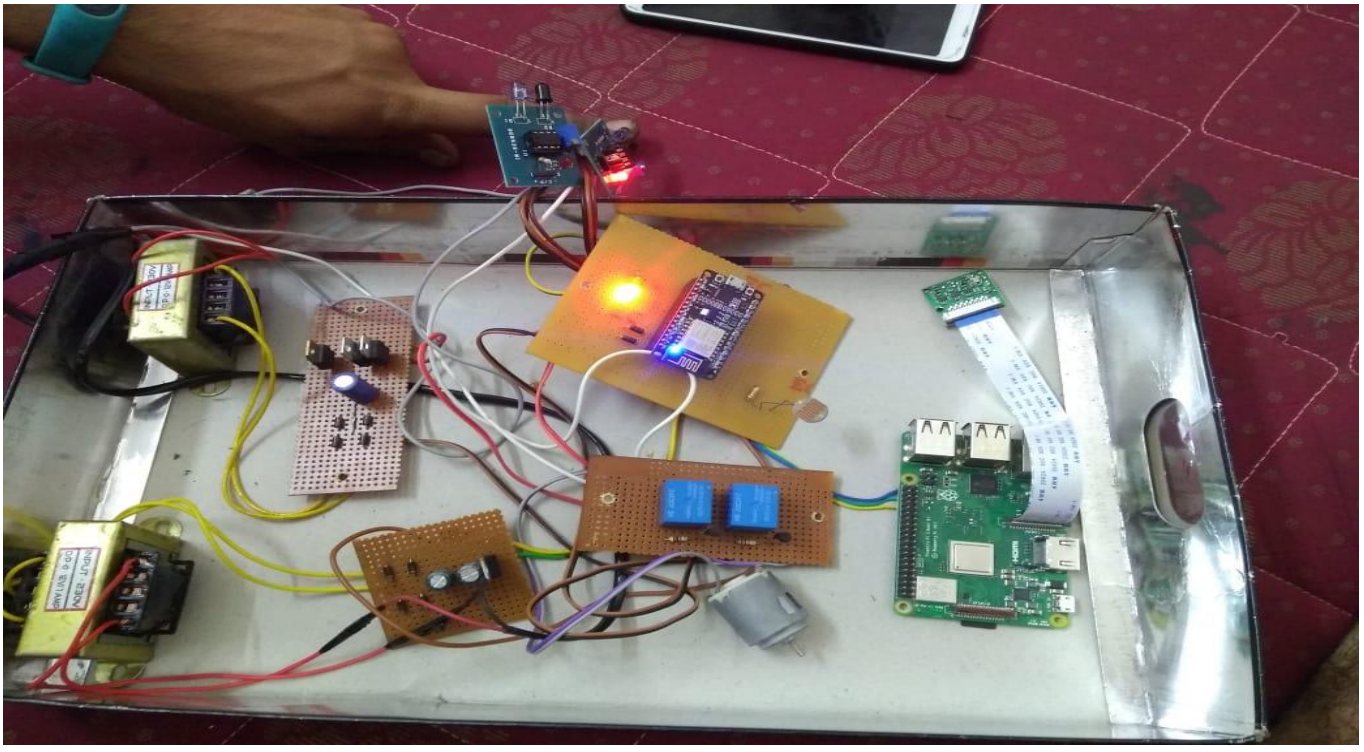


Figure 12: Smart Lighting System: Red LED Glowing

Figure 12 shows that only one light i.e Red is glowing due to hand of a person near to IR sensor and other is in off state.

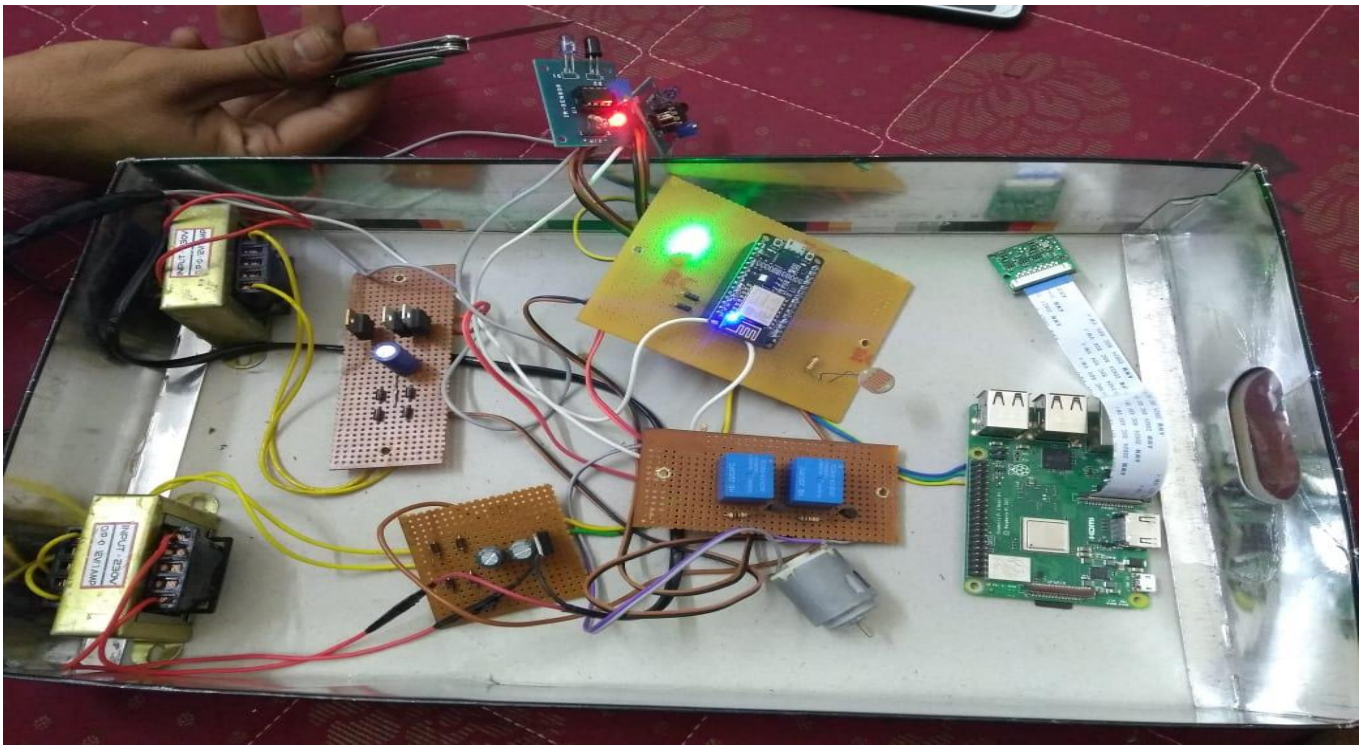


Figure 13: Smart Lighting System: Green LED Glowing

Figure 13 shows that only one light, that is, Green glows because of the hand of a person close to the infrared sensor and the other is in the deactivated state.

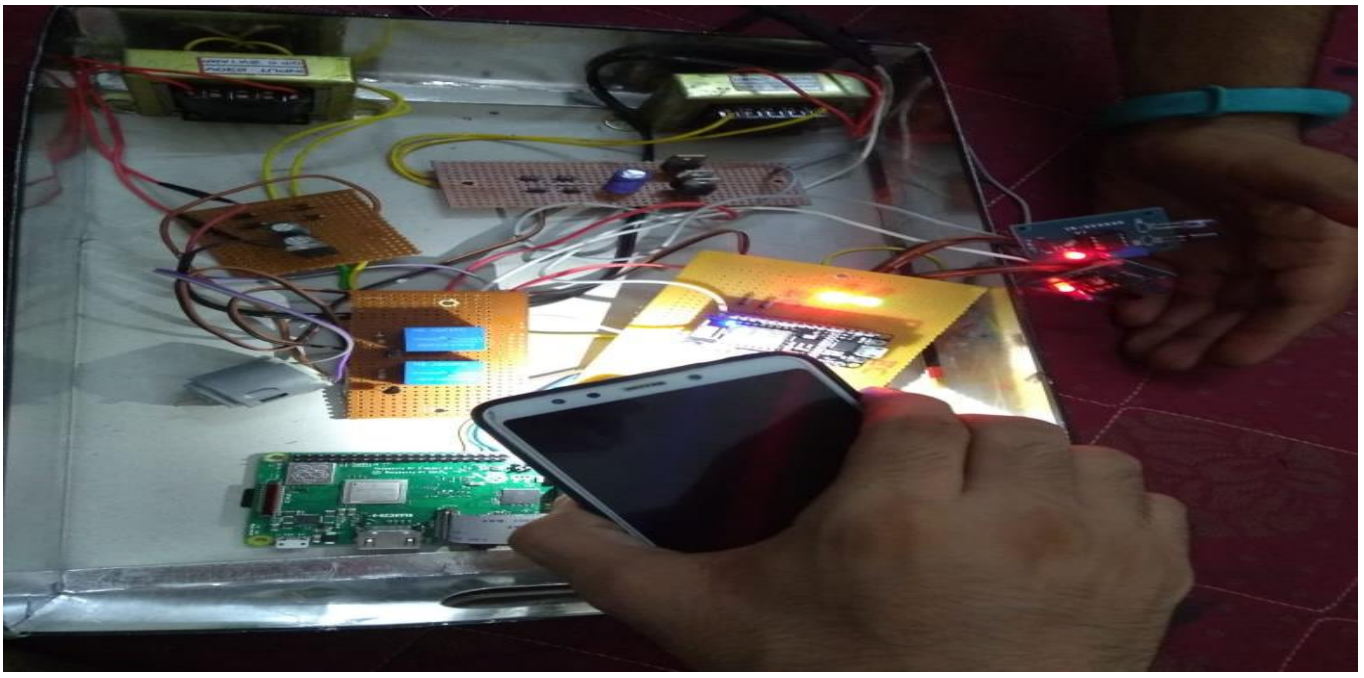


Figure 14: Light Intensity varying with LDR Sensor

In Figure 14 light intensity is changing as per change in the intensity of light here it is simulated using a mobile phone flash.

Figure 15: Complete System

Figure 15 is the complete setup for door monitoring and smart lighting and Raspberry Pi being used in the laptop.

