Smart Doorbell for Home Security using Raspberry Pi

Sahan Tharanga Lamahewa

158763B

Faculty of Information Technology

University of Moratuwa

February 2019

Smart Doorbell for Home Security using Raspberry Pi

Sahan Tharanga Lamahewa

158763B

Dissertation submitted to the Faculty of Information Technology, University of Moratuwa, Sri Lanka for the partial fulfillment of the requirements of the Honours Degree of Bachelor of Science in Information Technology.

February 2019

Declaration

I declare that this thesis is my own work and has not been submitted in any form for another degree or diploma at any university or other institution of tertiary education. Information derived from the published or unpublished work of others has been acknowledged in the text and a list of references is given.

Name of Student: Mr. S.T. Lamahewa

Signature of the Student:_____

Date:

Supervised by:

Name of the Supervisor: Mr. B.H. Sudantha

Signature of the Supervisor:_____

Date:

Dedication

I would dedicate this thesis to my beloved family members who have never failed to give me a tremendous support, for giving all not only throughout my project but also throughout my life as well. As well they teach me that even the largest task can be accomplished if it is done one step at a time,

Acknowledgement

I wish to express my eternally grateful and gratitude to my project supervisor Mr. B.H. Sudantha, Senior Lecturer, Faculty of Information Technology, University of Moratuwa untiringly shared his knowledge, precious guidance, encouragement, advices and help given to complete the project successfully. Also, I thank to Prof A.S. Karunananda who gave me guidance and advices to complete the project successfully.

I am also grateful to previous researchers who have contributed to the domain of distributed applications, academic and non academic staff of Faculty of Information Technology, University of Moratuwa and also batch mates who have given me valuable feedbacks to improve the project.

Importantly I would like to thank my wife and my family members for the encouragement and undying support not only throughout my project but also throughout my life.

Finally, the support of University of Moratuwa is also sincerely acknowledged.

Abstract

In recent years, the crimes targeting household properties has grown considerably high. Therefore, it's necessary to come up with a proper solution to secure a household and to let house owners aware about the surrounding activities. The proposed system is a hardware device that combined with technologies like OpenCV deep learning based facial recognition and WebRTC. Further, PhoneGap is used to develop the application for hand-held devices which will interact with the client (house owner). To implement one-way video conferencing between the user (house owner) and visitor, WebRTC with Socket.io will be used. Apart from that, WebRTC with Socket.io combined technologies are responsible for streaming live feed of household environment. Motion detection and video capturing are done using PIR sensor, camera module, speaker and microphone. The system will initiate by turning on the camera when either a user presses on the doorbell 'bell' push button or PIR sensor detects motion. When the visitor presses the doorbell 'call' push button, visitor is connected to an one-way voice conference call with the home owner (House owner can see and hear the voice of the visitor through a hand-held device and visitor can only hear the voice of the owner through doorbell device.). Home owners can record a pre-defined message for known users as well. The camera module will perform facial detection by resolving a classifier in OpenCV. Afterwards, an image will be captured with a brief video. Facial information is saved in facial repository and video files are stored in the cloud. During the facial recognition if a match is found, the house owner will be notified with visitor details. For unknown users, house owner will be prompted to identify. The system can be improved and customized as a security system that can be installed in server rooms and also it can be enhanced to use as a student attendance recorder module.

Keywords – Raspberry Pi, WebRTC, PhoneGap, OpenCV

Table of Contents

List of Figures

List of Tables

Chapter 1

1. Introduction

1.1. Prolegomena

To every person one of the main concerns of living is security. It is as important as having a roof over one's head or having water to drink. In this report ensuring security of a person's house using latest technologies to design and implement an effective solution will be discussed.

Many intruders often use the front door of house to access the inside. Therefore, if a front door of house is armed with a proper security method, these types of intrusions can be avoided. Not only avoiding intrusion the system can be adapted to assists disable people, detect other types of disturbances around the household such as a fire or flood as well. In the next sub menus introduction to the system will be discussed further.

1.2. Aim and Objectives

The aim of this project is to develop a computer-based cost effective solution to address security threats to the household properties and household members.

- Critical study of the literature in the approaches to develop household smart doorbells.
- Critical survey on techniques to solve the problem
- Design and development of a solution
- Evaluation of the solution
- Production of the final documentation

1.3. Background and Motivation

Security is an essential basic need in human life. According to Maslow's Hierarchy of needs, safety only becomes second to the basic physiological needs such as air, water and food [1]. Home security is essential because mainly it will protect a person's home and family from intruders. Not only burglars but drug addicts seeking money, rapists and scammers also pose a similar security threat to the property and they put the lives of household members at risk as well. In addition, it will also help keeping one's valuables safe while monitoring the home environment during the absence of owner. Therefore, this entire project addresses on how to fulfil one of those basic human need security.

According to Sri Lanka Police Department, there were around 35978 crimes reported in 2017 and among them majority were house breakings (8913) [2]. By observing these statistics, it can be clearly seen the importance of having proper security in a household. According to the statistics received from Sri Lanka Police Department, it can be stated that the number of crimes targeting households only increases day by day. When a thief breaks into a house, lives of the members who are inside the house when the break-in occurs will also face a critical condition as they will be in the middle of a dangerous situation. There's a high possibility of people getting hurt during a robbery as these types of news are typical with household breakings. The thief does not necessarily need to be armed to hurt a household member during such an event. At the end of any household break-ins two outcomes are certain. Either people can lose their valued possessions or in the worst case scenario lose their lives or something lesser unless the thief was not successful in doing such.

All of these unfortunate incidents can be avoided if a house is armed with a proper security system which will not allow intruders to get in to their property. If a thief has already broken into a house at least the member will get an alert to take necessary actions to mitigate the situation. This is rather an effective solution and it is going to prevent people from suddenly finding themselves trapped in the middle of a dangerous situation.

1.4. Problem in brief

With the rise of the doorstep crimes in Sri Lanka, it is necessary to have a security system installed at one's household to ensure the safety of the family members and the property. Since most of the available security systems are higher in cost and due to the complexity of some systems, general public avoid using home security products. The purpose of the research is to provide a low cost, platform independent security solution for home security.

1.5. Solution in brief

The proposed system is mainly using Raspberry Pi combined with several other technologies such as OpenCV, WebRTC & PhoneGap to deliver a low cost, platform independent, simple & distributed system to be installed in a household to mitigate doorstep crimes.

1.5.1. Users

Users of the proposed solution are House Owners

1.5.2. Input

- Visitor's images are captured by the camera
- Visitor's motion is detected by the PIR sensor
- Two physical push buttons
    1. Doorbell button – Ring the doorbell buzzer
    2. Call button – This button allows the visitor to have a one-way video conference with the house owner

1.5.3. Output

- Buzzer – Doorbell rings when a visitor presses the doorbell button
- Push notifications – a house owner will receive push notifications in three instances.
    1. When a known visitor is at the door, visitor's name will be shown to the user
    2. When an unknown visitor is at the door, user will be notified as an 'UNKONWN VISITOR'

3. When the call button is pushed by the visitor, user will receive a link to initiate a web conference with the visitor at the door

- Email – user will receive an email containing the image of the visitor
- Visitor history log – user can view the visitor history log via logging into the client app

1.5.4. Process

PIR sensor captures visitor's motion and triggers the camera module. Then camera module takes snapshots of visitors. Face recognition algorithms helps to matches similar faces in facial repository. If it matches, it returns visitors details and alert the house owner with the visitor's presence and read the pre-defined messages set by house owner. Otherwise send and notification to house owner to update the facial repository via client application. One-way video conference and video streaming facility of house hold environment.

1.5.5. Technologies

Hardware	Software
Raspberry Pi 3 B+	Raspbian Operating System
Web Camera	Web RTC with Socket.io
PIR Sensor	OpenCV 4.0
USB microphone	Apache Cordova (PhoneGap)
Bluetooth Speaker	Dropbox API
MicroSD Card	Semantic-UI
	Python
	Node.js

Table 1.1 - Technologies used in Smart Doorbell

1.5.6. Features

- Motion detection
- Visitor's face recognition
- Cloud image backups
- 24x7 live video of household environment
- Pre-defined voice message
- Live one-way video communication
- Platform independent

1.5.7. System Requirements

- Micro USB power supply, 100-240v input, 5.1v 2.5A output
- MicroSD card with Raspberian OS
- Android Nougat and above versions, iOS 10+, Windows 10
- Internet via Wi-Fi or Ethernet
- Python 3 environment

1.6. Summary

In conclusion, it is essential to have and expect a secure household as it's one of the main concerns of all human beings. These crime rates have increased and people are unable to feel safe about leaving their household premises unattended because of burglars and other intruders. It is a must to have a proper security system installed in your house and to get notification no matter where the user is through the installed security system. It is also better to find a cost effective application to monitor the household surrounding.

Chapter 2

2. Current Issues in Home Security Devices

2.1. Introduction

In the previous chapter, a comprehensive introduction about the research project was provided. This chapter presents all literature review done on similar researchers. The past work done regarding the subject will be carefully anaslyzed and their disadvantages and advantages are taken into consideration. In the chapter, previous research' features and functionalities are thoroughly discussed. Finally, a summarization of past work done on the research topic is going to be presented.

2.2. Other solutions for home security

Several studies have been conducted on the research area when going through the literature of similar work done by researchers around the world. In one of the most recent studies, a model driven development process for home security was developed based on the IoT architecture [3]. With the rapid development and latest additions to IoT, there's a tendency to use IoT platform in security focused software. They have used Telegrams to transmit data through the layers of IoT architecture. This system is efficient as it uses less power and can be easily implemented without using high-end gadgets. Regardless of being cost effective, this system lacks communication between the visitor and facial recognition features.

Another security surveillance system was implemented to be used inside buildings [4]. The system is developed using Raspberry Pi Single Board Computer (SBS) with WiFi network. The main features of the system are remote user alerts and live video streaming as well mobility as the system can be set up anywhere easily. All alerts, live streaming will be done via the WiFi network connectivity. For the surveillance purpose they have used wireless sensor nodes and controller section. It can be clearly seen the system is budget friendly and due its smaller size, the system can be handled with ease. Also IoT module gives better working range. The system is mainly capable of intruder alerts as well as fire detection. They have inserted the live video streaming as an additional feature of the system. The system is fairly low on budget, it's compact and can be afforded by a lot of users as security is an important concern for everyone.

Another recently developed home security system consists of a doorbell which can be connected to the user's smartphone and enabling real time user monitoring through an user interface [5]. The doorbell is IoT based and using their smartphone, can view a live feed of the surrounding environment. Apache web server will store the live feed coming from a web cam. Whenever someone presses on the doorbell, an SMS will be sent along with photos taken from a CCTV camera. Even though, the system provides live feed and photos, it is unable to identify visitors and does not allow communication between visitor and the owner.

A mobile application was developed mainly to assist the elderly and to mitigate doorstep crimes [6]. The app provides the elderly person with a choice to call for help if they deem it necessary when there's a person at the door. The app provides facial recognition facility unlike the discussed researches up to now. However, the developed app is platform dependent. A standalone app which was designed specifically to aid people with hearing disabilities has used Raspberry pi to lower development costs [7]. This system also provides a wearable device which acts as a receiver. The differently abled person will get notified via a transmitter fixed to the doorbell. It sends images and SMS to the user. The wearable device has a vibrator to alert the wearer when there's someone at the door. This system lacks facial recognition facilities but sends images of the visitor nevertheless.

Further, a more advanced embedded door access control system has been developed with ZigBee i.e a wireless network technique [8]. This system is armed with facial recognition feature. User can get multiple alerts via different platforms such as Email & SMS. Yet the system has not provided a communication method between the user and visitor. Another solution was proposed to create a smart doorbell to assist disabled people [9]. The proposed model will capture an image of the visitor once clicked on the doorbell using the camera installed. The image will be then processed to identify the visitor. Users will get notified via a notification sent to their mobile devices. The web interface of the system will be developed using AngularJS and Bootstrap. REST will be used for queries and data modifications. Additionally, this system needs improvements in areas such as facial recognition and fingerprint recognition as backup validation methods.

A research done in mid-twenties has proposed a security system that uses home wireless network and GSM [10]. This consists of an infrared security nodes and fire alarm nodes as well. The system mainly functions as a warning system when there's an intruder at the door. After detecting an unwanted presence at the door, the user will be notified via an SMS. Instead of using Internet like most of the security systems do, this system uses GSM/GPRS technologies. It was justified as having a wide spread coverage as opposed to other networks and the inability of unwanted network monitoring.

In conclusion most of the already built security systems or proposed ideas, seem to lack at least one crucial feature that is essential to a home security system. Some of these systems are far too complex and takes a lot of money to implement but managed to deliver all the expected features. In some cases, the systems are low cost and efficient but again lacks important features such as communication with the user or facial recognition features. The proposed system has addressed many of these problems by implementing a cost effective yet feature rich system to secure a household environment.

Product/Project	**Technologies used**	**Positive features**	**Negative features**
IoT Application Development: Home Security System	Raspberry Pi SaaS	Control door accessibility Capture faces	• Unavailability of facial recognition feature • Unable to communicate with the visitor
IoT Smart Bell Notification System: Design and Implementation	Apache web server SMS	Video saving Storing images Warning notice GUI features	• Unable to communicate with the visitor • Unavailability of facial recognition feature
Enhanced smart doorbell system on face recognition	Raspberry Pi model B ARMv7 Cortex-A7 OpenCV	Image recognition accuracy in controlled environment	• Complexity

	XRDP SQL, PHP, Apache Linux	Low power consumption Resources optimization Improved operation speed	
Doorstep: A doorbell security system for the prevention of doorstep crime	Raspberry Pi 2 Windows Universal Platform (UWP) app	Face detection & capturing	• Platform Depended (Android)
PiCam: IoT based Wireless Alert System for Deaf and Hard of Hearing	Raspberry Pi Bluetooth SMS	Low cost Storing images	• Cannot communicate with visitor • Unavailability of facial recognition feature
Web-Based Online Embedded Door Access Control and Home Security System Based on Face Recognition	Raspberry pi ZigBee Image processing techniques SMS	Face detection and recognition Control door accessibility	• Unable to communicate with visitor
Smart-dell Doorbell: An ICT solution enhance inclusion of disabled people	Single Board Computer Raspberry Pi Android SDK Django Python REST	Facial recognition Mobile notification One way audio communication	• Motion detection is unavailable • Video feed & images are not stored

	OpenCV XMPP, Apache web server WSGI		
A Low Cost GSM/GPRS Based Wireless Home Security System	GSM/GPRS	Low cost Low power consumption Control door accessibility	• Primary Notification method (SMS)

Table 2.1 - Comparison and contrast of past works

2.3. Summary

Most of these research projects have their own pros nevertheless some drawbacks were observed during the literature review. A project could be low on cost but it only facilitates the basic functionalities when considering security concerns of a household. Some projects have used very complex technologies which is not very effective in the long run as these sorts of projects tend to require higher budgets to implement and maintain. An ordinary person is unable to even make a simple configuration to the system by themselves due to the complexity. In conclusion, the proposed system will provide a compact yet simple solution for the same issues that are important to be tackled in a security system in a cost-effective manner as well.

Chapter 3

3. Technology Adapted

3.1. Introduction

Traditional automated doorbells/door phones facilitate many features like surveillance video, communication ability between visitor and house owner, face detection and recognition, 24x7 video streaming facility, accessible through hand held devices, pre-defined voice messages for known visitors, video footage when an incident occur in front of house owner's doorstep and access door control. Nevertheless, most of these devices are platform depended (Android or iOS frameworks). In the current market, only a handful of devices exists which are equipped with all the essential features mentioned above. Due to higher costs most of these products cannot be afforded by normal people.

The proposed research can provide a low cost smart doorbell offering all of the features mentioned above except the access door control feature. This solution contains hardware device (smart doorbell), mobile application for users' (house owners) hand held devices and intermediate servers to manage the communication between hardware device and mobile web application.