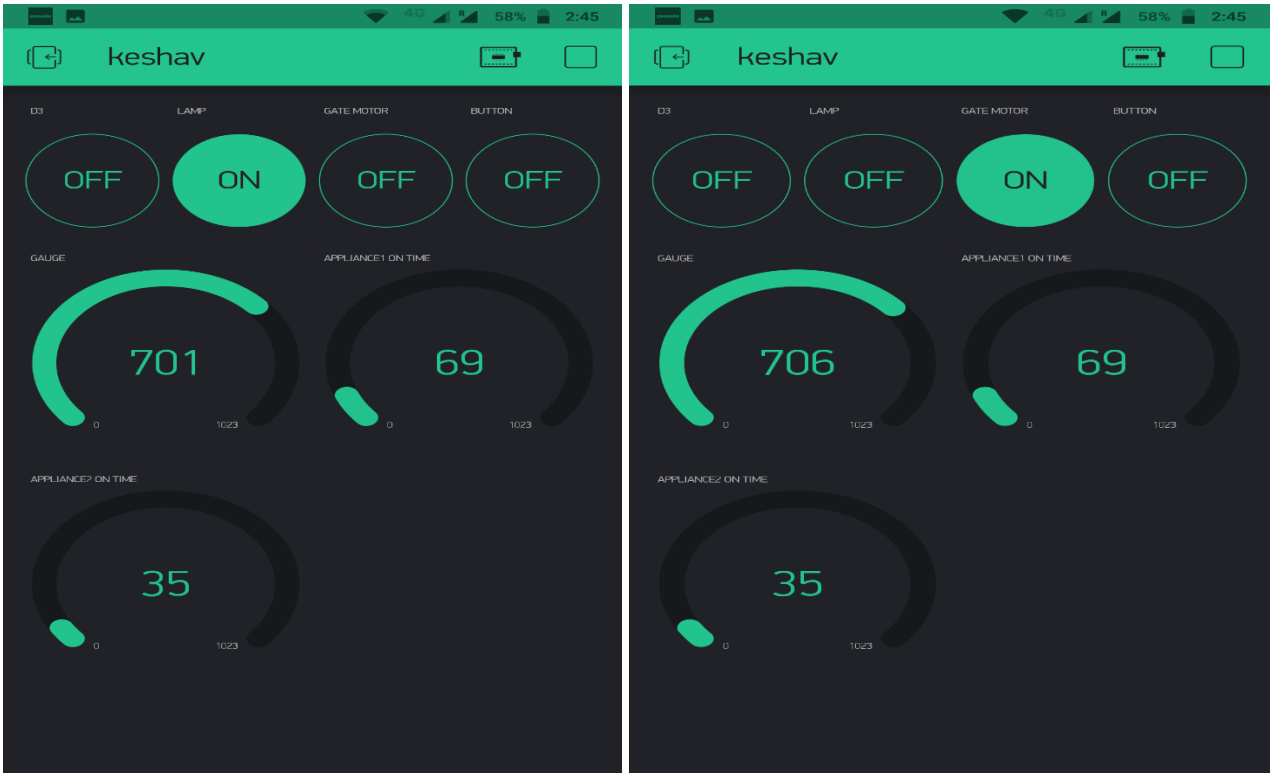


Figure 16: Android App to control devices

Figure 16 is an Android app from which lights, LEDs, door can be swithed ON/OFF anytime and anywhere if the system is connected to internet.

10. COST AND MARKET ANALYSIS

10.1 Market Analysis

Presently, the commercial applications of home automation are a bit costly. Also, there is less number of WOT projects available in the market. The project is highly scalable as well. It has been enriched with many different features such as smart lighting control and automation, door safety and monitoring. The project is designed in such a way that things can work smoothly even if used parallel sequence. The whole system can be migrated and integrated with cloud. We also provide an option to access lighting and security both remotely over the internet.

10.2 Cost Analysis

	Parts	**Cost**
1.	NodeMCU ESP8266	Rs 350.00
2.	Raspberry Pi	Rs 3000.00
3.	2-Channel Relay Switch	Rs 200.00
4.	2xTransformers	Rs 400.00
5.	DC Motor	Rs 50.00
6.	LEDs and Jumper Wires	Rs 50.00
7.	Pi Camera	Rs 600.00
	Total Parts	Rs 4,650.00
	Total Cost per Unit	**Rs 4,650.00**

11. CONCLUSION

11.1 How this project supports the present solution

In the current scenario people are focusing in making their building smart. Already ample of research exists on the same topic as given in references. Smart lighting is mostly implemented on Arduino whereas NodeMCU is more efficient and effective way of doing the same thing have used relays and transformers to give optimum supply of voltage to each appliance. As door monitoring system has been designed in a bit different way taking multiple real-life test cases.

It has some advantages over present solution like synchronization of two microcontrollers, action and reaction of appliances based on real life situations, controlling the devices connected to microcontroller using an android application such that both user and system has an access of internet to them.

However, the existing system exist as individual and has some limitations. If all taken together like cross platform synchronization, latency in communication however we have focused on minute details like mentioning the time a particular appliance was used. Also providing the remote access to whole system anywhere round the globe over internet. It is helpful for both domestic and commercial buildings to keep them safe from any kind of forgery and stealth action. Smart lighting helps in saving electricity as well as reduces the human effort to switching on and off the lights manually. This project can be helpful for elderly and disable people to operate the system using android application. After calculating the budget, it came out to be an effective project serving multiple purposes in reasonable cost of making.

11.2 Future implementations

In future in can be refurbished according to any other commercial or governmental organization and can be installed there. Talking about advancement in hardware powerful microcontroller can be used to add some more features in it like prediction of some hazard by taking constant reading and sending them on cloud as soon as some anomaly is going to happened it can send us a notification. The project is scalable enough to add some more features like showing the reading of temperature and humidity and same can be sent to an air-conditioner as and when there is a change in temperature to keep it constant.

Cloud data analytics can be done after collecting the ample of data about various things and same can be used to predict some of the things in advance based on previous observations. This a very keen step to bring some changes in society like some hazards can be stopped or avoided using such door monitoring system also it can help us intimating about some weird or unlikely to happens is happening outside the house by capturing the images from the camera and sending them to respective user.

It can also be powered with deep learning and artificial intelligence in the future to bring a lot of automation and advancements in the system.

12. REFERENCES

[1] Januzaj, Y., Luna, A., Ramaj, V. 2015 *Real time access control based on Facial Recognition.*

[2] Çarıkçı, M., , Özen, F. 2012 *A Face Recognition System Based on Eigen faces Method.*

[3] Lwin, H., Khaing, A., Tun, H. 2015.*Automic door access system using face recognition.*

[4] Senthikumar, G., Gopalkrishnan, K., Sathish Kumar, V.2014 *Embedded Image Capturing System Using Raspberry Pi System*

[5] Sowmiya, U., shafiq mansoor, J. 2015 *Raspberry pi based home door security through 3g dongle.*

[6] Kartik J. Srimadhavan V. 2013 *SMS Alert and Embedded Network Video Monitoring Terminal*

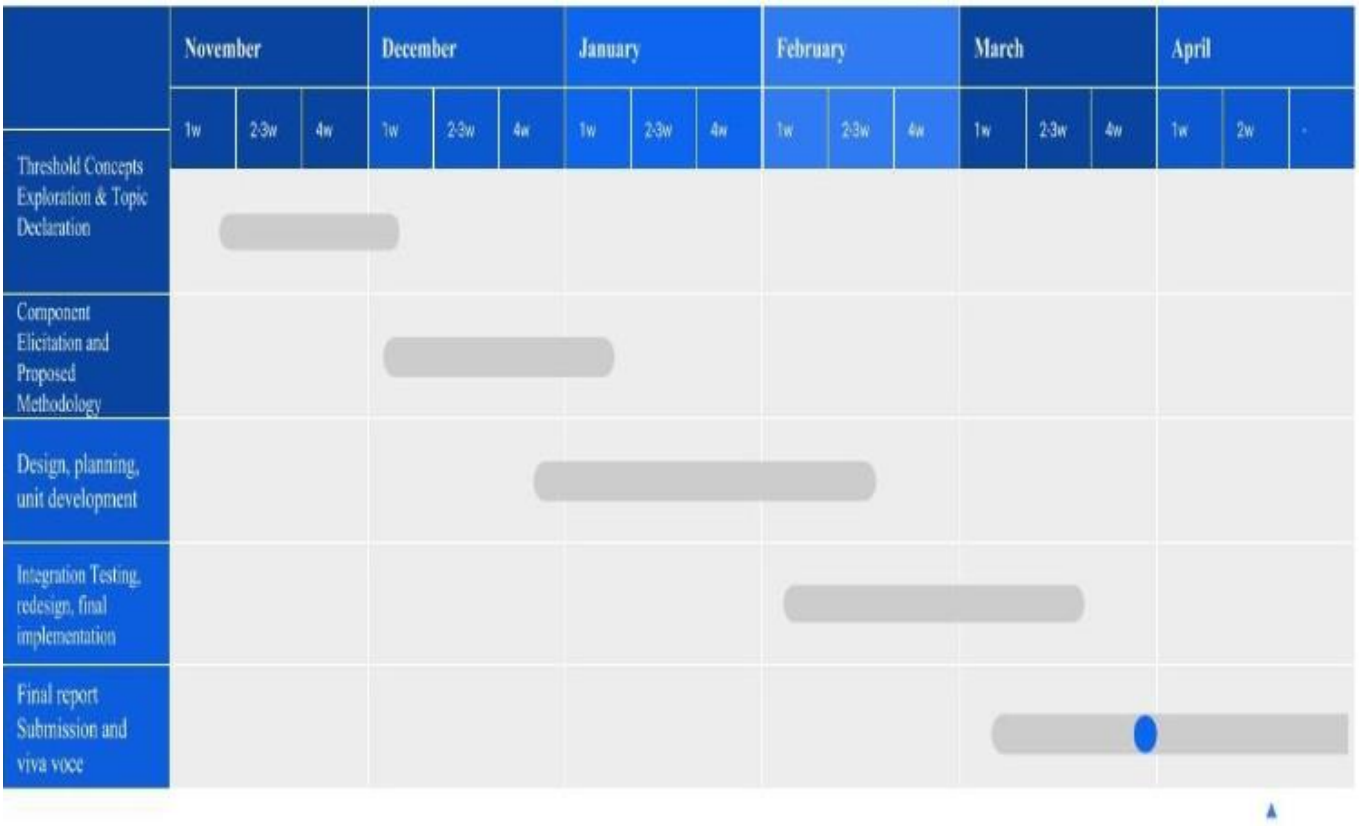
[7] Shrikanthan N. Tan F Karande A., *"Bluetooth Based Home Automation" , Microprocessors Micro systems, Elsevier Science, B.V. vol.26,no.6, pp281- 289,(2009)*

[8] Piyare R., Tazil M., *"Bluetooth based Home automation system using Cell Phone". Consumer Electronics, IEEE 15th International Symposium on, vol.45, no.3, pp.192-195(2011).*

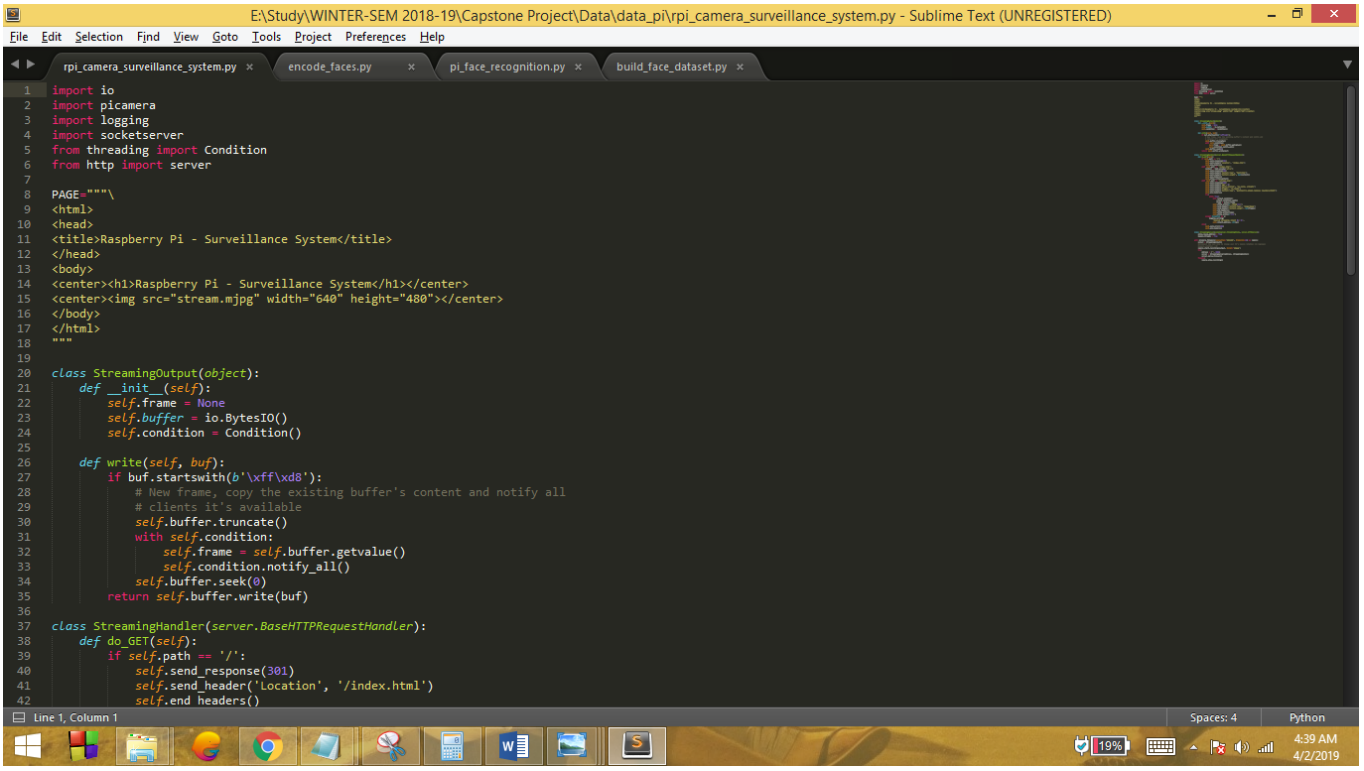
[9] EASAMBATTU, Thejaswini; REDDY, P. Ajay Kumar; RAMAIAH, G.N. Kodanda. *Controlling home appliances through GSM modem and Internet. International journal of Electronics Engineering Research, [S.I.], p. 1-7, oct 2013.*

.

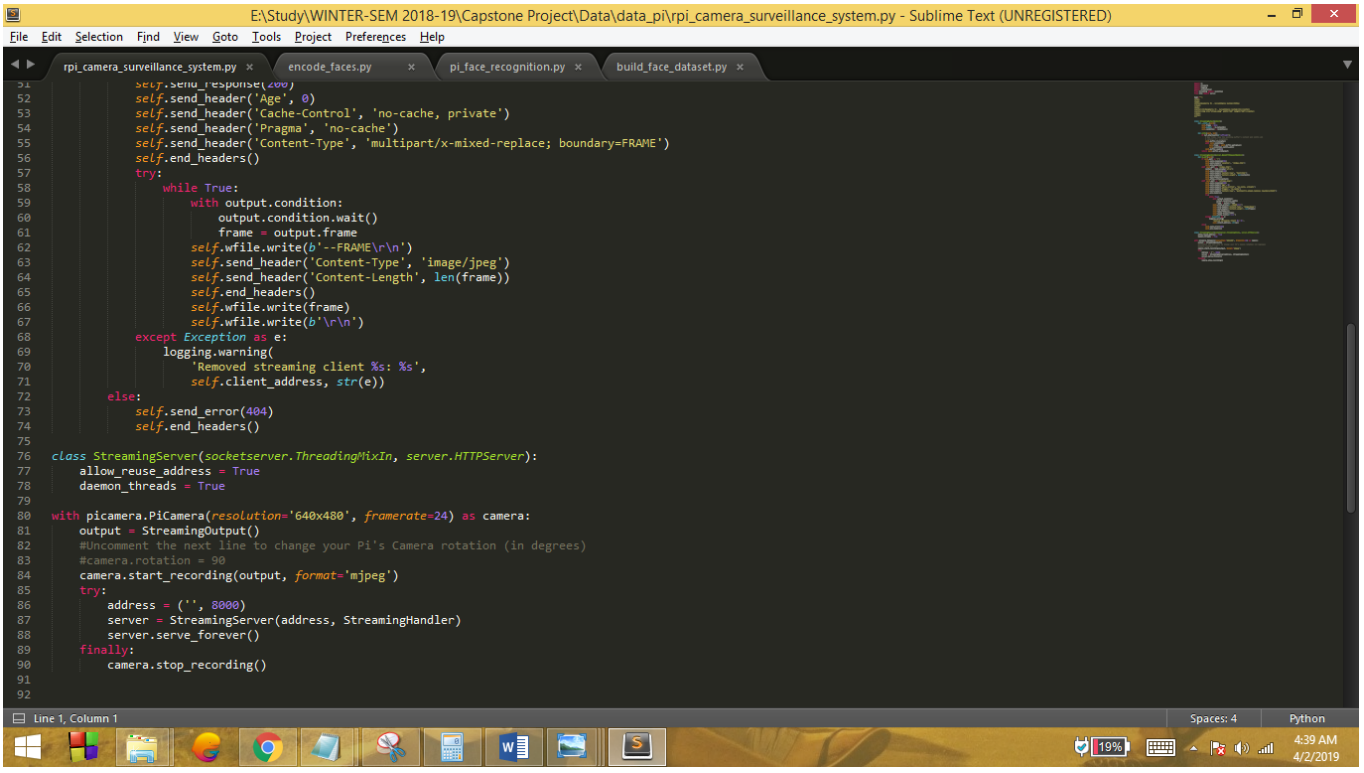
13. TIMELINE


November
December
January
February
March
April
1w
2-3w
4w
1w
2-3w
4w
1w
2-3w
4w
1w
2-3w
4w
1w
2-3w
4w
1w
2w
Threshold Concepts Exploration & Topic Declaration
Component Elicitation and Proposed Methodology
Design, planning, unit development
Integration Testing, redesign, final implementation
Final report Submission and viva voce