It is evident from the literature that one of the prominent application segment of Internet of Things framework includes the Security Sector. It is essential to have a unique yet low cost solution to prevent theft and ensure security of a household and members of a family [3]. Smart home security control systems have become vital in daily life [11]. Home security system is needed for occupants' convenience and safety [3]. Safety and security rank highly in the priorities of older people and differently abled people on both an individual and policy level. Older people are commonly targeted as victims of doorstep crime, as they can be perceived as being vulnerable. As a result, this can have a major effect on the victim's health and wellbeing. [4]

3.2. Technologies

3.2.1. Raspberry Pi 3 B+ Model

The Raspberry Pi is a low cost, credit-card sized computer that plugs into a computer monitor or TV, and uses a standard keyboard and mouse [12]. Raspberry pi takes an advantage in its size, portability, cost and programmability in most of the IoT projects. One powerful feature of the Raspberry Pi is the row of GPIO pins along the top edge of the board. These pins are a physical interface between the Raspberry Pi and the outside world. At the simplest level, you can think of them as switches that you can turn on or off (input) or that the Pi can turn on or off (output). There are 40 pins on the Raspberry, and they provide different functions [13]. Other than GPIO pins, the board is equipped with 4 USB ports, Ethernet port, 802.11n wireless LAN and Bluetooth.

In this solution, Raspberry Pi Model 3 B+ is used as the base of the smart doorbell. Web Camera, PIR sensor, 2 push buttons, Speaker and microphone is directly connected to Raspberry Pi 3 B+ module. PIR sensor and 2 push buttons are connected to Raspberry Pi via GPIO interface. Web camera and microphone are connected to USB interface. Speaker connects via Bluetooth. From now onwards, the software platform capability of Raspberry Pi is described.

Raspberry Pi runs on top of the Raspbian OS. It facilitates to run python scripts inside the Raspberry Pi. Additionally, the node server setup is shown in next figure.

3.2.2. Apache Cordova (PhoneGap)

Apache Cordova enables software programmers to build applications for mobile devices using CSS3, HTML5, and JavaScript instead of relying on platform-specific APIs like those in Android, iOS, or Windows Phone [14]. It enables wrapping up of CSS, HTML, and JavaScript code depending on the platform of the device. It extends the features of HTML and JavaScript to work with the device. The resulting applications are hybrid, meaning that they are neither truly native mobile application (because all layout rendering is done via Web views instead of the platform's native UI framework) nor purely Web-based (because they are not just Web apps, but are packaged as apps for distribution and have access to native device APIs).

PhoneGap is a key technology to preserve platform independency of the house owner's app. In this case PhoneGap is used to produce house owner's app. This app facilitates to view live streaming of house owner's doorstep, start one-way communication with visitor and configure pre-defined messages for a particular visitor.

3.2.3. Face Recognition with Deep Learning

Deep learning is a machine learning technique that teaches computers to do what comes naturally to humans: learn by example. Deep learning is a key technology behind driverless cars, enabling them to recognize a stop sign, or to distinguish a pedestrian from a lamppost. It is the key to voice control in consumer devices like phones, tablets, TVs, and hands-free speakers. Deep learning is getting lots of attention lately and for good reason. It's achieving results that were not possible before [15].

In deep learning, a computer model learns to perform classification tasks directly from images, text, or sound. Deep learning models can achieve state-of-the-art accuracy, sometimes exceeding human-level performance. Models are trained by using a large set of labeled data and neural network architectures that contains many layers. [15]

Computers are not smarter than humans to identify human faces. In the proposed system, face detection and face recognition are done using Deep Learning techniques.

3.2.4. WebRTC

Web Real-Time Communications (WebRTC) is a specification for a protocol implementation that enables web apps to transmit video, audio and data streams between client (typically a web browser) and server (usually a web server) [16].

In this solution WebRTC is used in one-way video communication between house owner and visitor. Visitor's saved images are sent to server by using this technology.

3.2.5. WebSockets

A WebSocket is a standard protocol for two-way data transfer between a client and server. The WebSockets protocol does not run over HTTP, instead it is a separate implementation on top of TCP [17].

WebSockets connection supports full-duplex communication between client and servers. Either client and server can push data to other or sever can push data to client via an established connection.

In this solution, WebSocket is used to pass messages between smart doorbell and house owner's client app via the signaling server. Throughout the solution Socket.io javascript library is used to implement WebSocket.

3.2.6. Python

Python 3.0 perfectly works in Raspberry Pi. In the proposed system, face recognition, face detection and image capturing is done by using OpenCV-python library. Also python combines with Numpy (Optimized library for numerical operations) make tasks easier and efficient.

3.2.7. Node.js

Node.js eliminates the waiting, and simply continues with the next request. Node.js runs single-threaded, non-blocking, asynchronously programming, which is efficient for the memory. After taking these features into consideration, node.js is used to implement the signalling server [18].

Chapter 4

4. Using Raspberry Pi for Home Security

4.1. Introduction

In the previous chapter, all the adapted technologies to develop the proposed system were thoroughly discussed. In this chapter, the use of Raspberry Pi is comprehensively discussed. Furthermore, the entire system's inputs, outputs, the process is also given in details. In addition, a flow diagram of the system is provided for better understanding of the system. System users and main features are mentioned at the end of the chapter.

4.2. Input

Mainly Smart Doorbell project collects input form hardware components (web camera and PIR sensor) and user will be able to feed inputs via house owner's app. This project detects the motion of the visitor and captures images and videos of visitor. These images are used to recognize the visitor. Motion detection captured in PIR sensor and after the trigger is fired, PIR sensor will automatically turn on the camera module. There are two physical push buttons in the device and one button triggers the buzzer and second button make a one-way video conference call between the user and visitor.

If unidentified visitor is detected in the system, user will be able to update the necessary details according to that user. As an example, if the visitor is your friend, you can tag him/her by name. Otherwise unknown user can be tagged in any identifiable name.

4.3. Output

After accepting the image of a visitor by Smart Doorbell product, it is then should to identify whether the visitor is a known or unknown person. If he/she is known, send push notification and email to user that a known visitor at the doorstep. Otherwise it will send push notification and email stating that an unknown visitor is at the doorstep and if the owner knows the person owner can update details by using the house owner's app. When the visitor presses the first push button it will make the buzzer noise. User

can download archived videos and images stored in cloud space. User will be able to view and download visitors' activities and log records.

4.4. Process

The system will initiate by turning on the web camera when either a user presses on the doorbell push button or PIR sensor detects motion. Then the camera captures 7 images of the visitor. Next face detection and recognition algorithms are triggered, and it checks whether the visitor is previously stored in face database or not. Face detection will extract the faces from captured images using Histogram of Oriented Gradient algorithm (HOG). After, the system runs the face recognition algorithm. Simply this will identify facial landmarks from captured visitor's images and convert into 128-d (list of 128 real numbers) then checks the match with all 128-d embedded in known face in the database. During the facial recognition, if a match is found, the house owner will be notified with visitor details via push notification and an email. For unknown visitors, house owner will be prompted to identify the visitor and house owner will be able to update the visitor's details by logging in to house owner's app. During midnight, unknown visitor's face extractions is converted into 128-d embedding and stored in the database.

By pressing on the call push button, the visitor can get a one-way voice conference call with the user. WebRTC and Socket.io technologies will bridge the gap between two hosts by connecting the doorbell and house owner's app which are in different networks and assisting data streaming between the doorbell and house owner's app. WebRTC supports sharing streaming data between hosts. Nevertheless, it needs proper mechanism to coordinate communication and to control signals. This process is called as 'Signalling' and it is not specified in WebRTC standards. Socket.io library fill the gap in signalling. Signalling server has the ability to connect users who are not located in the same network. Video files and images are stored in the cloud space using Dropbox API. A House owner also has the ability to pre-record a message for known visitors. When the system found a match after facial recognition, the pre-defined message will be automatically played for the visitor. User also has the ability to view logs such as visitor activities. Through the client app user will be able to get a live feed of the household environment.

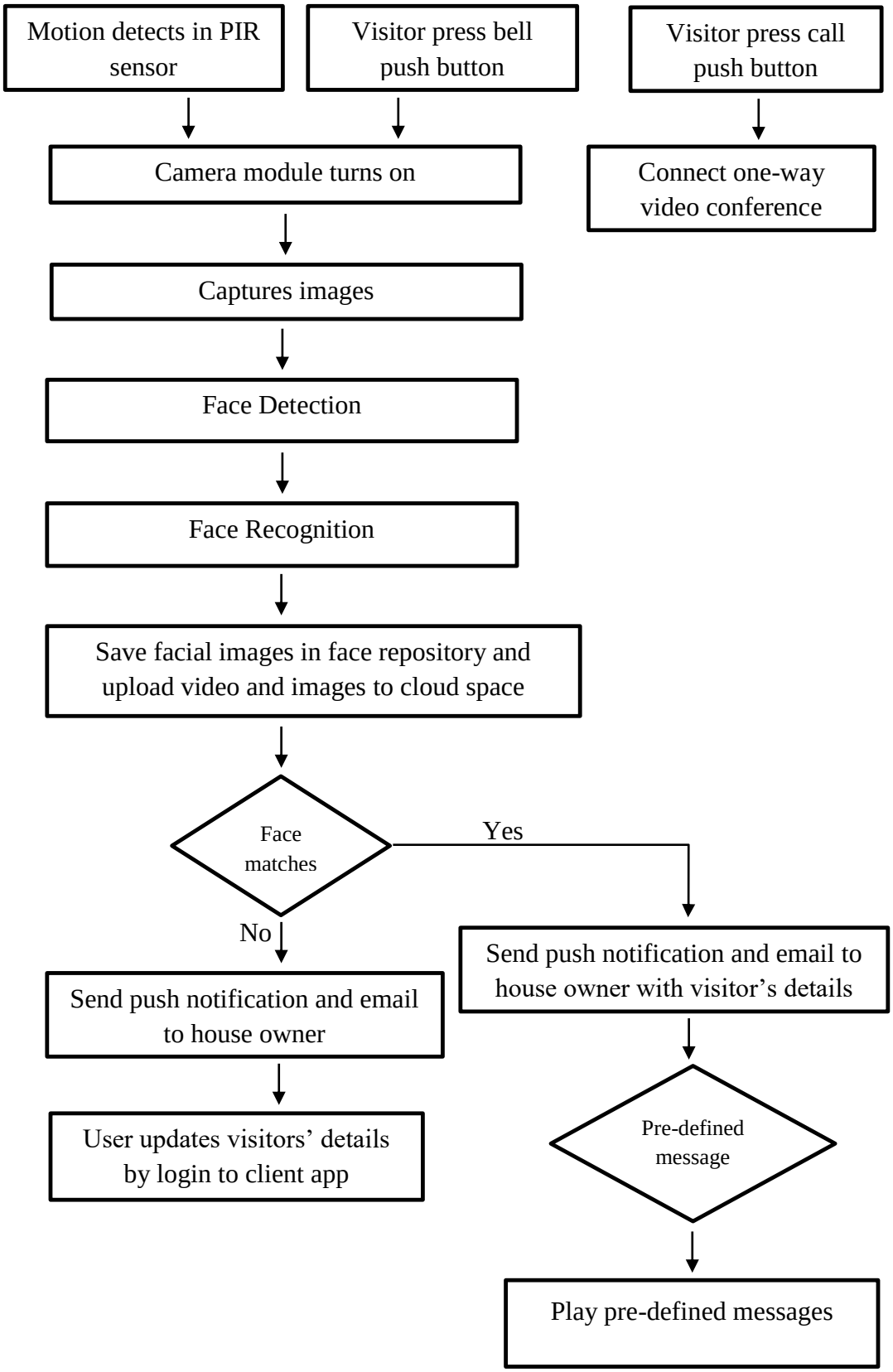

Figure 4.4.1- Process flow

4.5. Users

Within a given time, the facial recognition module of the Smart Doorbell system is capable of identifying multiple visitors. House owner can interact with the system by downloading images, videos, logs and feeding visitors' details into the database.

4.6. Features

- Motion detection
- Visitor's face recognition
- Cloud video and image backups
- 24x7 live video of household environment
- Pre-defined voice message
- Live one-way video communication
- Platform independent
- Low cost

4.7. Summary

Due to its low cost and compact size Raspberry Pi is the perfect solution to build up a security application such as the proposed research project which consists of image processing and PIR sensors. Therefore, Raspberry Pi is selected as the core hardware component of the project. It also provides an interface for each sub module of the system.

Chapter 5

5. Analysis and Design

5.1. Introduction

In the previous chapter a comprehensive introduction to the main hardware component of the proposed research project was given. In this chapter a high level diagram containing the system architecture will be provided. The below diagram shows all Smart Doorbell system modules and each module's connection with other modules of the system.

5.2. Smart Doorbell Architecture

This section stats high level architecture of Smart Doorbell and describe each module.

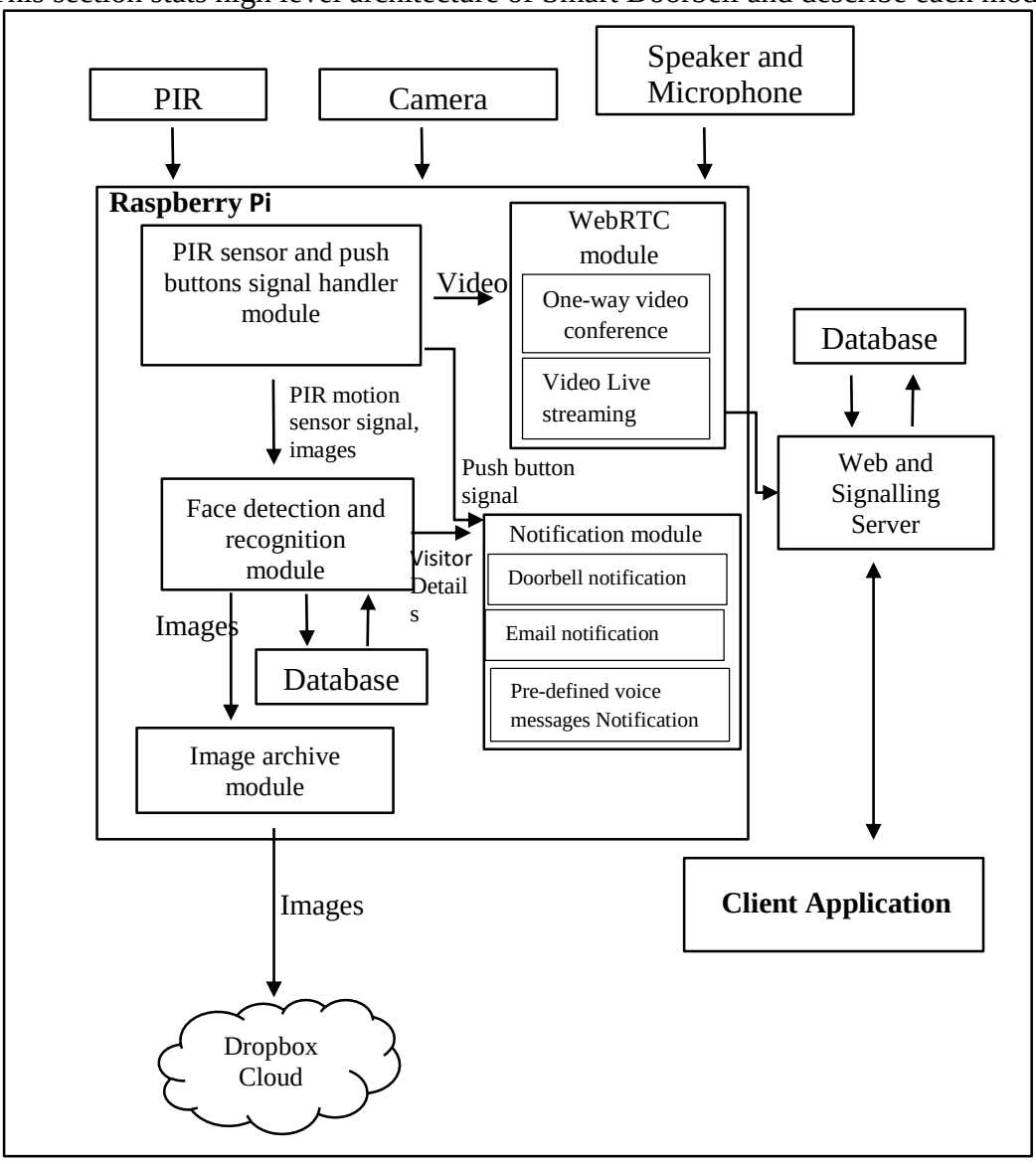

Figure 5.2.1 - High level Architecture of Smart Doorbell

5.3. PIR Sensor and push buttons signal handler module

This module directly interacts with hardware components (PIR sensor and push buttons). When visitor's motion is detected, PIR sensor sends signals to turn camera on. The camera captures images and sends to face detection and face recognition module. Captured images sends to archive module as well.