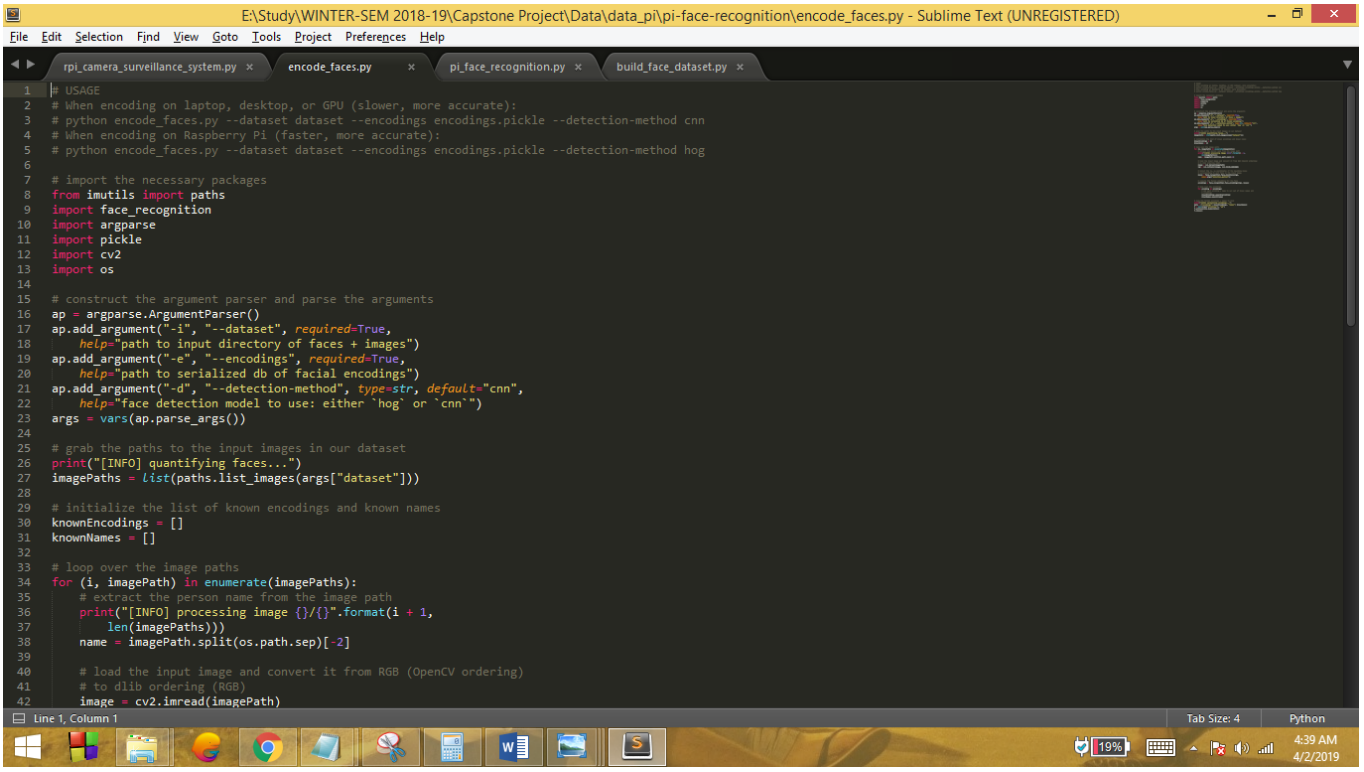
APPENDIX 1: CODE SNIPPETS

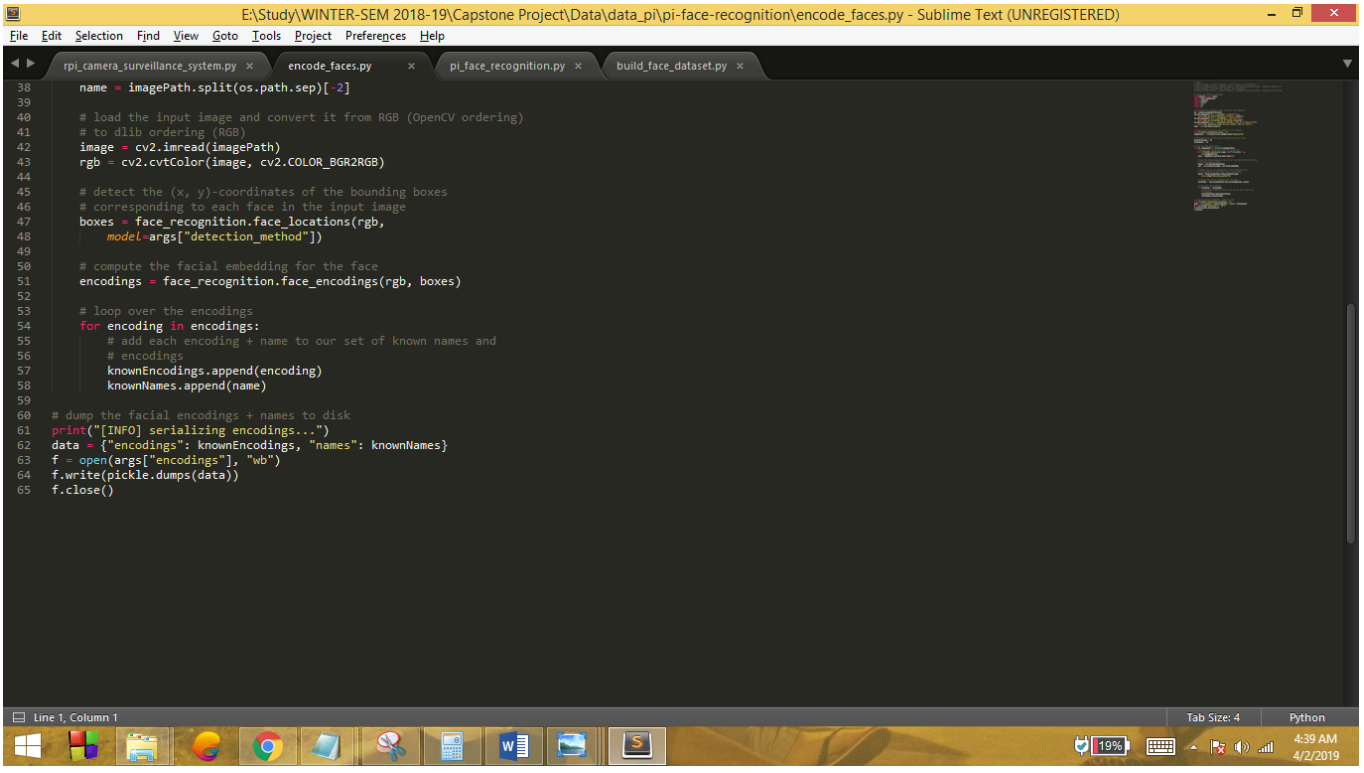


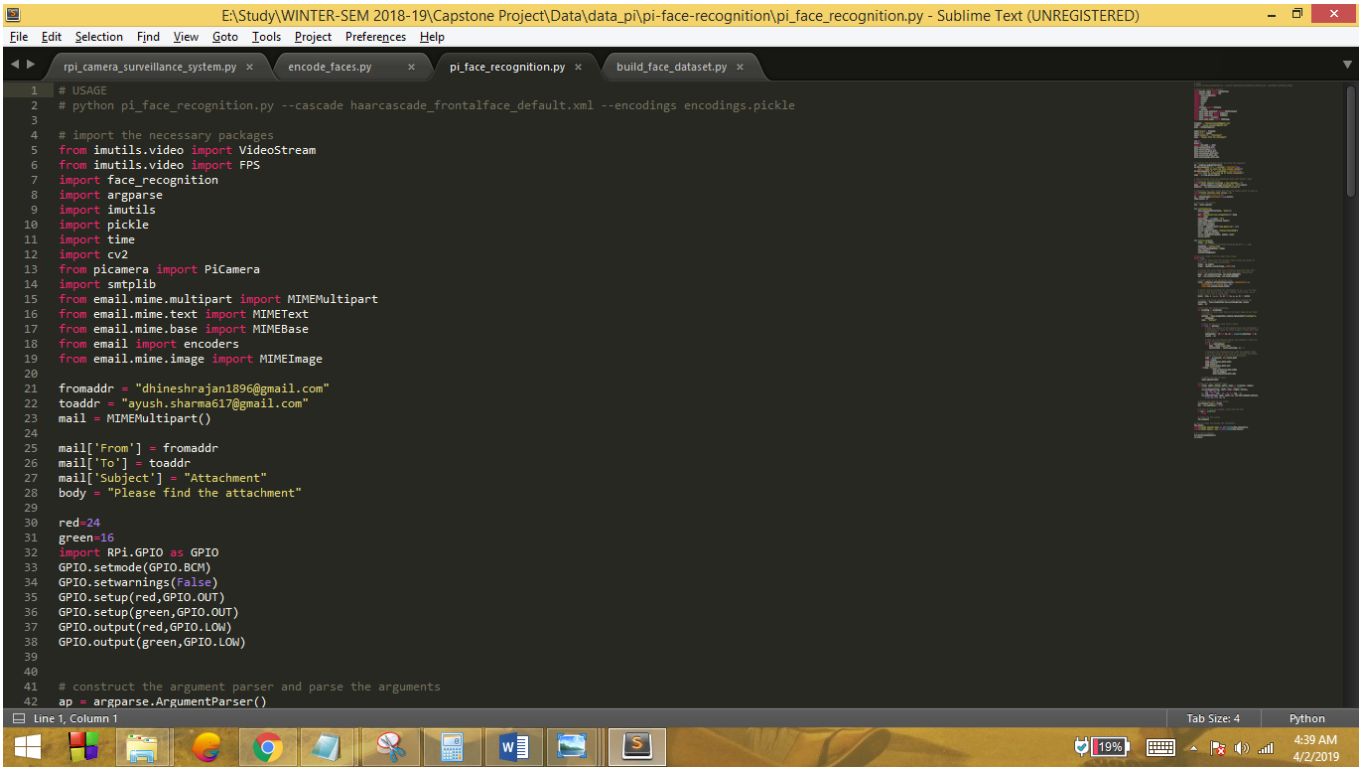
```
import io
import picamera
import logging
import socketserver
from threading import Condition
from http import server

PAGE="""\
<html>
<head>
<title>Raspberry Pi - Surveillance System</title>
</head>
<body>
<center><h1>Raspberry Pi - Surveillance System</h1></center>
<center><img src="stream.mjpg" width="640" height="480"></center>
</body>
</html>
"""

class StreamingOutput(object):
    def __init__(self):
        self.frame = None
        self.buffer = io.BytesIO()
        self.condition = Condition()

    def write(self, buf):
        if buf.startswith(b'\xff\xd8'):
            # New frame, copy the existing buffer's content and notify all
            # clients it's available
            self.buffer.truncate()
            with self.condition:
                self.frame = self.buffer.getvalue()
                self.condition.notify_all()
            self.buffer.seek(0)
        return self.buffer.write(buf)

class StreamingHandler(server.BaseHTTPRequestHandler):
    def do_GET(self):
        if self.path == '/':
            self.send_response(301)
            self.send_header('Location', '/index.html')
            self.end_headers()
```

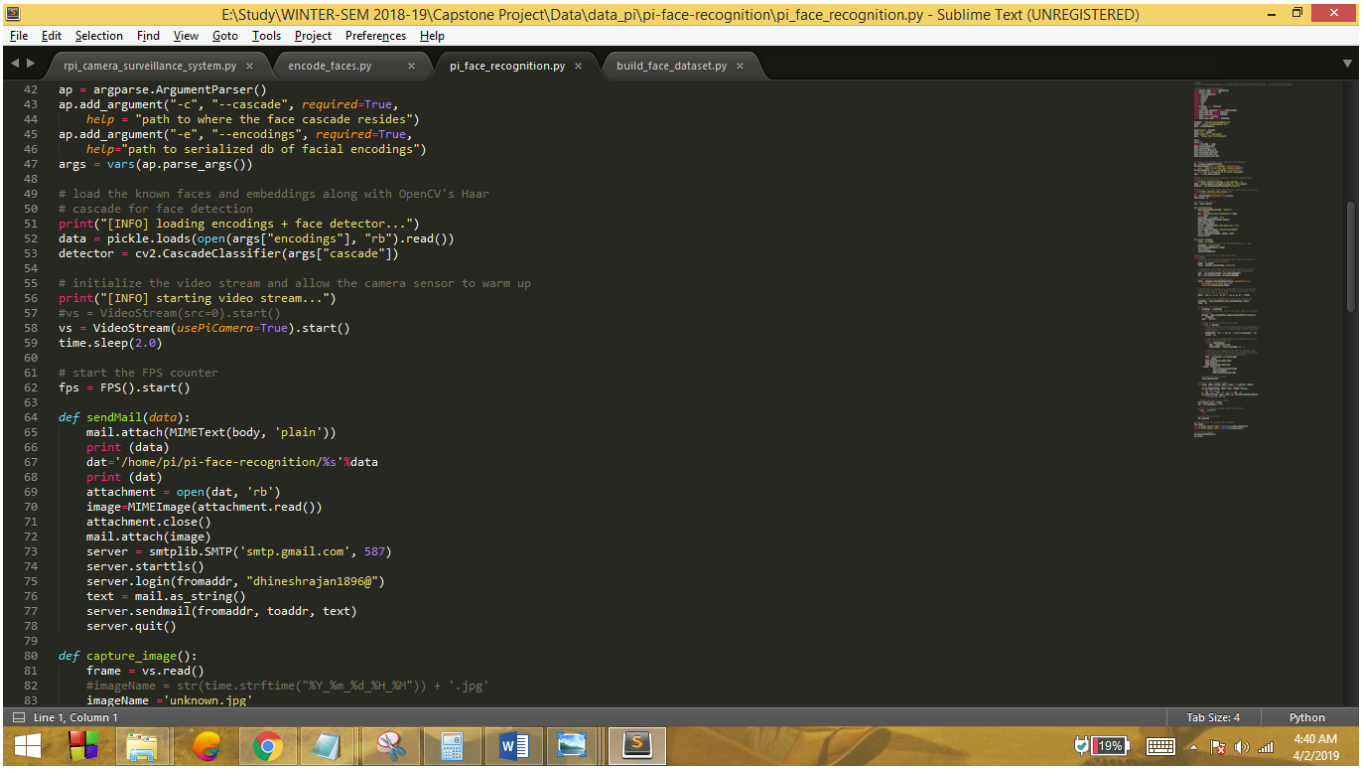


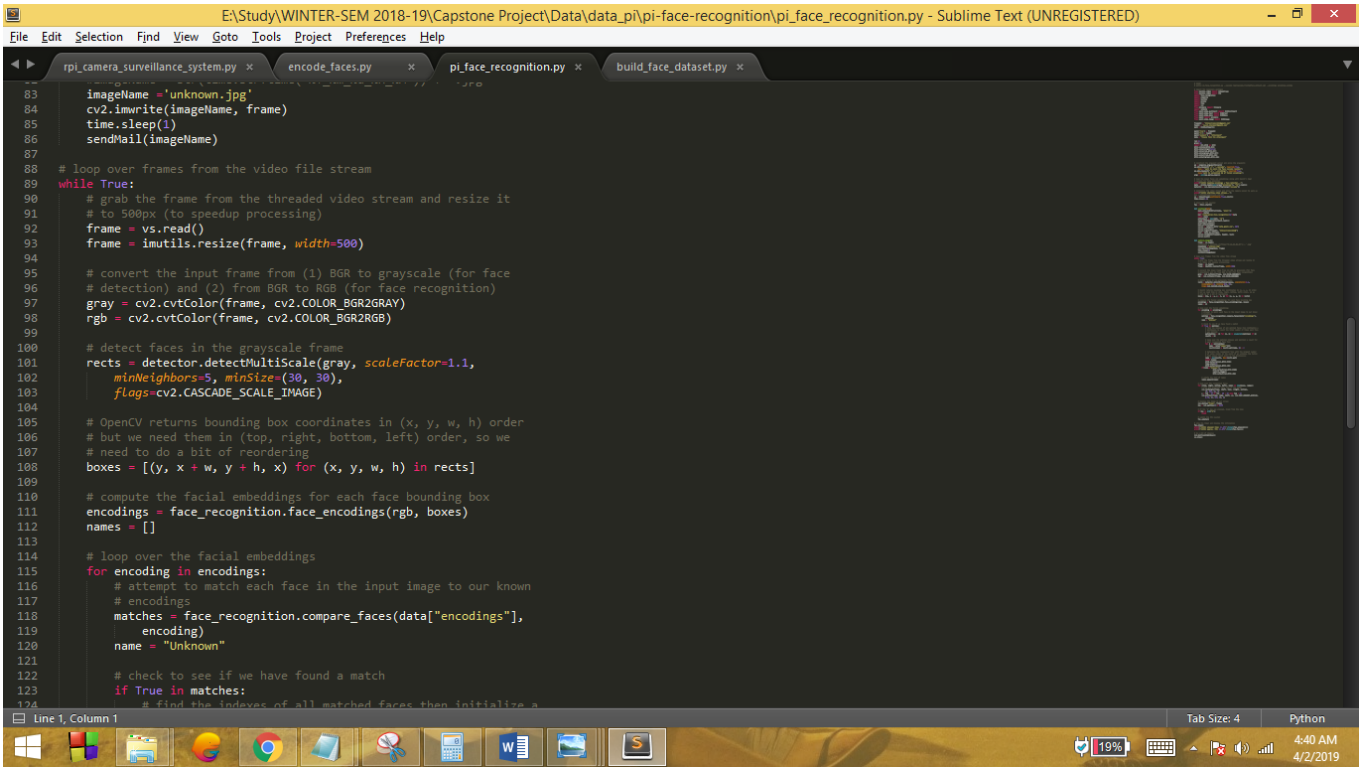
```
            self.send_response(200)
            self.send_header('Age', 0)
            self.send_header('Cache-Control', 'no-cache, private')
            self.send_header('Pragma', 'no-cache')
            self.send_header('Content-Type', 'multipart/x-mixed-replace; boundary=FRAME')
            self.end_headers()
            try:
                while True:
                    with output.condition:
                        output.condition.wait()
                        frame = output.frame
                    self.wfile.write(b'--FRAME\r\n')
                    self.send_header('Content-Type', 'image/jpeg')
                    self.send_header('Content-Length', len(frame))
                    self.end_headers()
                    self.wfile.write(frame)
                    self.wfile.write(b'\r\n')
            except Exception as e:
                logging.warning(
                    'Removed streaming client %s: %s',
                    self.client_address, str(e))
        else:
            self.send_error(404)
            self.end_headers()

class StreamingServer(socketserver.ThreadingMixIn, server.HTTPServer):
    allow_reuse_address = True
    daemon_threads = True

with picamera.PiCamera(resolution='640x480', framerate=24) as camera:
    output = StreamingOutput()
    #Uncomment the next line to change your Pi's Camera rotation (in degrees)
    #camera.rotation = 90
    camera.start_recording(output, format='mjpeg')
    try:
        address = ('', 8000)
        server = StreamingServer(address, StreamingHandler)
        server.serve_forever()
    finally:
        camera.stop_recording()
```

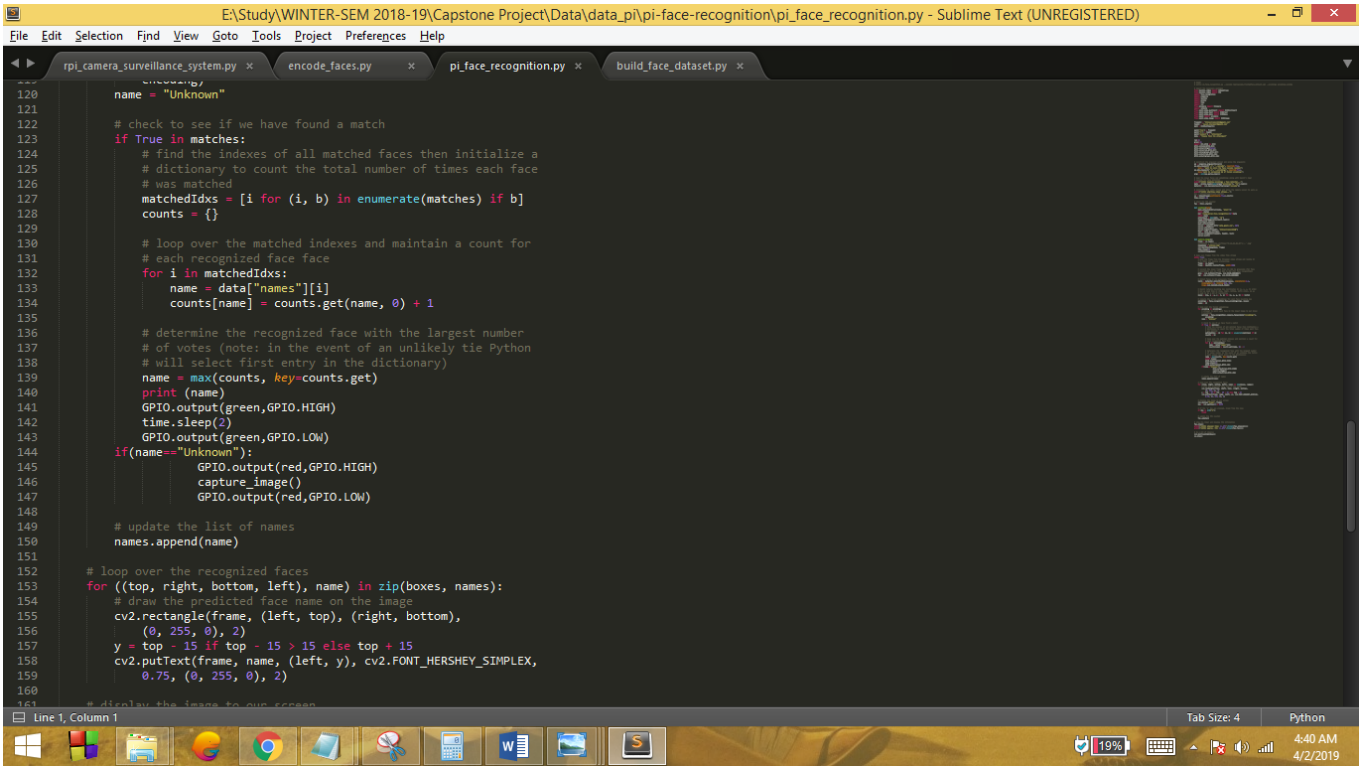

E:\Study\WINTER-SEM 2018-19\Capstone Project\Data\data_pi\pi-face-recognition\encode_faces.py - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
rpi_camera_surveillance_system.py
encode_faces.py
pi_face_recognition.py
build_face_dataset.py
# USAGE
# When encoding on laptop, desktop, or GPU (slower, more accurate):
# python encode_faces.py --dataset dataset --encodings encodings.pickle --detection-method cnn
# When encoding on Raspberry Pi (faster, more accurate):
# python encode_faces.py --dataset dataset --encodings encodings.pickle --detection-method hog
# import the necessary packages
from imutils import paths
import face_recognition
import argparse
import pickle
import cv2
import os
# construct the argument parser and parse the arguments
ap = argparse.ArgumentParser()
ap.add_argument("-i", "--dataset", required=True,
help="path to input directory of faces + images")
ap.add_argument("-e", "--encodings", required=True,
help="path to serialized db of facial encodings")
ap.add_argument("-d", "--detection-method", type=str, default="cnn",
help="face detection model to use: either `hog` or `cnn`")
args = vars(ap.parse_args())
# grab the paths to the input images in our dataset
print("[INFO] quantifying faces...")
imagePaths = list(paths.list_images(args["dataset"]))
# initialize the list of known encodings and known names
knownEncodings = []
knownNames = []
# loop over the image paths
for (i, imagePath) in enumerate(imagePaths):
# extract the person name from the image path
print("[INFO] processing image {}/{}".format(i + 1,
len(imagePaths)))
name = imagePath.split(os.path.sep)[-2]
# load the input image and convert it from RGB (OpenCV ordering)
# to dlib ordering (RGB)
image = cv2.imread(imagePath)
Line 1, Column 1
Tab Size: 4
Python
19%
4:39 AM
4/2/2019