When visitor presses the doorbell push button, it sends signal to notification module (Doorbell notification sub-module). After visitor press call push button, video data stream is outputted to WebRTC client module.

This module directly deals with Raspberry Pi GPIO pins and Node.js gpio libraries. Most of the inputs are gathered from hardware components and outputs are sent to software module like face detection and face recognition module, image archive module and WebRTC client module.

5.4. Face detection and face recognition module

This software module input images and analyze each frame through face detection algorithm. Face detection is done by using Histogram Oriented Gradients (HOG). If face detects, it sends visitor's facial image to face recognition module and save the images in facial repository.

Face recognition module responsible for identify visitor. This module consists with two sub modules.

1. Train the recognizer – Feed that face data and respective names of each to the recognizer.
2. Recognition - Feed new faces of that people and see if the face recognizer you just trained recognizes them.

If visitor's facial image matches with facial repository image, extract the visitor's name from local pickle (128 face embeddings storage) and sends to notification module (email notification).

Visitor's facial images are sends to image archive module.

5.5. Image archive module

Image archive module collects inputs (images) from face detection and face recognition module. This module responsible for upload those images for Dropbox cloud space by using Dropbox API.

5.6. WebRTC module

WebRTC client module plays major role to establish connection with peers and sent captured video data streams to other end. Collects inputs from PIR sensor and Push buttons signal handler module and output to web/signalling server located in remote location.

5.7. Notification module

Basically this module contents 3 sub modules. Overall this module responsible for notify to house owner and visitor.

1. Doorbell notification – Buzzer ring and indicate visitor's presence
2. Email notification – Send visitor's image via email
3. Pre-defined voice message notification – Pre-defined voice messages to particular users and after their presence it will play automatically.

5.8. Web and signalling server

This consists with web application and signalling server that supports for WebRTC. Web server runs node.js and signalling sever develops using in Socket.io library. Signalling server responsible to send messages between Doorbell and client application.

5.9. Client Application

House owner can view live stream of house hold surrounding, get push notification of visitor's presence, tag unknown visitors, one-way video communication, view visitor logs and set pre-defined voice messages.

5.10. Summary

For development purposes and to refer to clarify the flow the system, it’s necessary to design a diagram containing the system architecture. By studying the diagram any person who reads the report can get a clear and precise idea of the system’s functionalities and the inter-connectivity of each module.

Chapter 6

6. Implementation

6.1. Introduction

In the previous chapter design and analysis of the research project was done. It provides details on each module of the system and how modules interact with each other. In this chapter, implementation effort on all modules are described.

6.2. Implementation of PIR Sensor and push buttons signal handler module

GPIO diagram in Raspberry Pi is shown here. All the wiring between Raspberry Pi and other hardware equipment (PIR sensor and push buttons) are described in the diagram below.

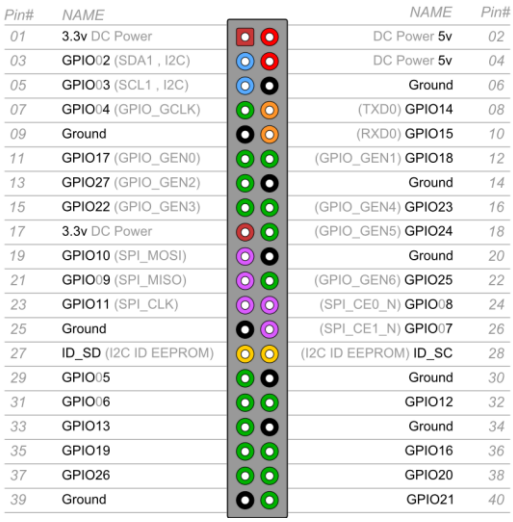

Figure 6.2.1 - Raspberry Pi 3 B+ Model GPIO Structure

6.2.1. PIR Sensor

Firstly, PIR sensor is mounted into Raspberry Pi using GPIO pins. PIR sensor has 3 pins and those are labelled as VCC, GND and OUT.

- Connect the PIR sensor's pin labelled VCC to the 5V pin (pin no 2) on the Raspberry Pi. This provides power to the PIR sensor.
- Connect the one labelled GND to a ground pin on the Pi (pin no 6). This completes the circuit.

- Connect the one labelled OUT to any numbered GPIO pin on the Pi. In this example, we have chosen GPIO 4 (pin no 7). The OUT pin will output a voltage when the sensor detects motion. The voltage will then be received by the Raspberry Pi.

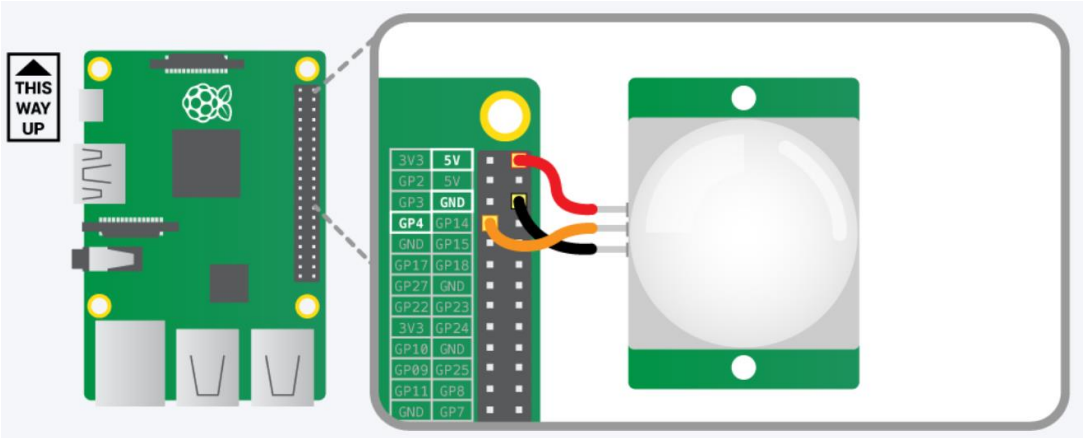

Figure 6.2.2 - Mount PIR sensor into Raspberry Pi

Node server is configured in Raspberry Pi device locally. Onoff module is used to read the PIR sensor.

```
var gpio = require('onoff').Gpio;
var pir = new gpio(4, 'in', 'both');
```

Figure 6.2.3 - Code snippet of onoff module and PIR sensor declare

Here, the first line indicates importing onoff module to node script. GPIO constructor has 3 parameters. In an orderly manner, those are pin, direction and edge. Use GPIO 4 (pin no 7) to communicate. The second parameter ‘direction’ refers to check whether the pin is used as an input or output. Here node script always listens to PIR sensor due to passing ‘in’ keyword a second parameter. Third parameter ‘edge’ has three values (rising, falling and both). In that case node script needs both pins to get high and low. Due to this reason ‘both’ has to be passed as the third parameter.

```
pir.watch(function(err, value) {
    if (value == 1) {
      //motion detected
    } else {
      //motion has stopped
    }
});
```

Figure 6.2.4 – Code snippet of watch function

Function ‘watch’ in onoff module tries to find out whether motion is detected or not. This function has two parameters (error and value). When value gets ‘1’, it indicates that motion is detected and when the value gets ‘0’ that indicates the motion has stopped. When motion detects, web camera will start and take 7 images of the visitor.

6.2.2. Push buttons (Tactile switch)

Throughout this solution, two push buttons are used. They are call button and doorbell button. Call button is attached with GPIO 18 (pin no 12) and GND (pin no 14). Doorbell button is attached with GPIO 23 (pin no 16) and GND (pin no 20).

```
var callButton = new gpio(12, 'in', 'both');
var bellButton = new gpio(26, 'in', 'both');
```

Figure 6.2.5 - Call button and Doorbell button GPIO declare

In here both tactile switches pass ‘in’ value for direction parameter and ‘both’ value for edge parameter. When call button is pressed ‘callButton.watch’ callback function triggers and will send message to signaling server via TCP socket connection. When visitor presses bell button, the doorbell rings and at the same time email & push notification are sent to the house owner.

6.3. Implementation of face detection and face recognition module

Face detection is locating faces in an image or a video clip. Verifying or identifying a person in an image or video clip is known as face recognition. This module is entirely based on OpenCV, Python, dlib and face_recognition modules. Face detection and face recognition module’s structure is illustrated in figure 6.6.

Face recognition is mainly divided into 3 steps.

1. Locating visitor’s face
2. Encoding faces
3. Finding the visitor’s name from the encoding

dataset - contains face images of visitors and its organized into sub-directories based on their respective name

unknown_visitors - contains face images of unknown visitors

visitors - 7 images of any visitor will be stored in this directory temporary. If user marked the images as a known person, those images are deleted. If the user is unable to identify, move to unknown_visitors directory

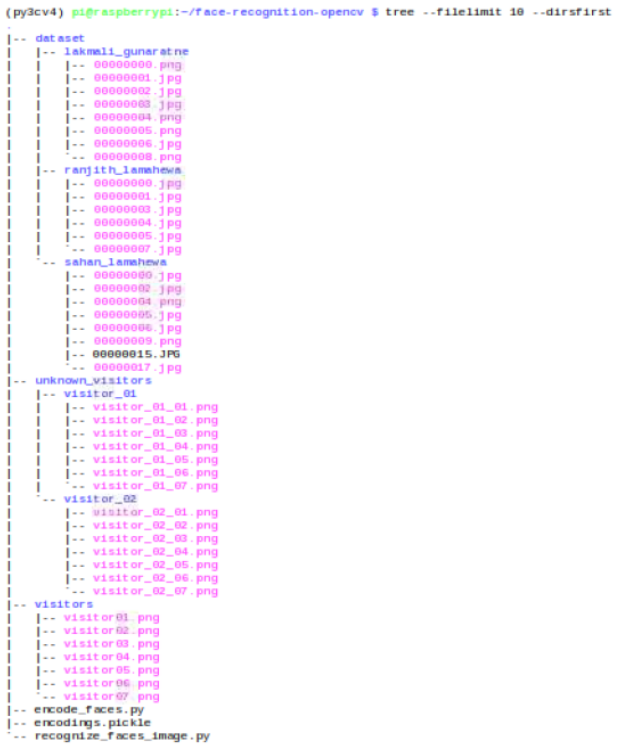

Figure 6.3.1 - Folder and file structure of face recognition and face detection module

6.3.1. Locating visitor's face

Histogram of Oriented Gradient (HOG) algorithm is used to locate visitor's face in a given image. The method is called 'face_locations' and is used to locate a visitor's face in a given image. In this method 2 parameters are passed (image and face recognition algorithm) and detect the (x, y)-coordinates of the corresponding bounding boxes.

```
boxes = face_recognition.face_locations(rgb, model="hog")
```

Figure 6.3.2 - Code snippet of find faces in an given image

6.3.2. Encoding faces

In this module, coordinate of the face located in the given visitor's image are obtained and the facial embedding is extracted into 128 decimal array. Simply, the face extractions are stored as 128 decimal numbers.

```
encodings = face_recognition.face_encodings(rgb, boxes)
```

Figure 6.3.3 - Find face location

The method is called as 'face_encodings' and is used to generate 128 face embedding.

6.3.3. Finding the visitor's name from the encoding

In this step, previously generated encoding of a given visitor's image compares with known visitors' encoding. All the known visitors' encoding are stored in 'encodings.pickle' file. If a match is found out, then it will return visitor's name otherwise it will return as 'unknown'.

When camera captures visitor's 7 images, they are stored in the visitor' directory. Then recognize_faces_image.py script will run and check whether the visitor is already in the known visitor's database. If the visitor is a known person, simply returns the name. Otherwise, it will return unknown and move unknown visitor's images into unknown visitors folder by creating a new subdirectory (ex : visitor_01, visitor_02). Last two digits of the folder maintains in a separate file. That file holds one number and when an unknown visitor detects, it will increment by 1 and make subdirectory name with the number prefixed with 'visitor_'. Through the house owner's app (client app) house owner can see unknown visitors' faces listed in unknown_visitors subdirectory. Then house owner will be able to tag the unknown visitor. Then images in the unknown visitor's subdirectory will moved to 'dataset' directory with the tagged name by the house owner. During midnight, 'encode_faces.py' script will run and known visitors images respective to vistor's name sub-folders (sahan_lamahewa, lakmali_gunaratne) with the extracted facial embeddings and are stored in encording.pickle file. This pickle file stores consist of all face images in 'dataset' directory as 128 facial embeddings decimal numbers respective to visitor's name.

6.4. Image archive module

Dropbox API is used for image archive purposes. This module will be scheduled to run at the end of the day (11.00 pm).

Node	Description	Method	Parameter
/oauth2/authorize	This starts the OAuth 2.0 authorization flow. This isn't an API call—it's the web page that lets the user sign in to Dropbox and authorize the app. After user decides whether or not to authorize the app, user will then be redirected to the URI specified by the redirect_uri [61].	Get	response_type - (String) The grant type requested, either token or code. client_id - (String) The app's key, found in the App Console. [61] Example - https://www.dropbox.com/oauth2/authorize?client_id=<APP_KEY>&response_type=code
/oauth2/token	This endpoint only applies to apps using the authorization code flow. An app calls this endpoint to acquire a bearer token once the user has authorized the app. Calls to/oauth2/token has to be authenticated using the apps's key and secret [61].	Post	Code - (String) The code acquired by directing users to /oauth2/authorize?response_type=code. grant_type - (String) The grant type, which must be authorization_code. client_id - (String) If credentials are passed in POST parameters, this parameter should be present and should be the app's key (found in the App Console). client_secret - (String) If credentials are passed in POST parameters, this

			parameter should be presented and should be the app's secret.
/create_folder	Create a folder at a given path.		Path - (String) (pattern="(/(.|[\r\n])*)|(ns:[0-9]+(/.*)?)") Path in the user's Dropbox to create. Autorename - (Boolean) If there's a conflict, have the Dropbox server try to auto rename the folder to avoid the conflict. The default for this field is False.
/upload	Create a new file with the contents provided in the request.		path - (String) (pattern="(/(.|[\r\n])*)|(ns:[0-9]+(/.*)?)|(id:.*)") Path in the user's Dropbox to save the file. Mode - (WriteMode) Selects what to do if the file already exists. The default for this union is add. (add, overwrite, update)

Table 6.1 - Dropbox API methods

6.5. WebRTC module

WebRTC module is used to facilitate the one-way video conference and video streaming capabilities.

The core of the WebRTC communication is peer to peer and to establish this communication requires coordination between peers. At that point web server/signaling server bridges the gap. This facilitate two or more WebRTC capable devices to find each other, transfer contact details and exchange session data and finally transfer data streams between peers.

6.5.1. One-way video conference

Main tasks of one-way video conference

- User connection
- Signaling
- Exchange network information
- Exchange media sessions
- Exchange media streams

User Connection

Initially two peers need to connect with each other. In this solution, web application is used and both peers access web pages which resides in that application. Then web pages will be able to connect to server via websockets. Each peer browsers can share the signaling server. For the security perspective, use unique id to communicate between peers then if an intruder tries to connect to web server via a web page, they won't be able to communicate without knowing the unique id. When caller sends the unique URL (URL with unique id) to the callee, and once they both have this page open at the same time, both of them can establish the connection.

Signaling

After establishing the connection between two peers, any message can be shared among them. These messages are used to control the WebRTC tasks. This signaling is out of the WebRTC scope.

Exchange Network Information

This step is commonly known as 'finding candidates.' This step is responsible for exchanging their network information with another peer on how they can be connected to the network. Finally, each peer should be mapped to a directly accessible from the network interface and port. Each peer located behind the router and must use Network Address Translation (NAT) to connect to internet. Due to some firewall rules, certain ports and connections may be blocked. Finding a path to connect this type of routers is called NAT traversal and it is crucial to establish connection in WebRTC. Session

Traversal Utilities for NAT (STUN) servers is used to connect with public internet and collecting network information from peers behind restricted firewalls.