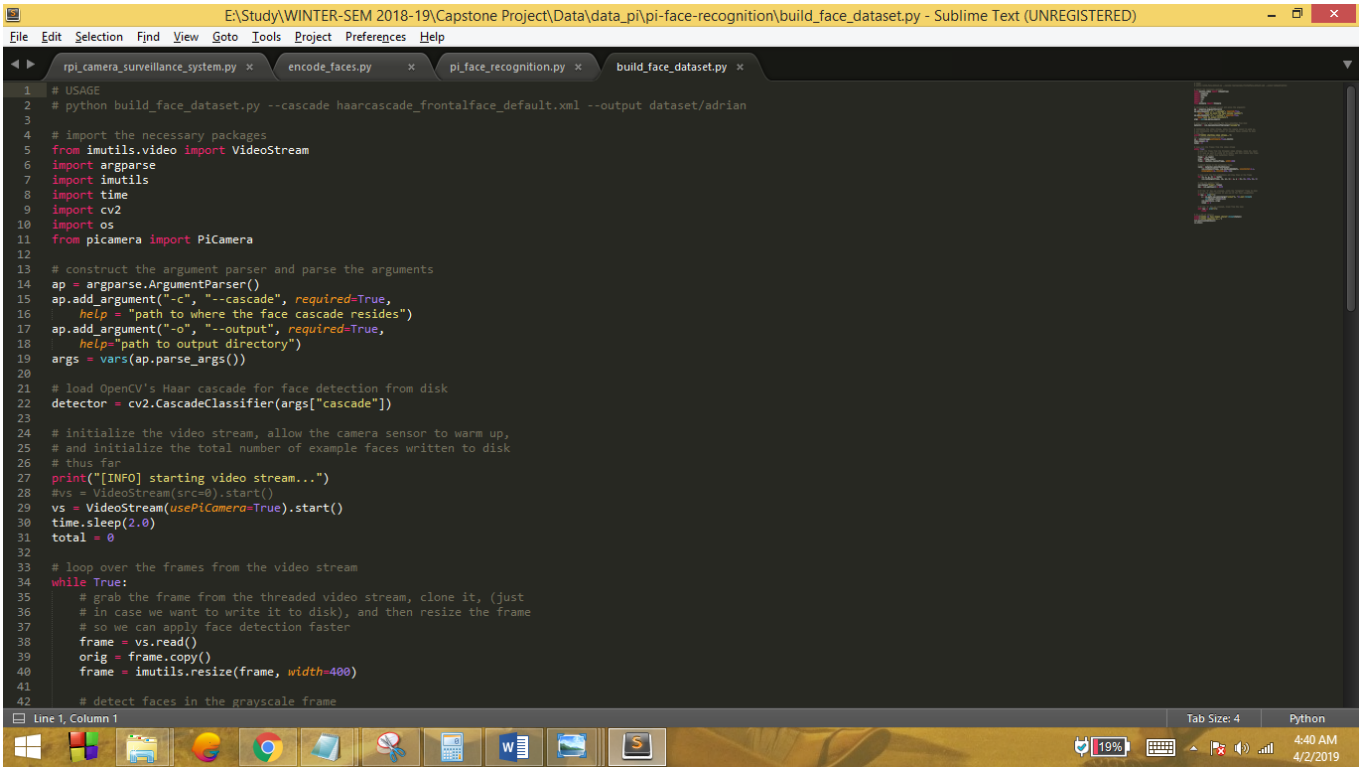

E:\Study\WINTER-SEM 2018-19\Capstone Project\Data\data_pi\pi-face-recognition\encode_faces.py - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
rpi_camera_surveillance_system.py
encode_faces.py
pi_face_recognition.py
build_face_dataset.py
name = imagePath.split(os.path.sep)[-2]
# load the input image and convert it from RGB (OpenCV ordering)
# to dlib ordering (RGB)
image = cv2.imread(imagePath)
rgb = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
# detect the (x, y)-coordinates of the bounding boxes
# corresponding to each face in the input image
boxes = face_recognition.face_locations(rgb,
model=args["detection_method"])
# compute the facial embedding for the face
encodings = face_recognition.face_encodings(rgb, boxes)
# loop over the encodings
for encoding in encodings:
# add each encoding + name to our set of known names and
# encodings
knownEncodings.append(encoding)
knownNames.append(name)
# dump the facial encodings + names to disk
print("[INFO] serializing encodings...")
data = {"encodings": knownEncodings, "names": knownNames}
f = open(args["encodings"], "wb")
f.write(pickle.dumps(data))
f.close()
Line 1, Column 1
Tab Size: 4
Python
19%
4:39 AM
4/2/2019

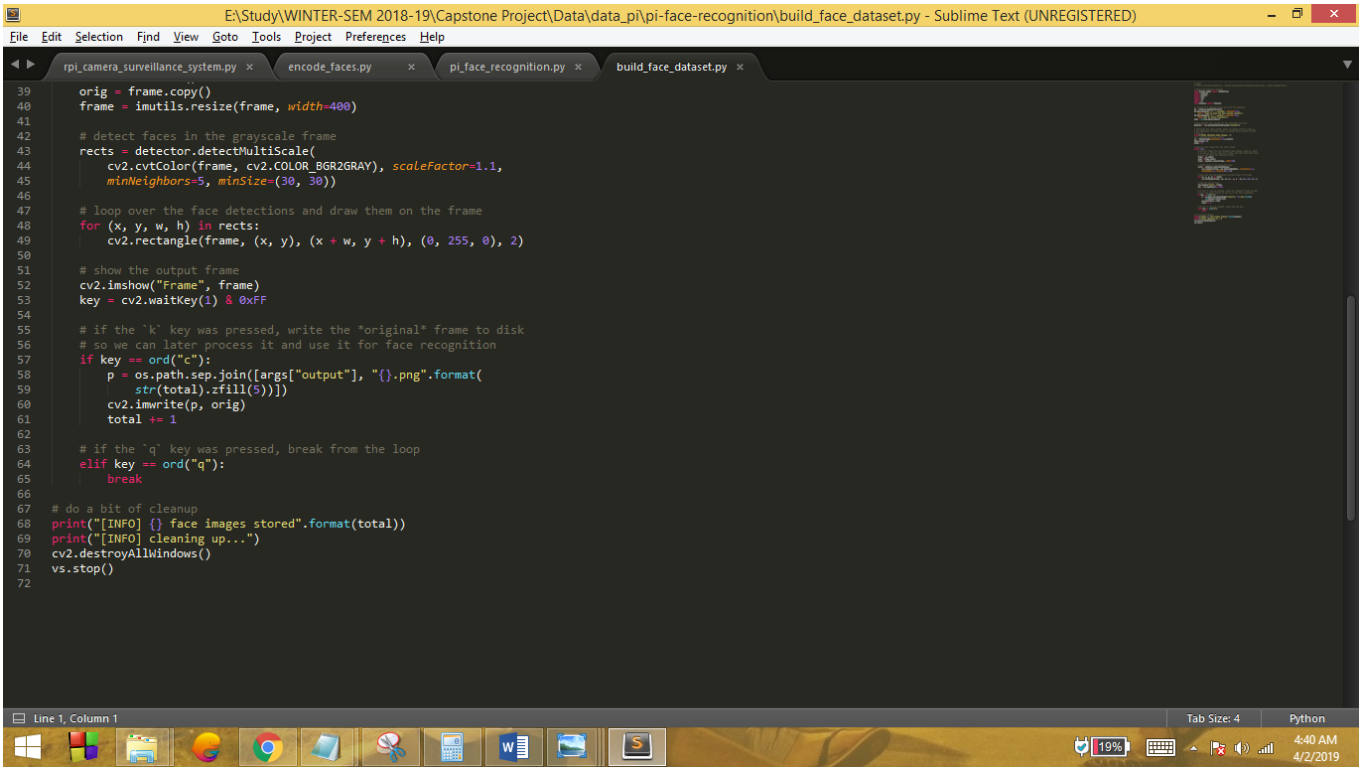

E:\Study\WINTER-SEM 2018-19\Capstone Project\Data\data_pi\pi-face-recognition\pi_face_recognition.py - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
rpi_camera_surveillance_system.py
encode_faces.py
pi_face_recognition.py
build_face_dataset.py
# USAGE
# python pi_face_recognition.py --cascade haarcascade_frontalface_default.xml --encodings encodings.pickle
# import the necessary packages
from imutils.video import VideoStream
from imutils.video import FPS
import face_recognition
import argparse
import imutils
import pickle
import time
import cv2
from picamera import PiCamera
import smtplib
from email.mime.multipart import MIMEMultipart
from email.mime.text import MIMEText
from email.mime.base import MIMEBase
from email import encoders
from email.mime.image import MIMEImage
fromaddr = "dhineshrajan1896@gmail.com"
toaddr = "ayush.sharma617@gmail.com"
mail = MIMEMultipart()
mail['From'] = fromaddr
mail['To'] = toaddr
mail['Subject'] = "Attachment"
body = "Please find the attachment"
red=24
green=16
import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)
GPIO.setup(red,GPIO.OUT)
GPIO.setup(green,GPIO.OUT)
GPIO.output(red,GPIO.LOW)
GPIO.output(green,GPIO.LOW)
# construct the argument parser and parse the arguments
ap = argparse.ArgumentParser()
Line 1, Column 1
Tab Size: 4
Python
19%
4:39 AM
4/2/2019