Exchange media session

Session Description Protocol (SDP) is used to exchange media related data (codec, resolution and bitrate). Basically, exchange type and formats of media (audio and video).

Exchange media streams

In this step, browser can directly send media streams between peers.

Media Stream API

Media Stream API is used to access streams of local media devices like camera and microphone. The getUserMedia() method is the interface to access those media streams. MediaStream objects can then be output in two key ways. First, they can be used to render output into a MediaElement such as a <video> or <audio> element. Secondly, they can be used to send to an RTCPeerConnection, which can then send this media stream to a remote peer.

RTCPeerConnection API

The RTCPeerConnection API is the major component of the peer-to-peer connection between WebRTC enabled peers. RTCPeerConnection constructor is used to create RTCPeerConnection object and will pass configuration parameter. The configuration variable contains array of STUN servers' information (URL, username and password) which is named as iceServers. The application of peerconnection object is slightly different according to the client, whether you are the caller or the callee.

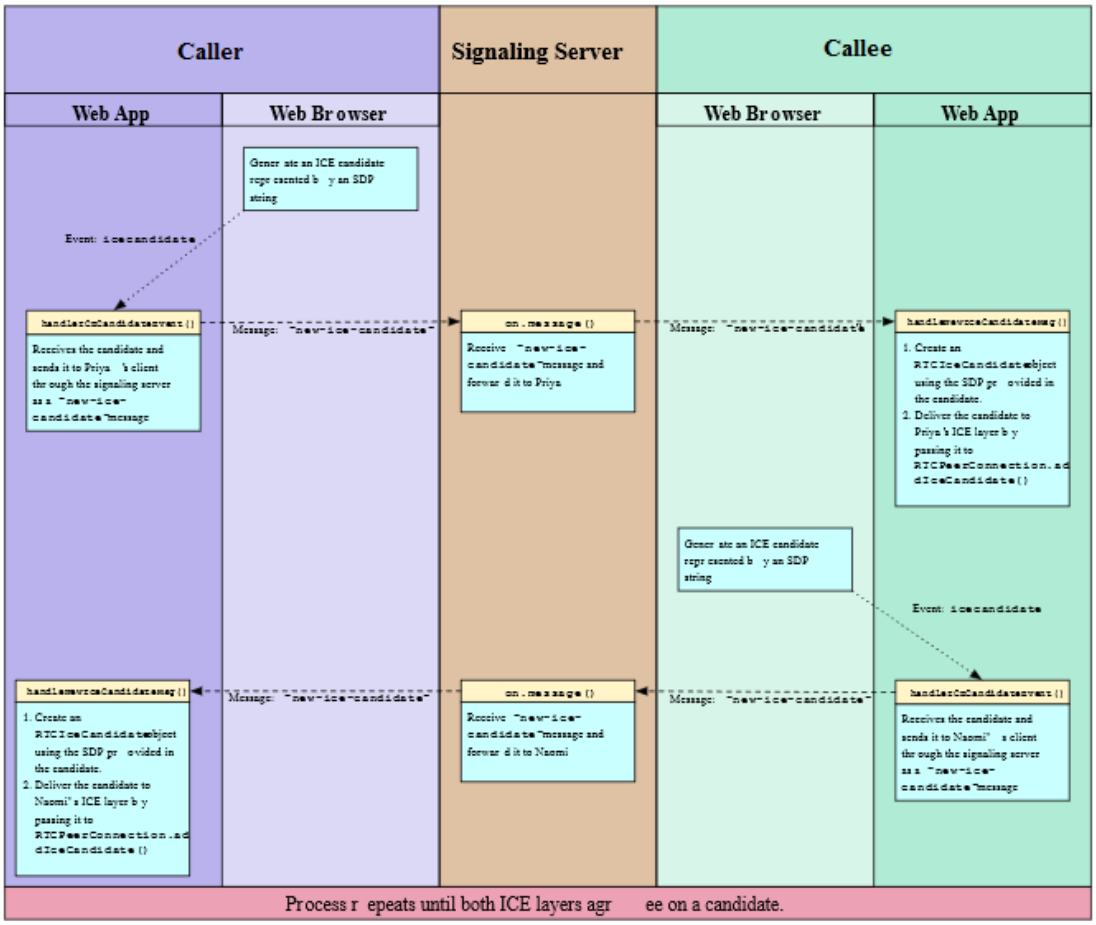

Figure 6.5.1 - ICE candidate exchange process

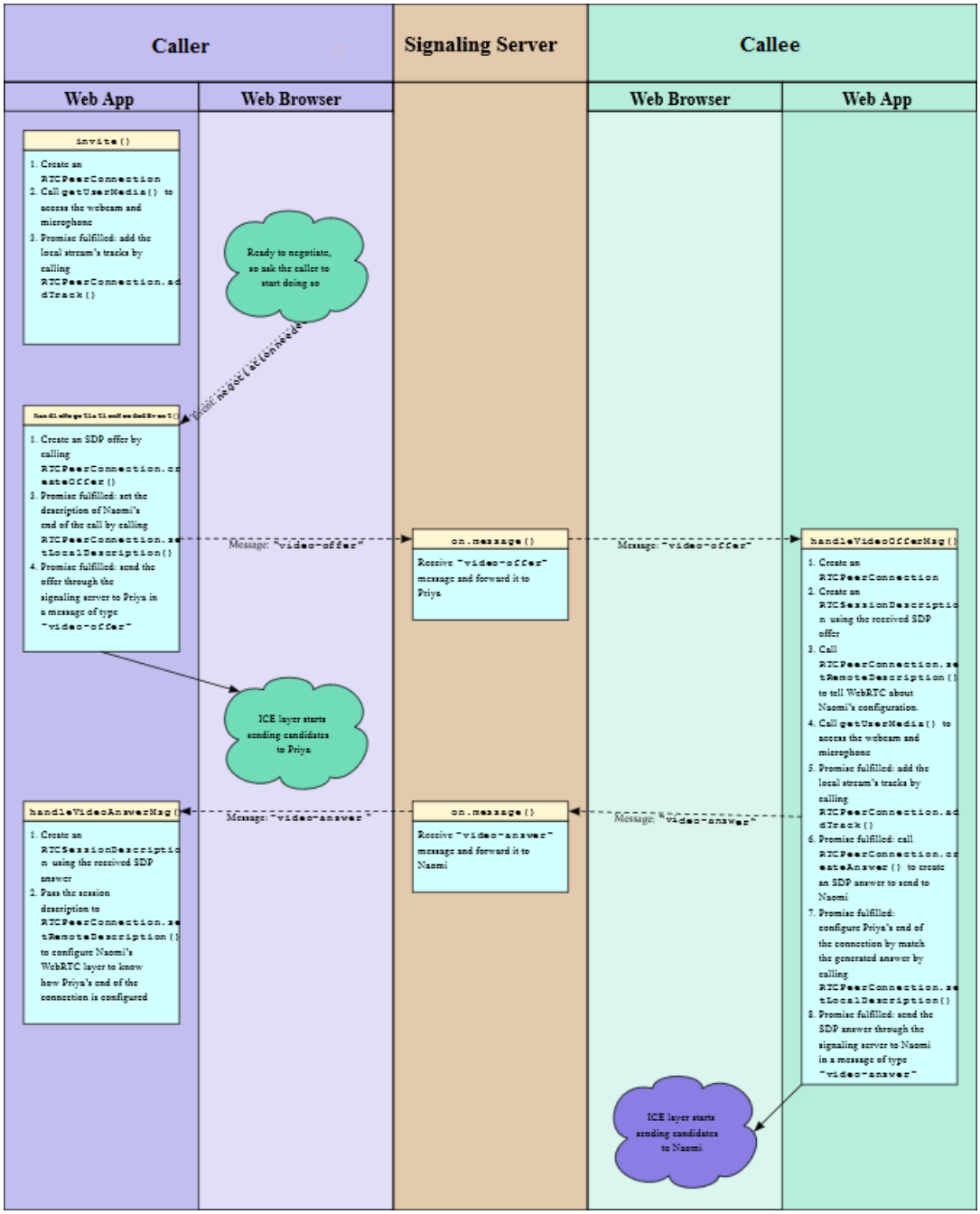

Figure 6.5.2 - Signalling transaction flow

6.5.2. Video Live Streaming

In this sub module, web camera media stream is accessed via getUserMedia() API and directly bind stream into video element as shown in Figure 6.9.

```
var constraints = { audio: false, video: { width: 1280, height: 720 } };

navigator.mediaDevices.getUserMedia(constraints)
.then(function(mediaStream) {
  var video = document.querySelector('video');
  video.srcObject = mediaStream;
  video.onloadedmetadata = function(e) {
    video.play();
  };
})
.catch(function(err) { console.log(err.name + ": " + err.message); });
```

Figure 6.5.3 - Live video streaming code snippet

Then HTML canvas element is used to draw frames in certain intervals that are playing in video element. After, converts images drawn in canvas to data URI containing a representation of the image in the format image/webp and send this to signaling server via socket connection. When house owner wants to watch the live video stream through the client app, he/she will click stream button in client app and after couple of seconds live stream video will play in the client app.

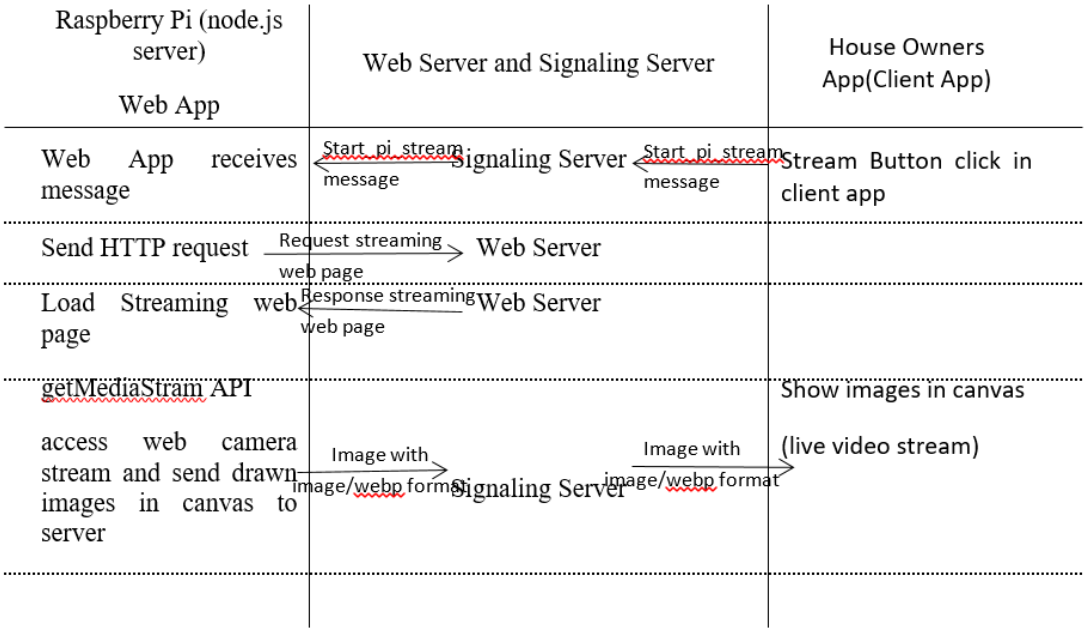

Figure 6.5.4 - Streaming Flow

6.6. Notification module

Notification module contains 3 sub modules.

1. Doorbell notification - When visitor presses the doorbell push button, buzzer rings
2. Email Notification - When visitor is identified, send email to house owner
3. Predefined Voice message to identified visitors - When visitor is identified, play predefined voice messages

6.7. Signaling Server

Mainly, signaling server is the message distributor among the house owner's application and the Raspberry Pi hardware. Node is used to build this signaling server. 'Socket.io' javascript library is used throughout the solution to build websockets among hosts.

6.8. Client App (House Owner's app)

Client application is built using Apache Cordova PhoneGap. Socket.io is used to make communication with web/signaling server. Front end is done using Semantic-ui.

6.9. Summary

To summarize the implementation of the system up to now, the PIR Sensor, camera module and push buttons signal module is already finished. Furthermore, in face detection and face recognition module of the project, up to face detection is done and facial recognition sub module remains to be developed before the final report.

Chapter 7

7. Evaluation

7.1. Introduction

In the previous chapter, how the implementation of the proposed system was extensively addressed and discussed by each core module. In this chapter some test scenarios were performed on the core functionalities of the system. Mainly facial detection and facial recognition module and making a call to a handheld device via WebRTC technology was tested. The collected test results are laid out below.

7.2. Functional testing

Some basic functional tests on image detection & recognition and WebRTC video conferencing call modules were done. The findings of the testing scenarios are listed below.

7.2.1. Facial detection & recognition

This function can identify known visitors. For testing purposes, to identify known faces, algorithm was prior trained after inputting several training data. When the facial recognition algorithm ran after inputting images with different lighting conditions and other scenarios such as identifying an unknown person (algorithm is not trained early to identify any person under this category), following test results were gathered.

1. Training Datasets

The following 16 images (dataset) were trained as dataset 1 (Sahan_Lamahewa) and dataset 2 (Lakmali_Gunaratne) and the face recognition results outputs shown below were all derived using these two trained datasets.

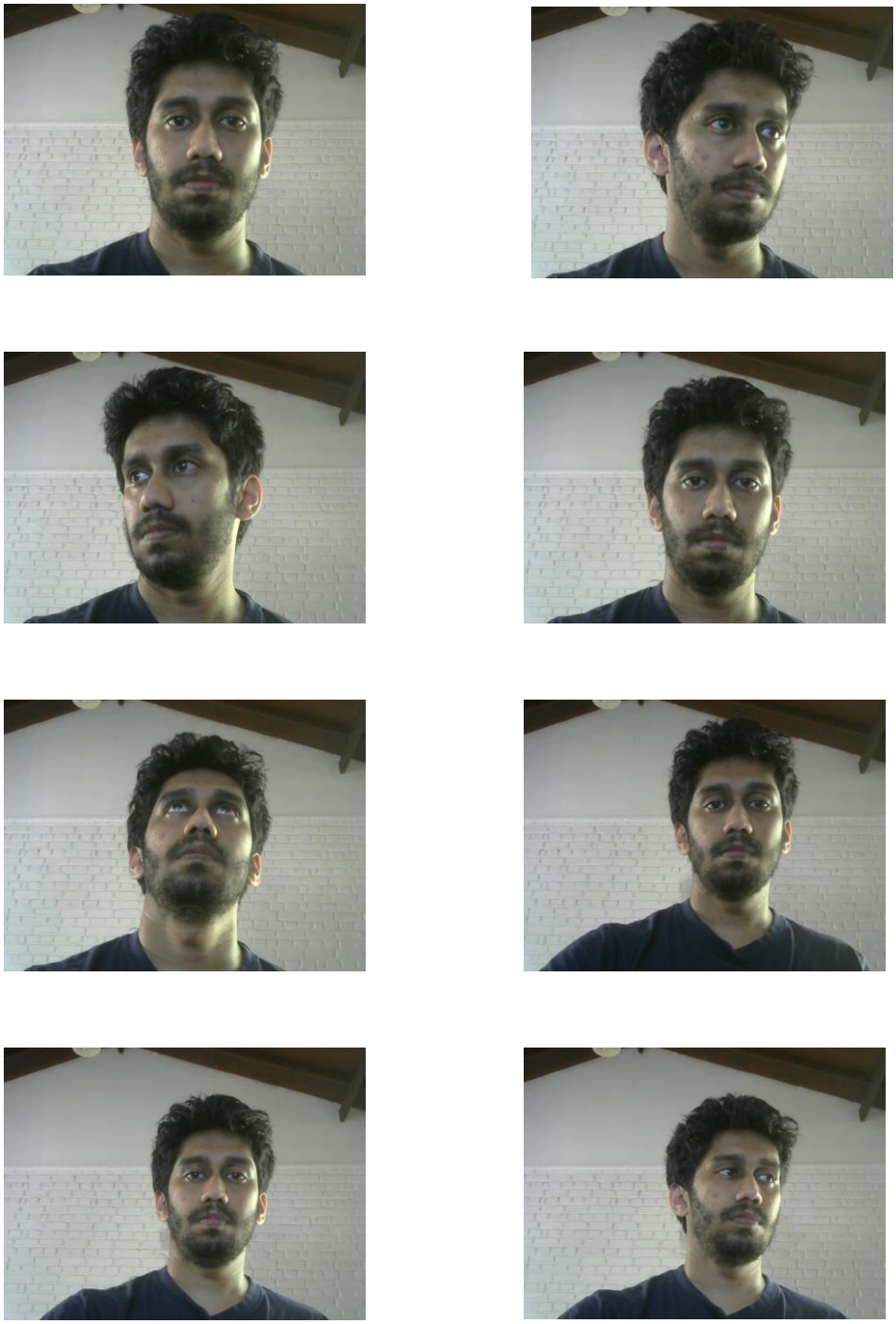

Figure 7.2.1 - Trained dataset1 (Sahan Lamahewa)

Figure 7.2.2 - Trained dataset2 (Lakmali Gunaratne)

2. Image recognition in low light condition

As shown in the figure below, the facial recognition algorithm worked as expected and recognized faces even in a low lighting condition.

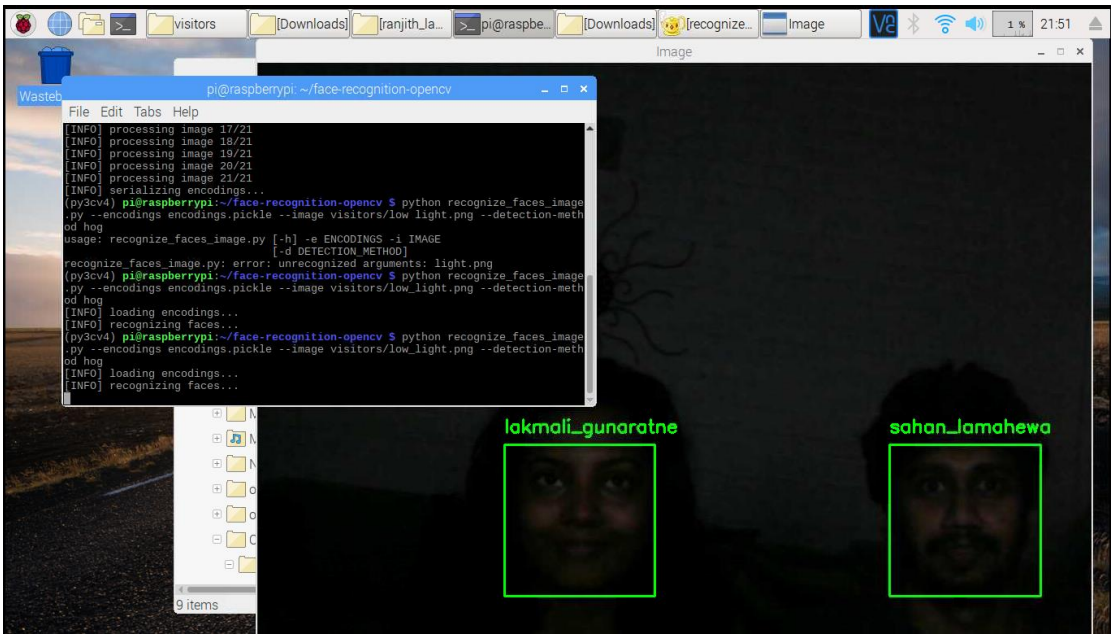

Figure 7.2.3 - Image recognition in low light condition

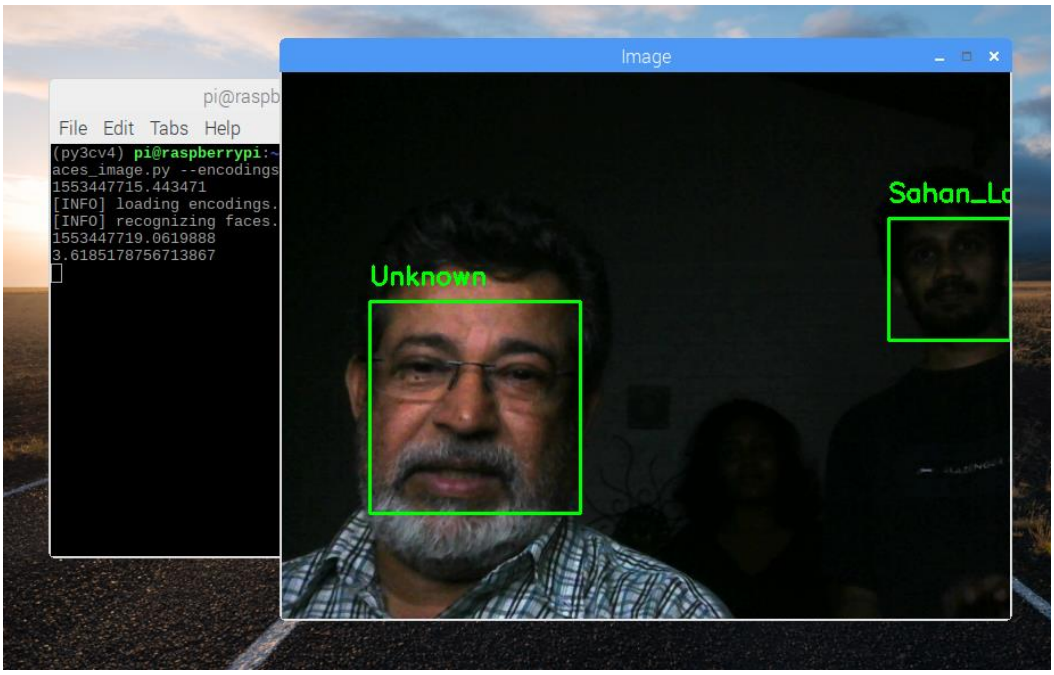

Figure 7.2.4 – Image recognition in low light condition with different distance levels

In here dataset 1 is identified as Sahan Lamahewa as expected although the face recognition algorithm has not been able to identify dataset 2 as Lakmali Gunaratne due to distance and bad lighting conditions. Overall 3.618 seconds were taken to analyse and output the results.

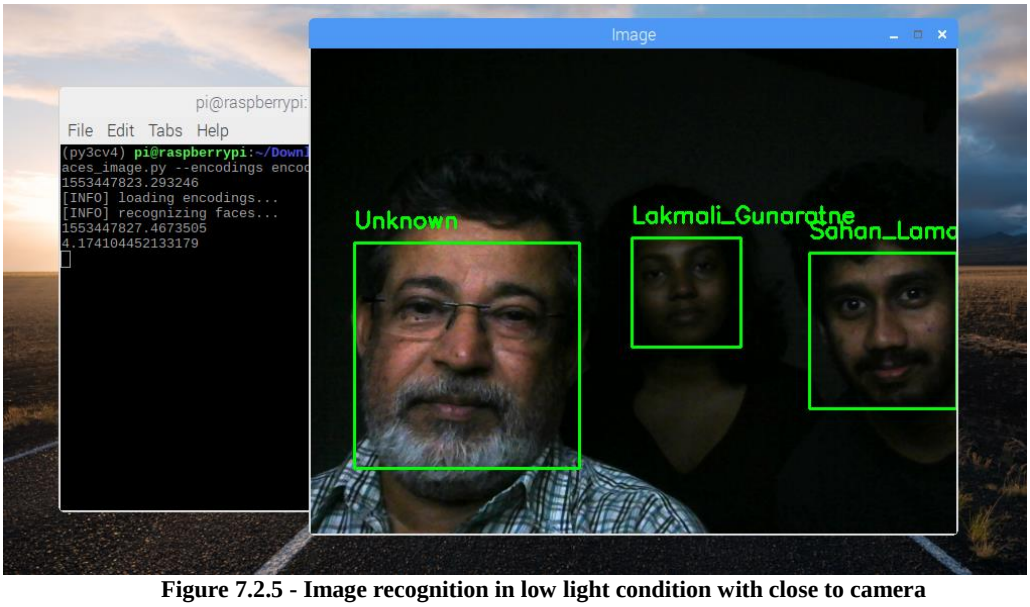

Figure 7.2.5 - Image recognition in low light condition with close to camera

In here dataset 1 is identified as Sahan Lamahewa and dataset 2 is identified as Lakmali Gunaratne as expected since all the subjects were within similar distance from the camera and also the subjects were under bad lighting conditions. The person which is

not trained to be identified by the facial recognition algorithm using training data is identified as ‘Unknown’. Overall 4.174 seconds taken to analyse and output results.

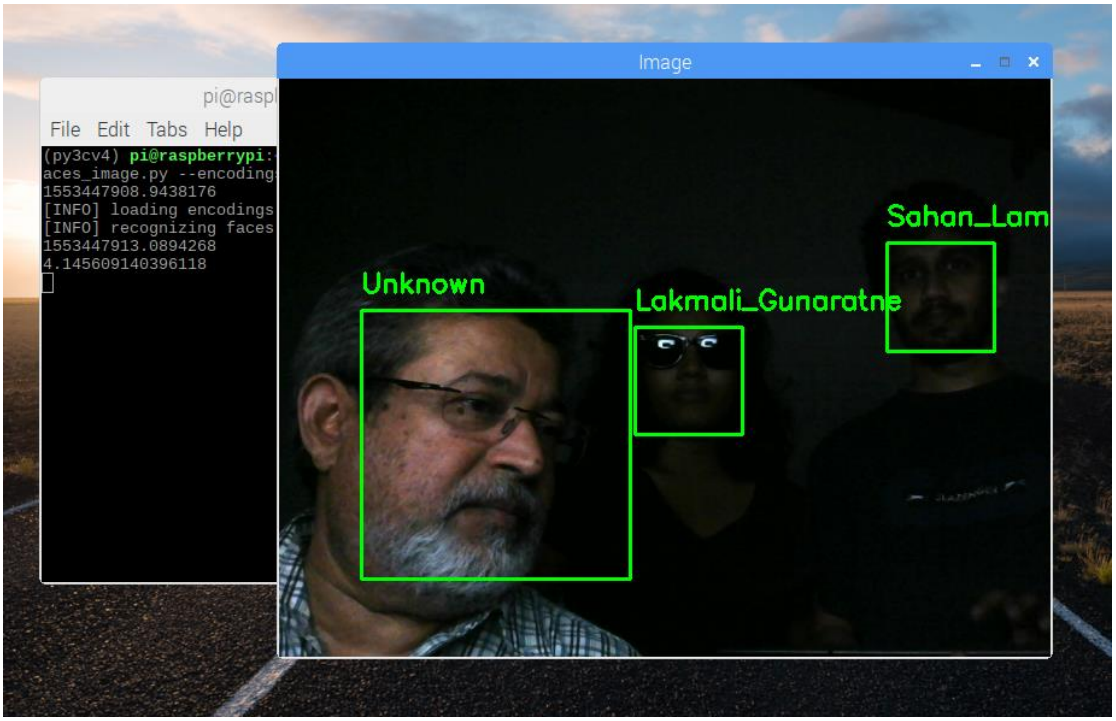

Figure 7.2.6 - Image recognition in low light condition with close to camera and wearing accessories

In here dataset 1 is identified as Sahan Lamahewa and dataset 2 is identified as Lakmali Gunaratne as expected but here the person represented in dataset 2 is wearing sun glasses. Like the fig. 7.2.5 in this scenario also the subjects were within similar distance from the camera and also the subjects were under bad lighting conditions. The person which is not trained to be identified by the facial recognition algorithm using training data is identified as ‘Unknown’. Overall 4.146 seconds taken to analyse and output results.

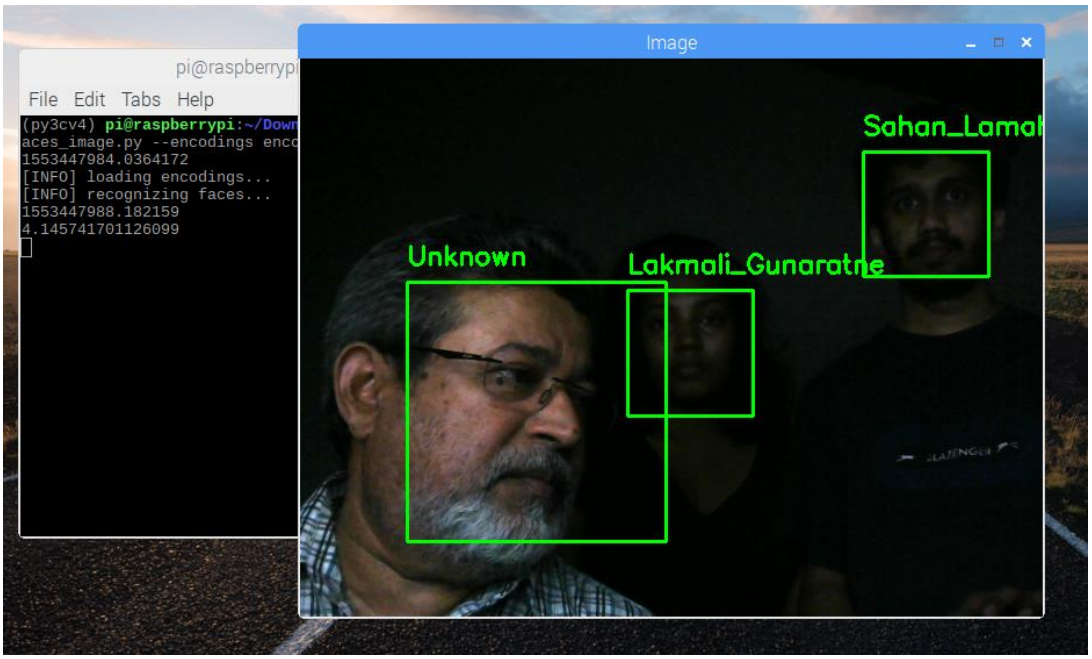

Figure 7.2.7 - Image recognition in low light condition with close to camera and one subject facing away from camera

In here dataset 1 is identified as Sahan Lamahewa and dataset 2 is identified as Lakmali Gunaratne as expected. Like the fig. 7.2.5 in this scenario also the subjects were within similar distance from the camera and also the subjects were under bad lighting conditions. The person which is not trained to be identified by the facial recognition algorithm using training data is facing away from the camera and yet it is identified as 'Unknown'. Overall 4.146 seconds taken to analyse and output results.

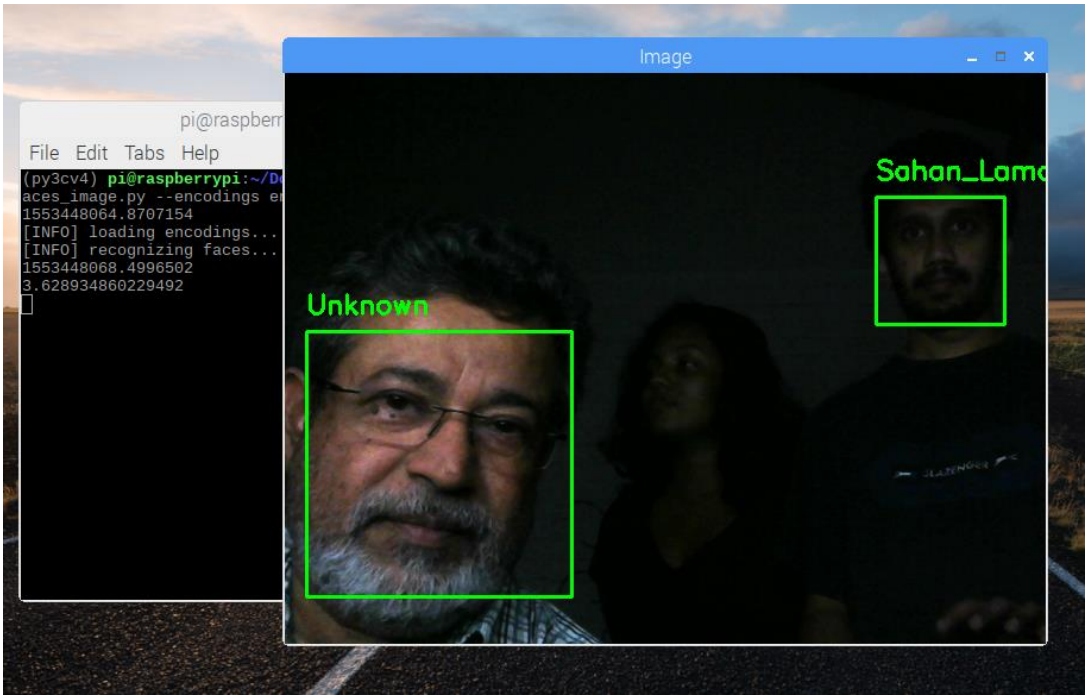

Figure 7.2.8 - Image recognition in low light condition with close to camera and one trained subject facing away from camera

In this scenario dataset 1 is identified as Sahan Lamahewa and dataset 2 is not identified as Lakmali Gunaratne as expected. The reason behind the facial recognition algorithm not being able to identify the person represented by the dataset 2 could be because the person is facing away from the camera and the algorithm is unable to properly extract the matching characteristics of the facial features. Like the fig. 7.2.5 in this scenario also the subjects were within similar distance from the camera and also the subjects were under bad lighting conditions. The person which is not trained to be identified by the facial recognition algorithm using training data is identified as ‘Unknown’. Overall 3.629 seconds taken to analyse and output results.