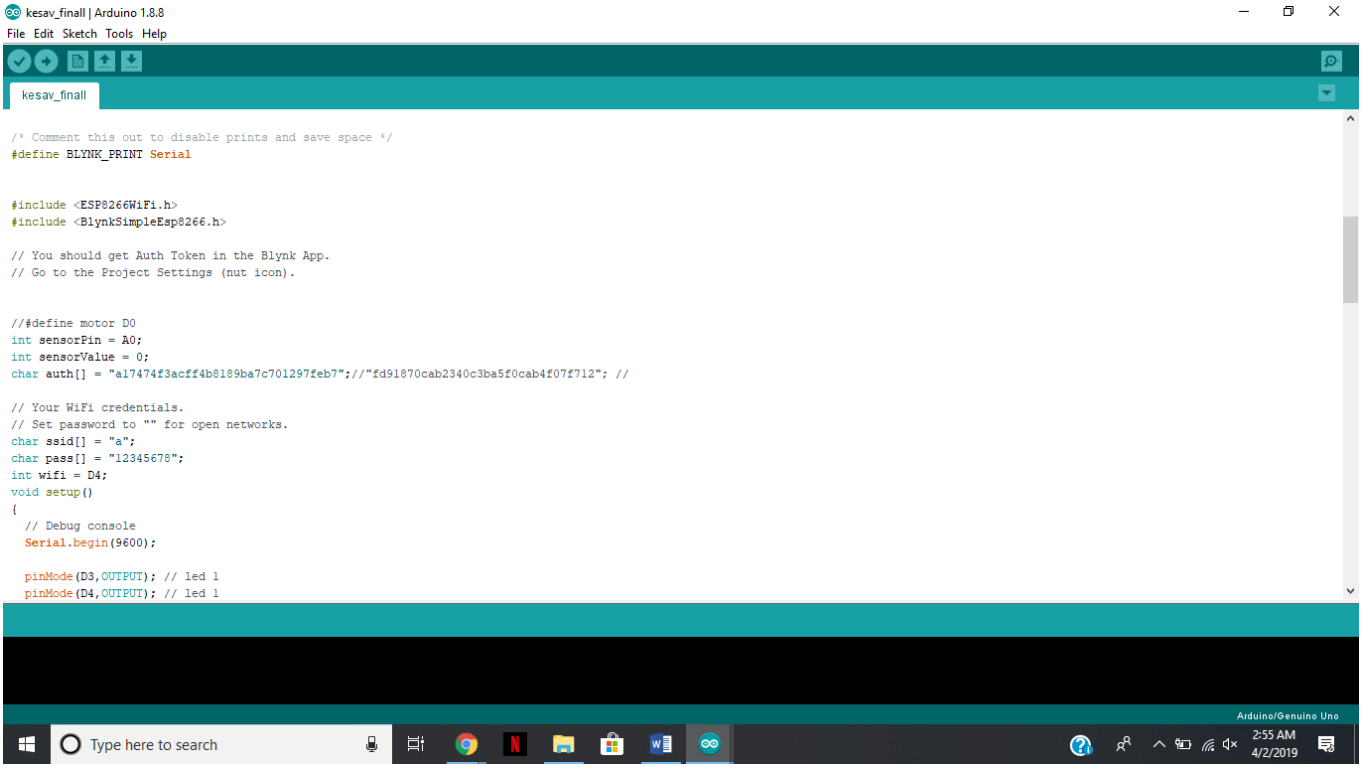

E:\Study\WINTER-SEM 2018-19\Capstone Project\Data\data_pi\pi-face-recognition\pi_face_recognition.py - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
rpi_camera_surveillance_system.py
encode_faces.py
pi_face_recognition.py
build_face_dataset.py
ap = argparse.ArgumentParser()
ap.add_argument("-c", "--cascade", required=True,
help = "path to where the face cascade resides")
ap.add_argument("-e", "--encodings", required=True,
help="path to serialized db of facial encodings")
args = vars(ap.parse_args())
# load the known faces and embeddings along with OpenCV's Haar
# cascade for face detection
print("[INFO] loading encodings + face detector...")
data = pickle.loads(open(args["encodings"], "rb").read())
detector = cv2.CascadeClassifier(args["cascade"])
# initialize the video stream and allow the camera sensor to warm up
print("[INFO] starting video stream...")
#vs = VideoStream(src=0).start()
vs = VideoStream(usePiCamera=True).start()
time.sleep(2.0)
# start the FPS counter
fps = FPS().start()
def sendMail(data):
mail.attach(MIMEText(body, 'plain'))
print (data)
dat='/home/pi/pi-face-recognition/%s'%data
print (dat)
attachment = open(dat, 'rb')
image=MIMEImage(attachment.read())
attachment.close()
mail.attach(image)
server = smtplib.SMTP('smtp.gmail.com', 587)
server.starttls()
server.login(fromaddr, "dhineshrajan1896@")
text = mail.as_string()
server.sendmail(fromaddr, toaddr, text)
server.quit()
def capture_image():
frame = vs.read()
#imageName = str(time.strftime("%Y_%m_%d_%H_%M")) + '.jpg'
imageName ='unknown.jpg'
Line 1, Column 1
Tab Size: 4
Python
19%
4:40 AM
4/2/2019

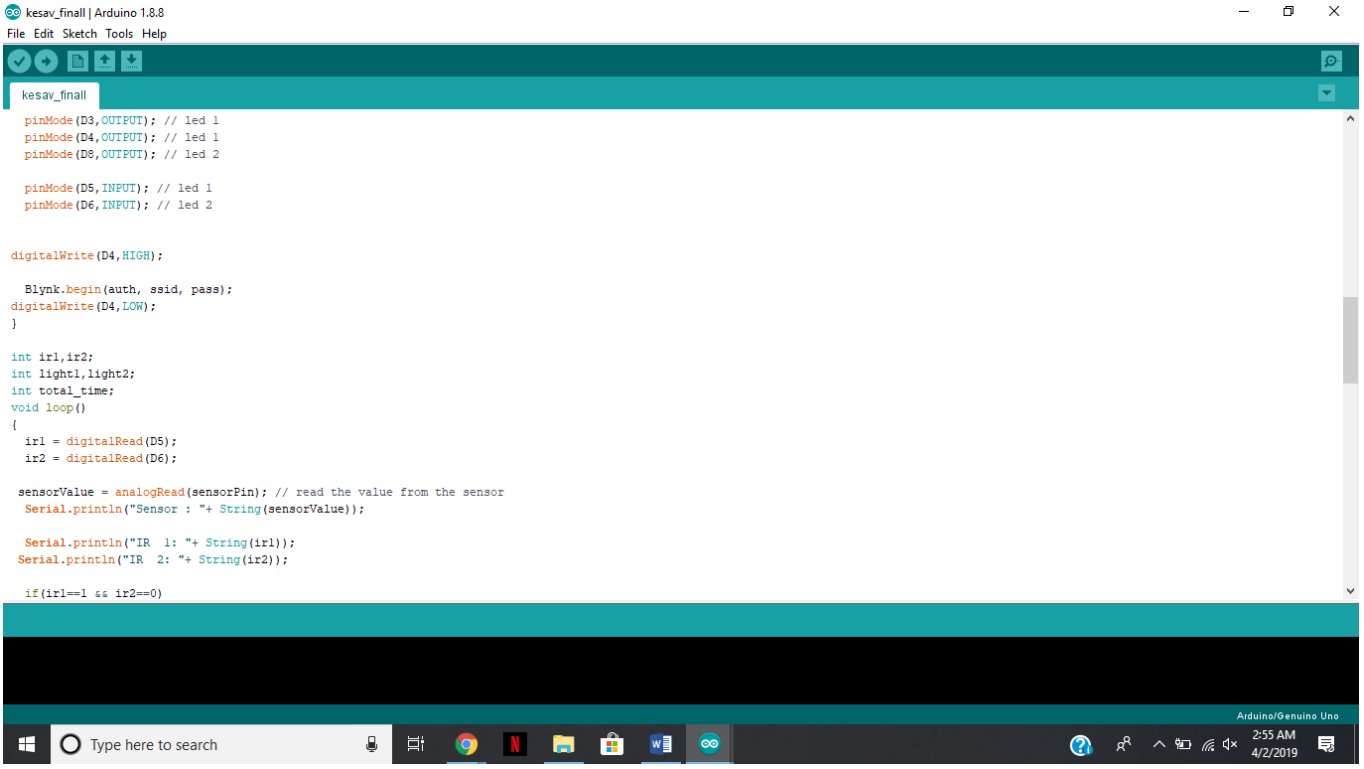

E:\Study\WINTER-SEM 2018-19\Capstone Project\Data\data_pi\pi-face-recognition\pi_face_recognition.py - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
rpi_camera_surveillance_system.py
encode_faces.py
pi_face_recognition.py
build_face_dataset.py
imageName ='unknown.jpg'
cv2.imwrite(imageName, frame)
time.sleep(1)
sendMail(imageName)
# loop over frames from the video file stream
while True:
# grab the frame from the threaded video stream and resize it
# to 500px (to speedup processing)
frame = vs.read()
frame = imutils.resize(frame, width=500)
# convert the input frame from (1) BGR to grayscale (for face
# detection) and (2) from BGR to RGB (for face recognition)
gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
# detect faces in the grayscale frame
rects = detector.detectMultiScale(gray, scaleFactor=1.1,
minNeighbors=5, minSize=(30, 30),
flags=cv2.CASCADE_SCALE_IMAGE)
# OpenCV returns bounding box coordinates in (x, y, w, h) order
# but we need them in (top, right, bottom, left) order, so we
# need to do a bit of reordering
boxes = [(y, x + w, y + h, x) for (x, y, w, h) in rects]
# compute the facial embeddings for each face bounding box
encodings = face_recognition.face_encodings(rgb, boxes)
names = []
# loop over the facial embeddings
for encoding in encodings:
# attempt to match each face in the input image to our known
# encodings
matches = face_recognition.compare_faces(data["encodings"],
encoding)
name = "Unknown"
# check to see if we have found a match
if True in matches:
Line 1, Column 1
Tab Size: 4
Python
19%
4:40 AM
4/2/2019

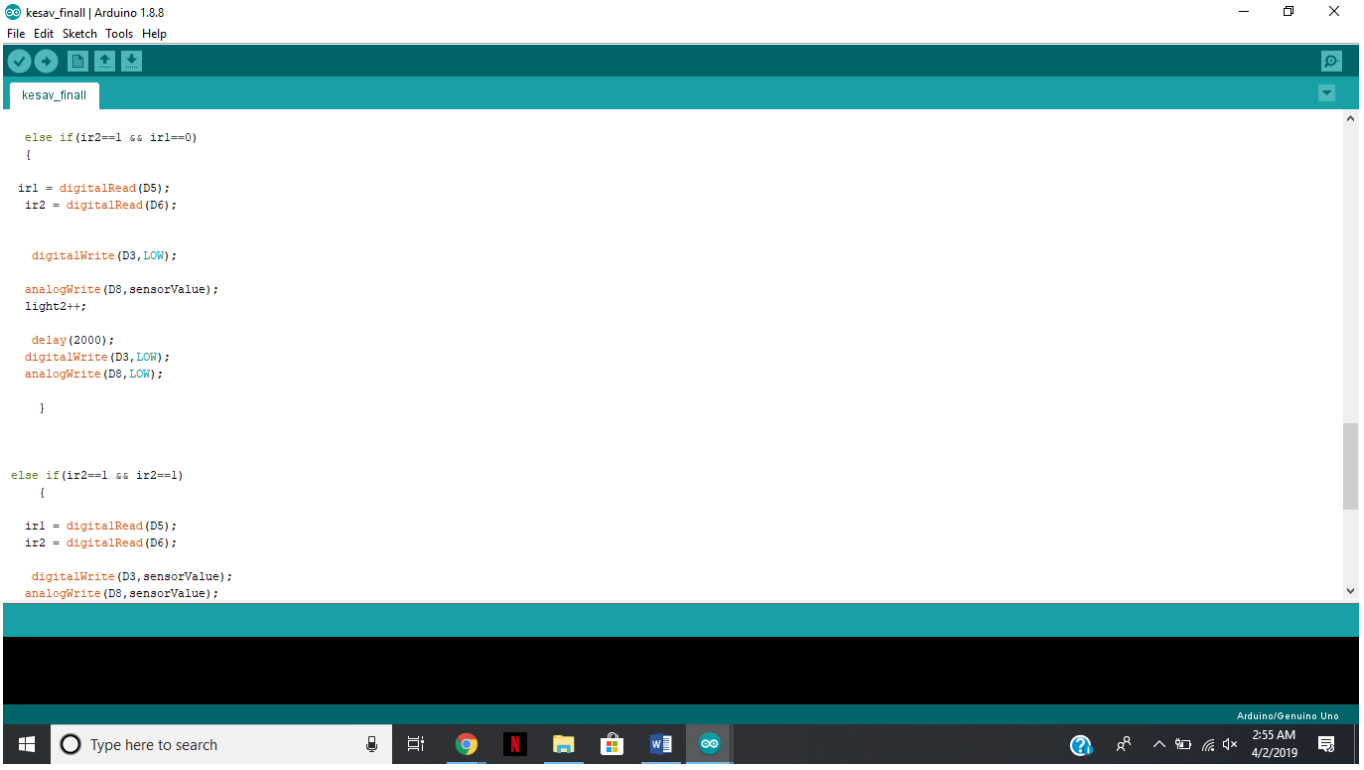

E:\Study\WINTER-SEM 2018-19\Capstone Project\Data\data_pi\pi-face-recognition\pi_face_recognition.py - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
rpi_camera_surveillance_system.py
encode_faces.py
pi_face_recognition.py
build_face_dataset.py
name = "Unknown"
# check to see if we have found a match
if True in matches:
# find the indexes of all matched faces then initialize a
# dictionary to count the total number of times each face
# was matched
matchedIdxs = [i for (i, b) in enumerate(matches) if b]
counts = {}
# loop over the matched indexes and maintain a count for
# each recognized face face
for i in matchedIdxs:
name = data["names"][i]
counts[name] = counts.get(name, 0) + 1
# determine the recognized face with the largest number
# of votes (note: in the event of an unlikely tie Python
# will select first entry in the dictionary)
name = max(counts, key=counts.get)
print (name)
GPIO.output(green,GPIO.HIGH)
time.sleep(2)
GPIO.output(green,GPIO.LOW)
if(name=="Unknown"):
GPIO.output(red,GPIO.HIGH)
capture_image()
GPIO.output(red,GPIO.LOW)
# update the list of names
names.append(name)
# loop over the recognized faces
for ((top, right, bottom, left), name) in zip(boxes, names):
# draw the predicted face name on the image
cv2.rectangle(frame, (left, top), (right, bottom),
(0, 255, 0), 2)
y = top - 15 if top - 15 > 15 else top + 15
cv2.putText(frame, name, (left, y), cv2.FONT_HERSHEY_SIMPLEX,
0.75, (0, 255, 0), 2)
Line 1, Column 1
Tab Size: 4
Python
19%
4:40 AM
4/2/2019