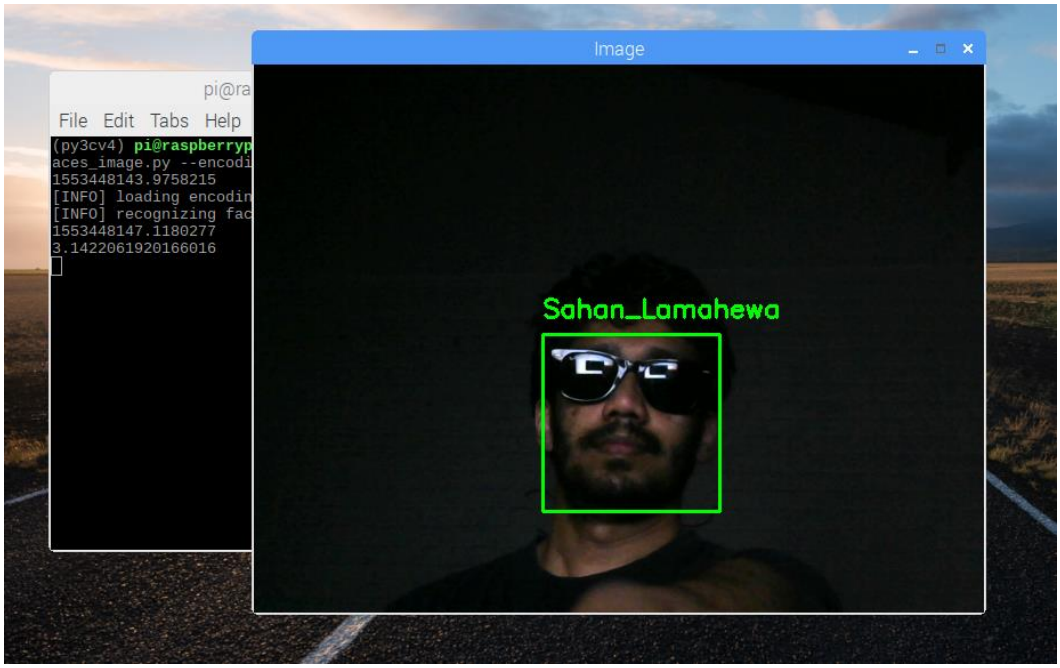

Figure 7.2.9 - Image recognition when the trained subject is directly facing the camera under low light and wearing accessories

In this scenario dataset 1 is identified as Sahan Lamahewa as expected. Regardless of the poor lighting condition and the sunglasses, the person is successfully identified due the fact the subject is directly facing the camera. Overall 3.142 seconds taken to analyse and output results.

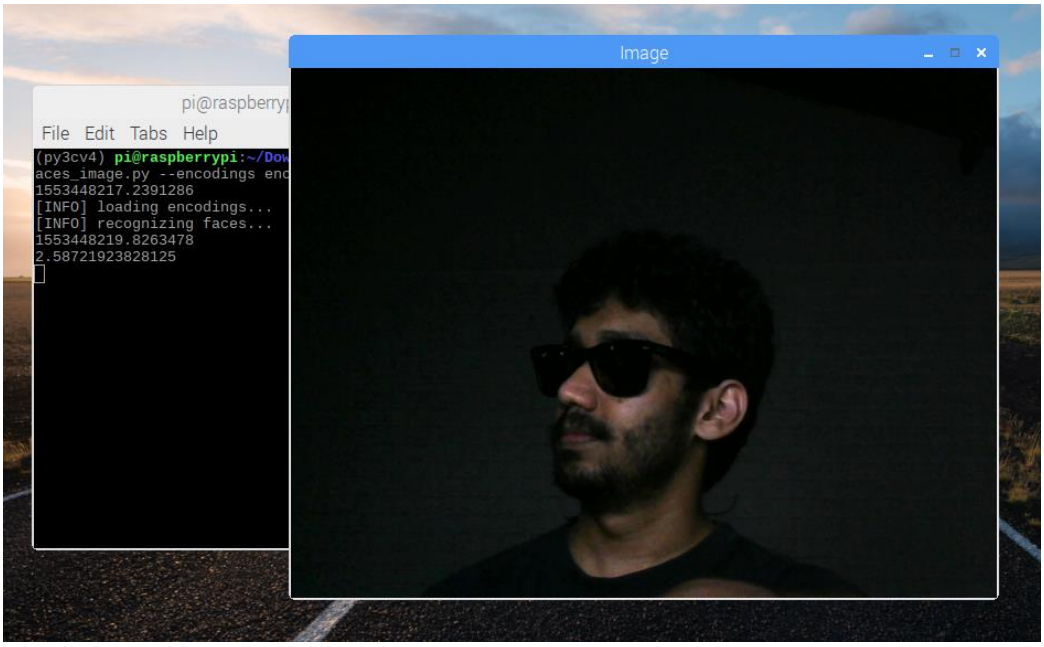

Figure 7.2.10 - Image recognition when the person is facing the camera in forty-five-degree angle under low light and wearing accessories

In this scenario dataset 1 is not identified as Sahan Lamahewa. The person is facing the camera in a forty-five-degree angle and is staying within a close proximity to the camera under low light wearing a sunglass. Apart from the low lighting condition and the facial accessory, clearly due to the angle of the face, the facial recognition algorithm

is unable to properly extract the matching facial features and identify the person. Overall 2.587 seconds taken to analyse and output results.

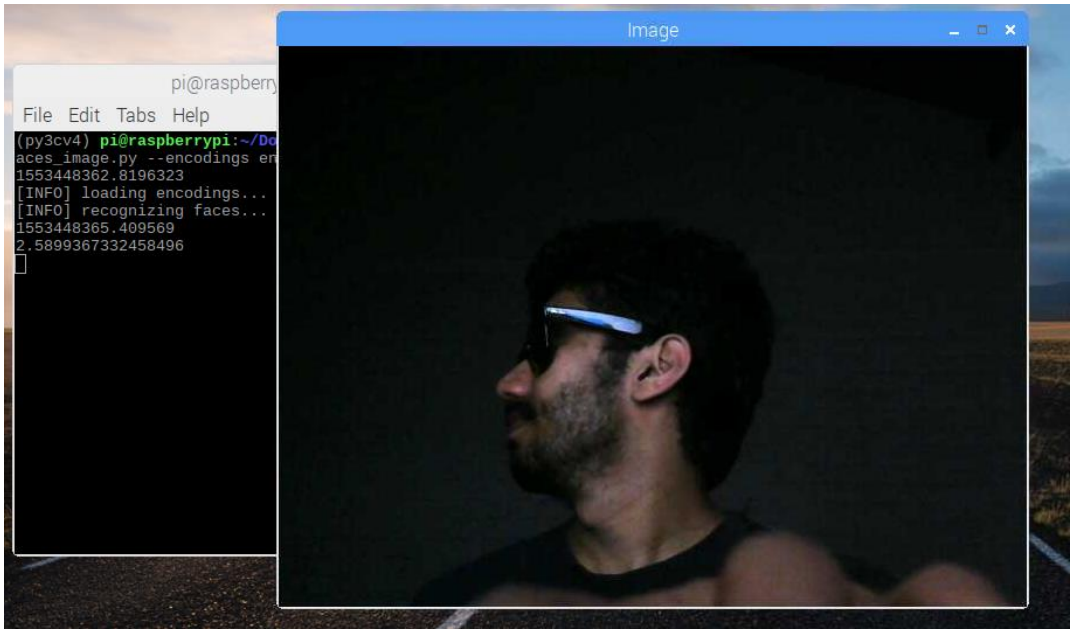

Figure 7.2.11 - Image recognition when the person is facing the camera in ninety-degree angle under low light and wearing accessories

In this scenario dataset 1 is not identified as Sahan Lamahewa. The person is facing the camera in a ninety-degree angle and is staying within a close proximity to the camera under low light wearing a sunglass. Similar as shown in figure 7.2.10, apart from the low lighting condition and the facial accessory, clearly due to the angle of the face, the facial recognition algorithm is unable to properly extract the matching facial features and identify the person. Overall 2.587 seconds taken to analyse and output results.

3. Image recognition in under light condition

The capability of the facial recognition algorithm was tested in a considerable lighting condition. Faces were properly identified.

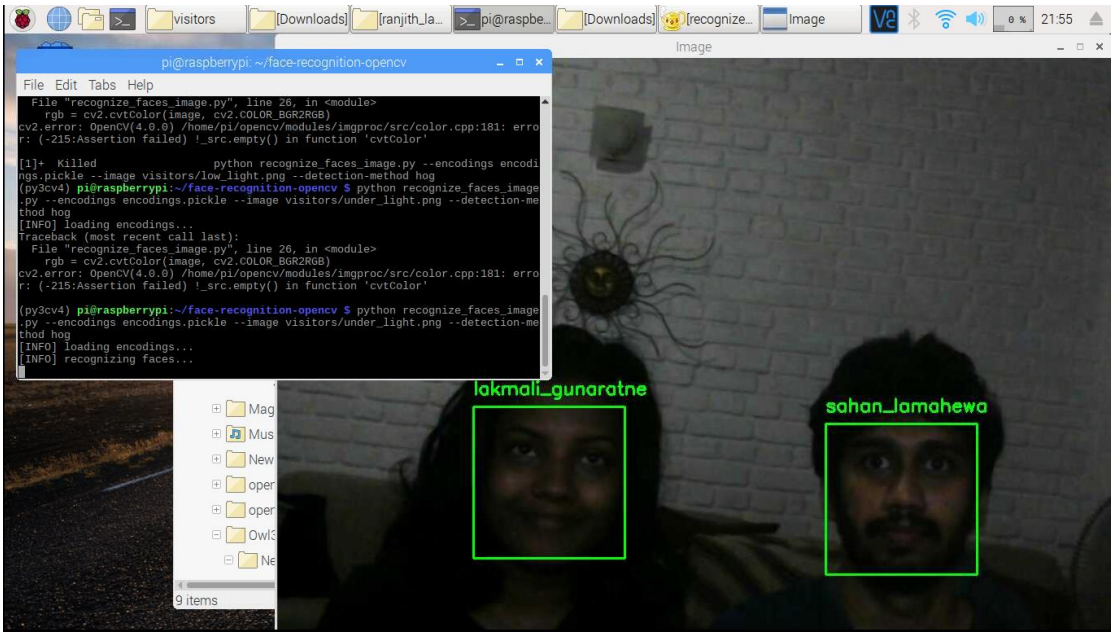

Figure 7.2.12 - Image recognition in under light condition

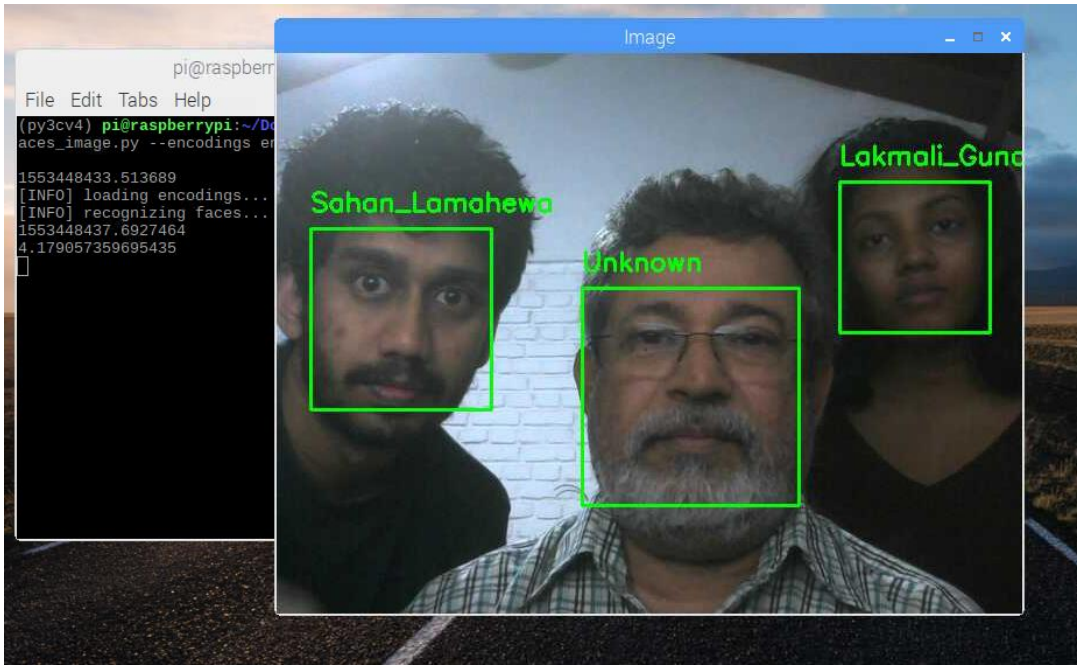

Figure 7.2.13 - Image recognition in under lighting condition with close to camera

In here dataset 1 is identified as Sahan Lamahewa and dataset 2 is identified as Lakmali Gunaratne as expected since all the subjects were within similar distance from the camera and also the subjects were under lighting conditions. The person which is not

trained to be identified by the facial recognition algorithm using training data is identified as ‘Unknown’. Overall 4.179 seconds taken to analyse and output results.

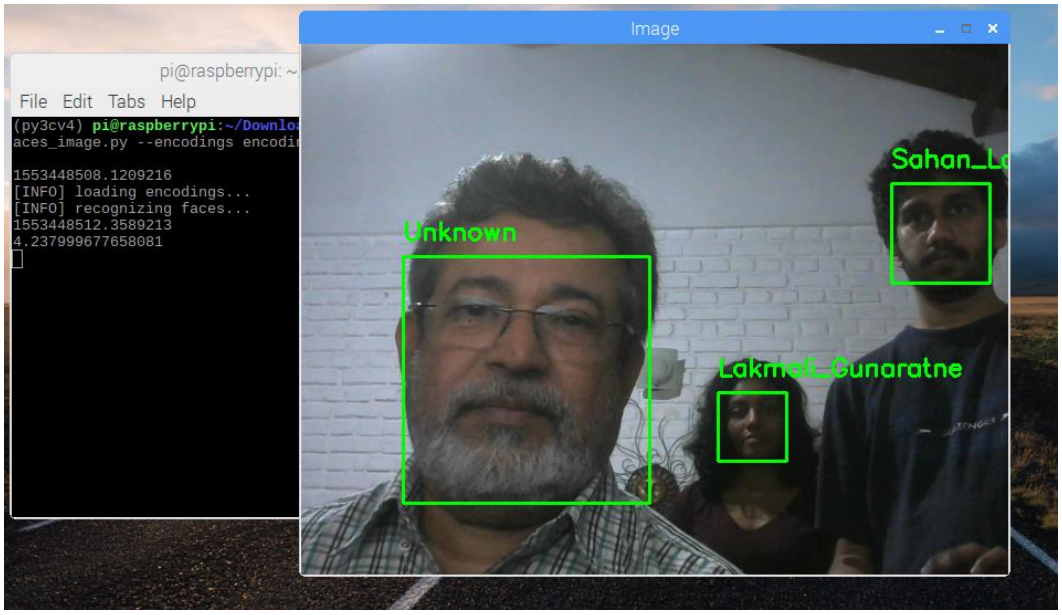

Figure 7.2.14 - Image recognition in under lighting condition with different distance levels

In here dataset 1 is identified as Sahan Lamahewa as expected although the face recognition algorithm has been able to identify dataset 2 as Lakmali Gunaratne even far away from the camera. Overall 4.238 seconds were taken to analyse and output the results.

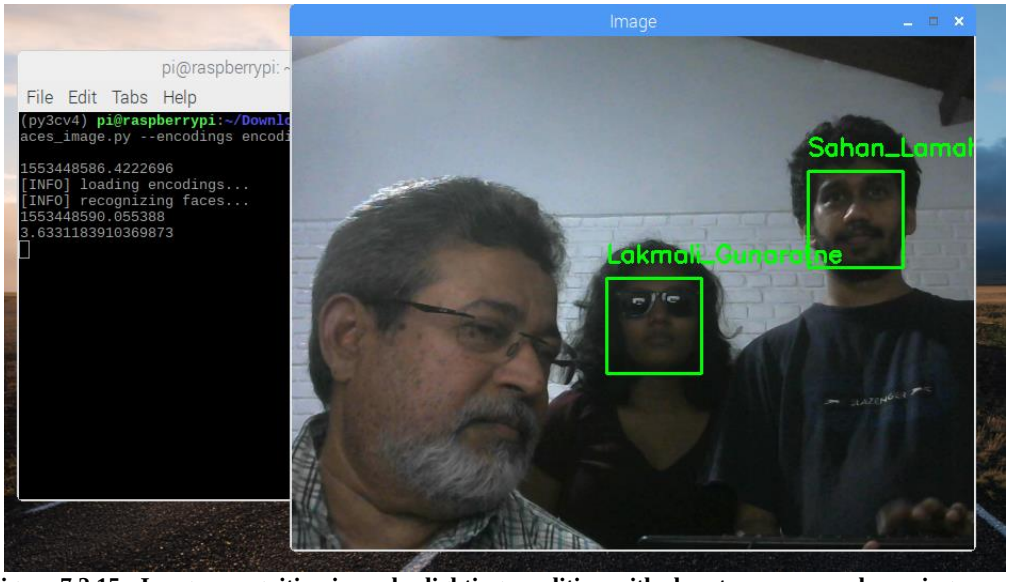

Figure 7.2.15 - Image recognition in under lighting condition with close to camera and wearing accessories

In here dataset 1 is identified as Sahan Lamahewa and dataset 2 is identified as Lakmali Gunaratne as expected but here the person represented in dataset 2 is wearing sunglasses. Like the fig. 7.2.5 in this scenario also the subjects were close to the camera and also the subjects were under lighting conditions. The person which is not trained to be identified by the facial recognition algorithm using training data is not identified as 'Unknown'. Overall 3.633 seconds taken to analyse and output results.

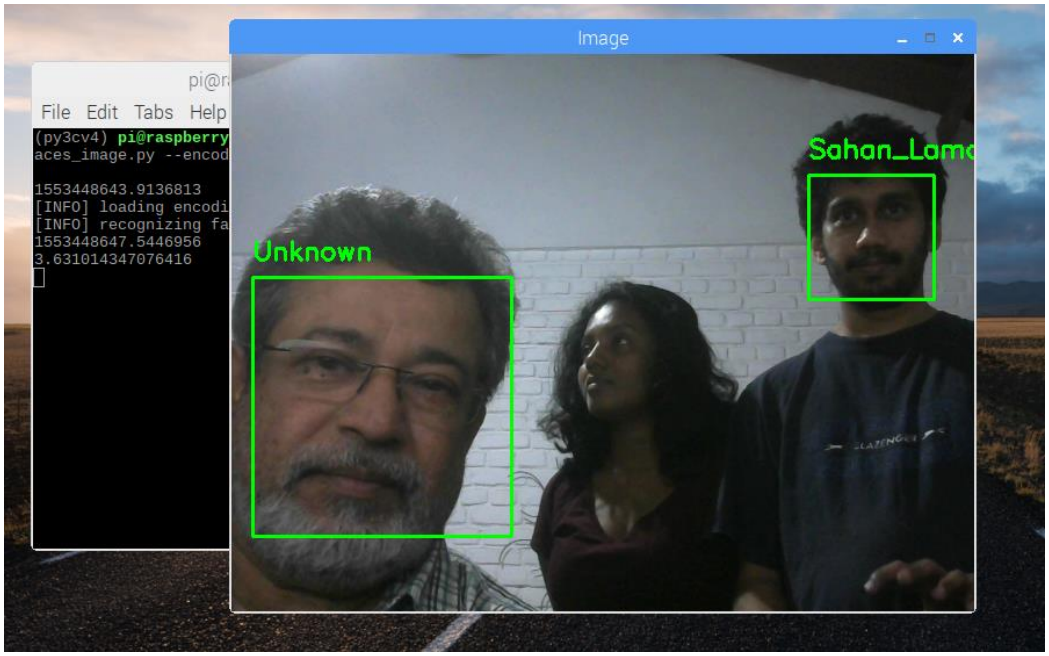

Figure 7.2.16 - Image recognition in under lighting condition with close to camera and one trained subject facing away from camera

In this scenario dataset 1 is identified as Sahan Lamahewa and dataset 2 is not identified as Lakmali Gunaratne as expected. The reason behind the facial recognition algorithm not being able to identify the person represented by the dataset 2 could be because the person is facing away from the camera and the algorithm is unable to properly extract the matching characteristics of the facial features. Like the fig. 7.2.8 in this scenario also the subjects were within similar distance from the camera and also the subjects were under lighting conditions. The person which is not trained to be identified by the facial recognition algorithm using training data is identified as 'Unknown'. Overall 3.631 seconds taken to analyse and output results.

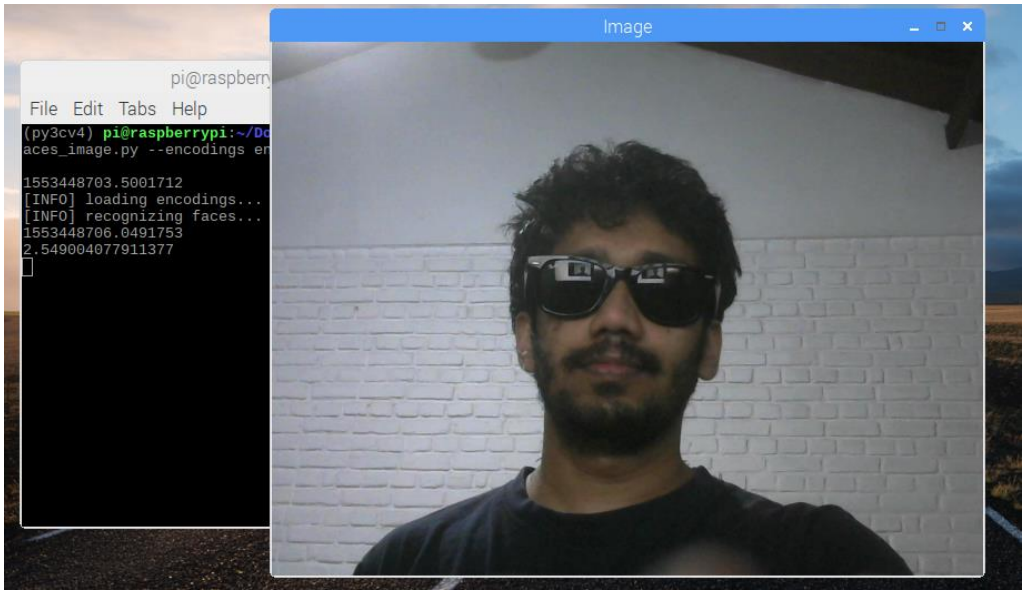

Figure 7.2.17 - Image recognition when the trained subject is directly facing the camera under lighting and wearing accessories

In here dataset 1 is not identified as Sahan Lamahewa. In this instance the facial features are not properly extracted there for the person is not identified. Overall 2.549 seconds taken to analyse and output results.

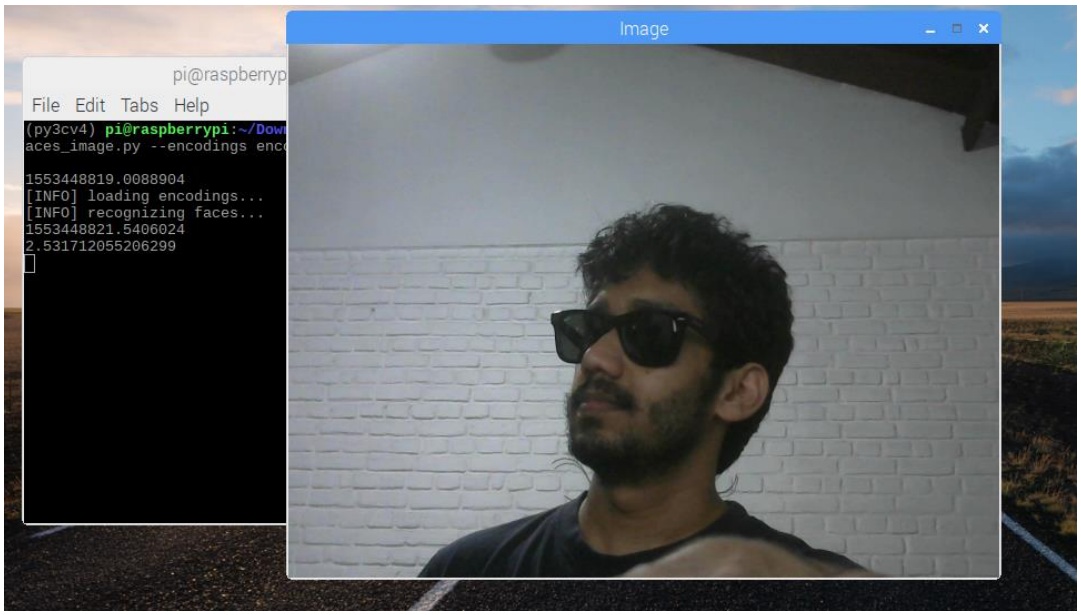

Figure 7.2.18 - Image recognition when the trained subject is facing the camera in forty-five-degree angle under lighting and wearing accessories

In this scenario dataset 1 is not identified as Sahan Lamahewa. The person is facing the camera in a forty-five-degree angle and is staying within a close proximity to the camera under lighting wearing a sunglass. Apart from the facial accessory, clearly due to the angle of the face, the facial recognition algorithm is unable to properly extract

the matching facial features and identify the person. Overall 2.532 seconds taken to analyse and output results.

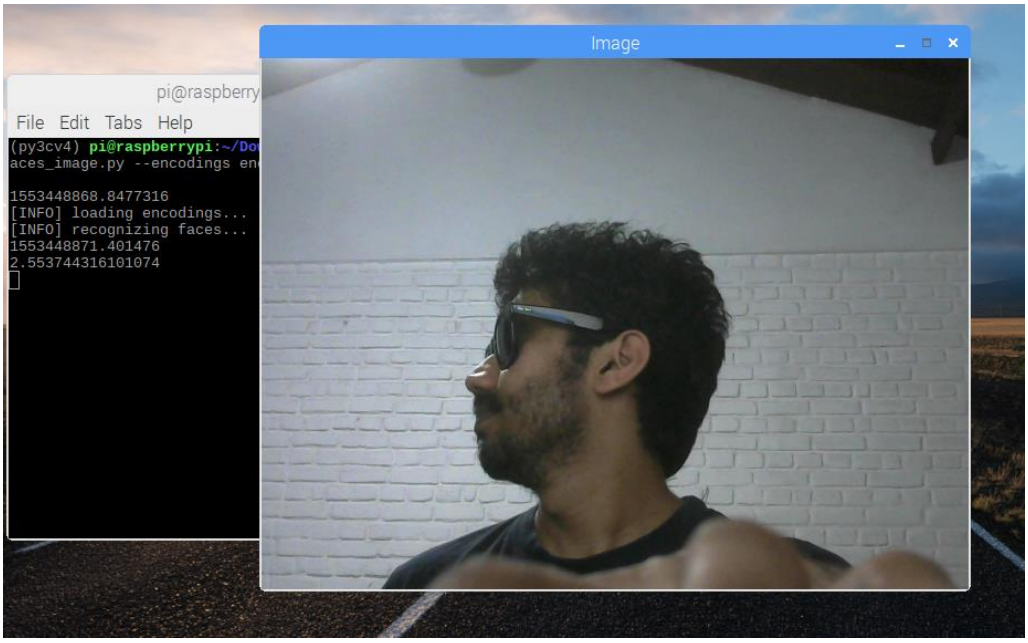

Figure 7.2.19 - Image recognition when the trained subject is facing the camera in ninety-degree angle under lighting and wearing accessories

In this scenario dataset 1 is not identified as Sahan Lamahewa. The person is facing the camera in a ninety-degree angle and is staying within a close proximity to the camera under lighting wearing a sunglass. Similar as shown in figure 7.2.18, apart from the facial accessory and since the lighting condition is also good it's clear that due to the angle of the face, the facial recognition algorithm is unable to properly extract the matching facial features and identify the person. Overall 2.553 seconds taken to analyse and output results.

4. Image recognition in daytime

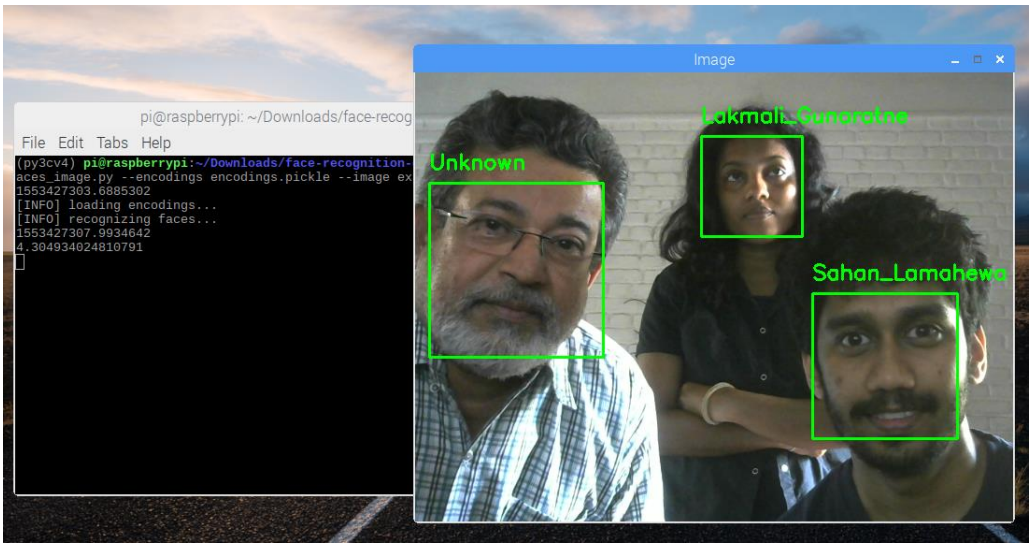

Figure 7.2.20 - Image recognition in under daylight condition with close to camera

In here dataset 1 is identified as Sahan Lamahewa and dataset 2 is identified as Lakmali Gunaratne as expected since all the subjects were within similar distance from the camera and also the subjects were under daylight conditions. The person which is not trained to be identified by the facial recognition algorithm using training data is identified as 'Unknown'. Overall 4.305 seconds taken to analyse and output results.