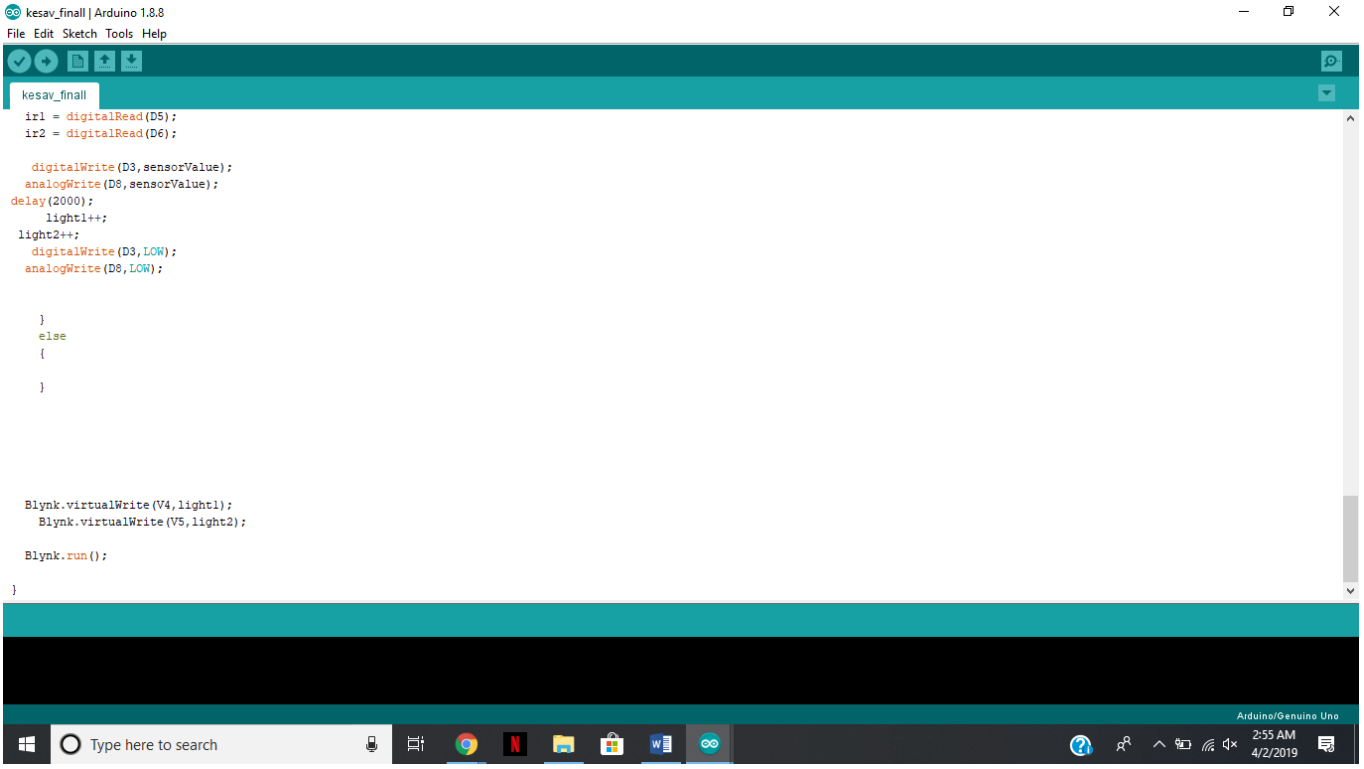

E:\Study\WINTER-SEM 2018-19\Capstone Project\Data\data_pi\pi-face-recognition\build_face_dataset.py - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
rpi_camera_surveillance_system.py
encode_faces.py
pi_face_recognition.py
build_face_dataset.py
# USAGE
# python build_face_dataset.py --cascade haarcascade_frontalface_default.xml --output dataset/adrian
# import the necessary packages
from imutils.video import VideoStream
import argparse
import imutils
import time
import cv2
import os
from picamera import PiCamera
# construct the argument parser and parse the arguments
ap = argparse.ArgumentParser()
ap.add_argument("-c", "--cascade", required=True,
help = "path to where the face cascade resides")
ap.add_argument("-o", "--output", required=True,
help="path to output directory")
args = vars(ap.parse_args())
# load OpenCV's Haar cascade for face detection from disk
detector = cv2.CascadeClassifier(args["cascade"])
# initialize the video stream, allow the camera sensor to warm up,
# and initialize the total number of example faces written to disk
# thus far
print("[INFO] starting video stream...")
#vs = VideoStream(src=0).start()
vs = VideoStream(usePiCamera=True).start()
time.sleep(2.0)
total = 0
# loop over the frames from the video stream
while True:
# grab the frame from the threaded video stream, clone it, (just
# in case we want to write it to disk), and then resize the frame
# so we can apply face detection faster
frame = vs.read()
orig = frame.copy()
frame = imutils.resize(frame, width=400)
# detect faces in the grayscale frame
Line 1, Column 1
Tab Size: 4
Python
19%
4:40 AM
4/2/2019


E:\Study\WINTER-SEM 2018-19\Capstone Project\Data\data_pi\pi-face-recognition\build_face_dataset.py - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
rpi_camera_surveillance_system.py
encode_faces.py
pi_face_recognition.py
build_face_dataset.py
orig = frame.copy()
frame = imutils.resize(frame, width=400)
# detect faces in the grayscale frame
rects = detector.detectMultiScale(
cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY), scaleFactor=1.1,
minNeighbors=5, minSize=(30, 30))
# loop over the face detections and draw them on the frame
for (x, y, w, h) in rects:
cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 255, 0), 2)
# show the output frame
cv2.imshow("Frame", frame)
key = cv2.waitKey(1) & 0xFF
# if the `k` key was pressed, write the *original* frame to disk
# so we can later process it and use it for face recognition
if key == ord("c"):
p = os.path.sep.join([args["output"], "{}.png".format(
str(total).zfill(5))])
cv2.imwrite(p, orig)
total += 1
# if the `q` key was pressed, break from the loop
elif key == ord("q"):
break
# do a bit of cleanup
print("[INFO] {} face images stored".format(total))
print("[INFO] cleaning up...")
cv2.destroyAllWindows()
vs.stop()
Line 1, Column 1
Tab Size: 4
Python
19%
4:40 AM
4/2/2019


kesav_finall | Arduino 1.8.8
File Edit Sketch Tools Help
kesav_finall
/* Comment this out to disable prints and save space */
#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
// You should get Auth Token in the Blynk App.
// Go to the Project Settings (nut icon).
//#define motor D0
int sensorPin = A0;
int sensorValue = 0;
char auth[] = "a17474f3acff4b8189ba7c701297feb7";//"fd91870cab2340c3ba5f0cab4f07f712"; //
// Your WiFi credentials.
// Set password to "" for open networks.
char ssid[] = "a";
char pass[] = "12345678";
int wifi = D4;
void setup()
{
// Debug console
Serial.begin(9600);
pinMode(D3,OUTPUT); // led 1
pinMode(D4,OUTPUT); // led 1
Arduino/Genuino Uno
Type here to search
2:55 AM
4/2/2019


kesav_finall | Arduino 1.8.8
File Edit Sketch Tools Help
kesav_finall
pinMode(D3,OUTPUT); // led 1
pinMode(D4,OUTPUT); // led 1
pinMode(D8,OUTPUT); // led 2
pinMode(D5,INPUT); // led 1
pinMode(D6,INPUT); // led 2
digitalWrite(D4,HIGH);
Blynk.begin(auth, ssid, pass);
digitalWrite(D4,LOW);
}
int ir1,ir2;
int light1,light2;
int total_time;
void loop()
{
ir1 = digitalRead(D5);
ir2 = digitalRead(D6);
sensorValue = analogRead(sensorPin); // read the value from the sensor
Serial.println("Sensor : "+ String(sensorValue));
Serial.println("IR 1: "+ String(ir1));
Serial.println("IR 2: "+ String(ir2));
if(ir1==1 && ir2==0)
Arduino/Genuino Uno
Type here to search
2:55 AM
4/2/2019


kesav_finall | Arduino 1.8.8
File Edit Sketch Tools Help
kesav_finall
else if(ir2==1 && ir1==0)
{
ir1 = digitalRead(D5);
ir2 = digitalRead(D6);
digitalWrite(D3,LOW);
analogWrite(D8,sensorValue);
light2++;
delay(2000);
digitalWrite(D3,LOW);
analogWrite(D8,LOW);
}
else if(ir2==1 && ir2==1)
{
ir1 = digitalRead(D5);
ir2 = digitalRead(D6);
digitalWrite(D3,sensorValue);
analogWrite(D8,sensorValue);
Arduino/Genuino Uno
Type here to search
2:55 AM
4/2/2019


kesav_finall | Arduino 1.8.8
File Edit Sketch Tools Help
kesav_finall
ir1 = digitalRead(D5);
ir2 = digitalRead(D6);
digitalWrite(D3,sensorValue);
analogWrite(D8,sensorValue);
delay(2000);
light1++;
light2++;
digitalWrite(D3,LOW);
analogWrite(D8,LOW);
}
else
{
}
Blynk.virtualWrite(V4,light1);
Blynk.virtualWrite(V5,light2);
Blynk.run();
}
Arduino/Genuino Uno
Type here to search
2:55 AM
4/2/2019