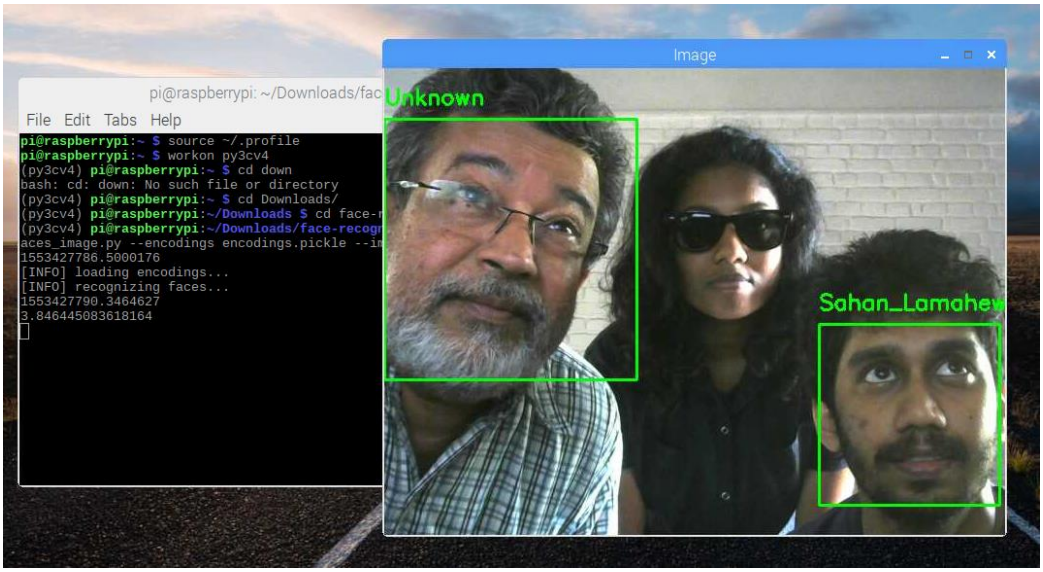

Figure 7.2.21 - Image recognition in under daylight condition with close to camera and wearing accessories

In here dataset 1 is identified as Sahan Lamahewa as expected but here the person represented in dataset 2 is wearing sun glasses and algorithm unable to identify. The subjects were within similar distance from the camera and also the subjects were under daylight conditions. The person which is not trained to be identified by the facial recognition algorithm using training data is identified as 'Unknown'. Overall 3.846 seconds taken to analyse and output results.

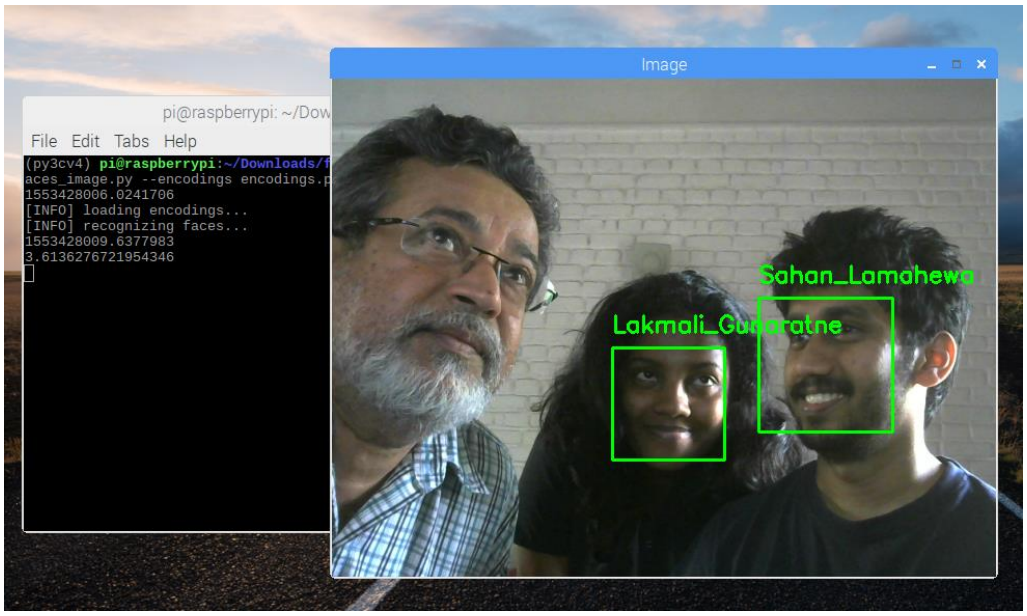

Figure 7.2.22 – One trained subject facing away from camera under daylight

In here dataset 1 is identified as Sahan Lamahewa and dataset 2 is identified as Lakmali Gunaratne as expected since all the subjects were within similar distance from the camera and also the subjects were under daylight conditions. Dataset 1 is facing away from the camera. The person which is not trained to be identified by the facial recognition algorithm using training data is not identified as 'Unknown'. Overall 3.614 seconds taken to analyse and output results.

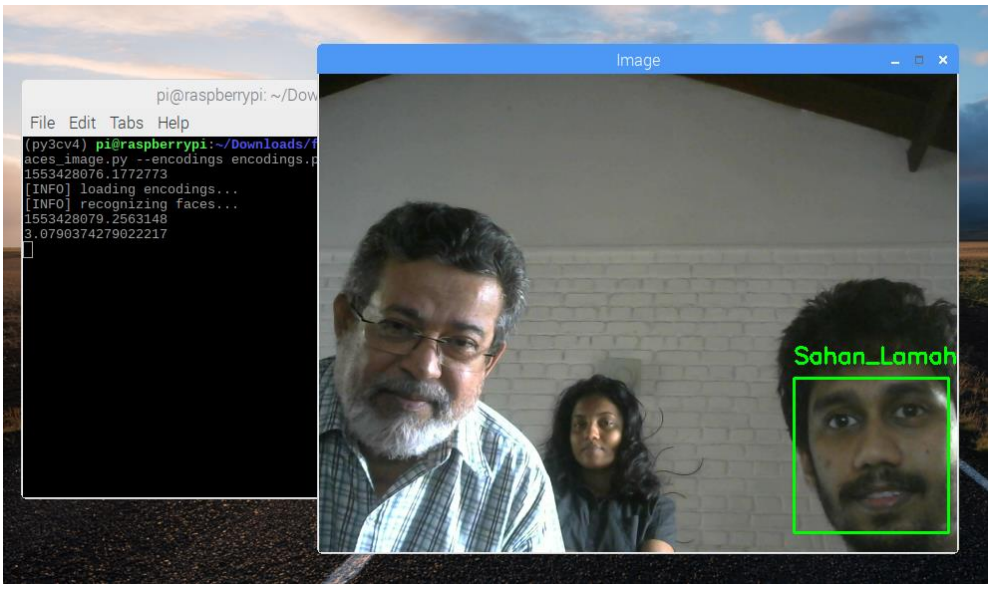

Figure 7.2.23 - Image recognition in under daylight condition with different distances

In this scenario only dataset 1 is identified as Sahan Lamahewa. Dataset 2 and unkown person were unable to identified by the algorithm. Overall 3.079 seconds taken to analyse and output results.

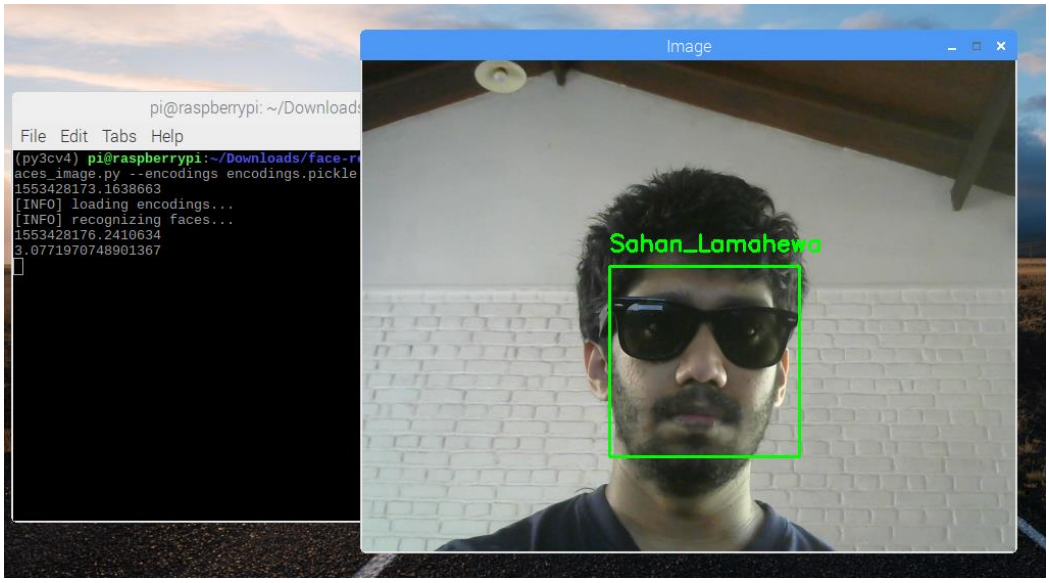

Figure 7.2.24 - Image recognition when the trained subject is directly facing the camera under daylight and wearing accessories

In here dataset 1 is identified as Sahan Lamahewa. Overall 3.077 seconds taken to analyse and output results.

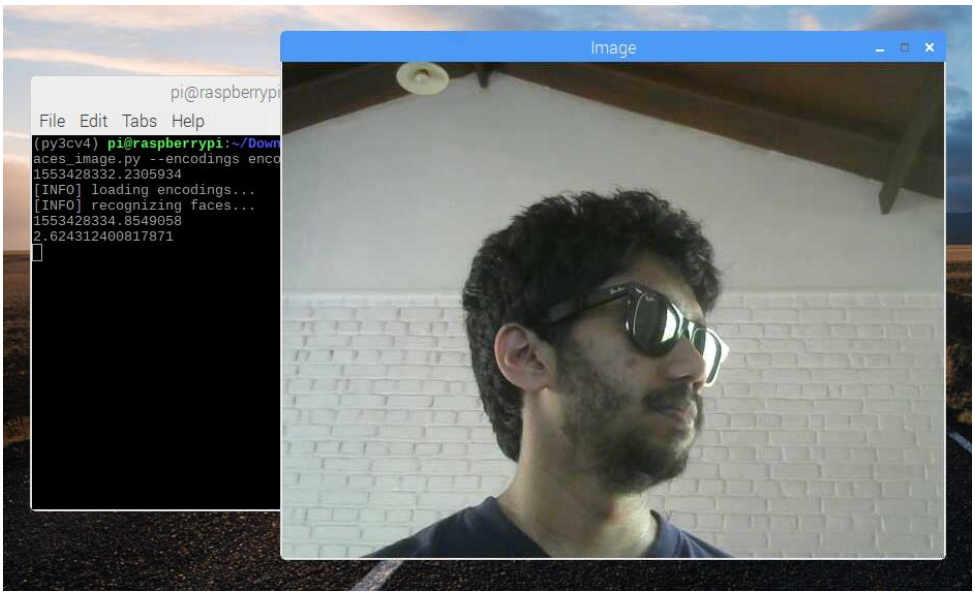

Figure 7.2.25 - Image recognition when the trained subject is facing the camera in forty-five-degree angle under daylight and wearing accessories

In this scenario dataset 1 is not identified as Sahan Lamahewa. The person is facing the camera in a forty-five-degree angle and is staying within a close proximity to the camera under daylight wearing a sunglass. Apart from the facial accessory, clearly due to the angle of the face, the facial recognition algorithm is unable to properly extract the matching facial features and identify the person. Overall 2.624 seconds taken to analyse and output results.

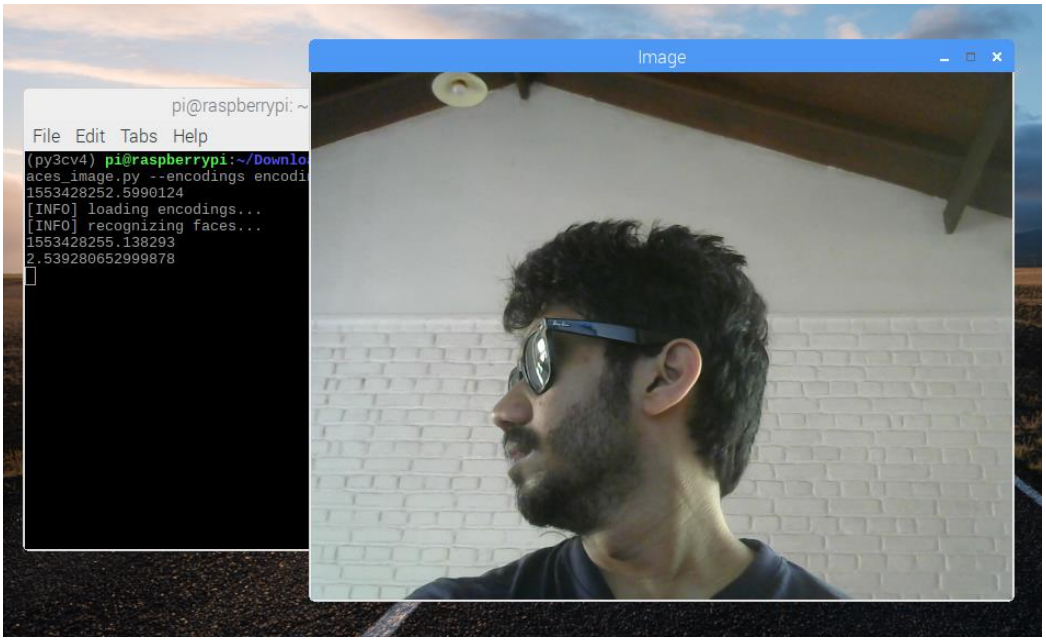

Figure 7.2.26 - Image recognition when the trained subject is facing the camera in ninety-degree angle under daylight and wearing accessories

In this scenario dataset 1 is not identified as Sahan Lamahewa. The person is facing the camera in a ninety-degree angle and is staying within a close proximity to the camera under daylight wearing a sunglass. Similar as shown in figure 7.2.25, apart from the facial accessory and since the lighting condition is good it's clear that due to the angle of the face, the facial recognition algorithm is unable to properly extract the matching facial features and identify the person. Overall 2.539 seconds taken to analyse and output results.

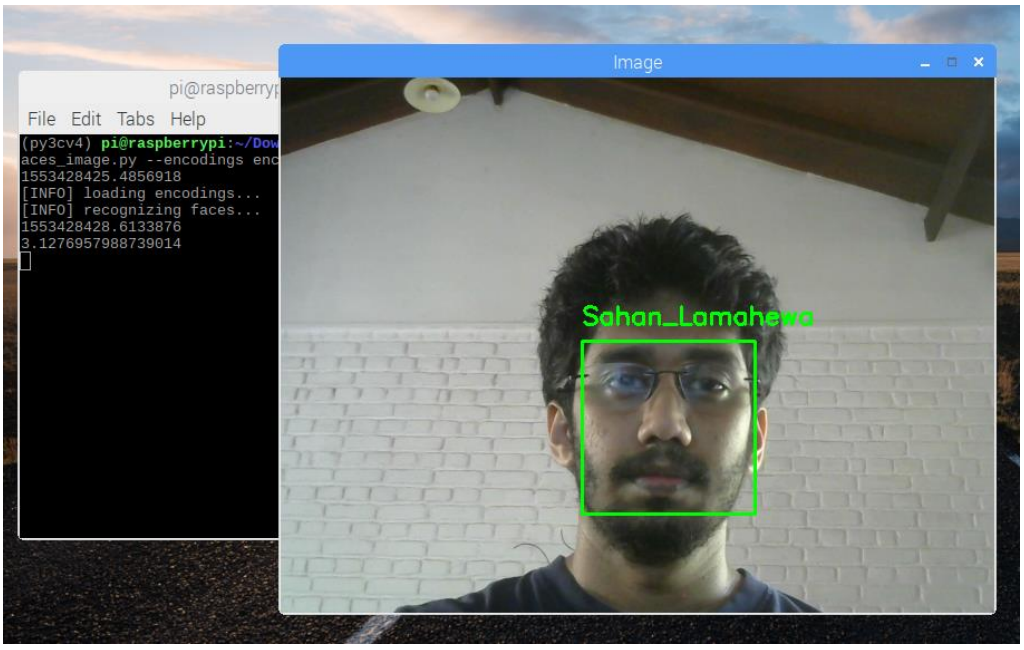

Figure 7.2.27 - Image recognition when the trained subject is directly facing the camera under daylight and wearing accessories (spectacles)

In here dataset 1 is identified as Sahan Lamahewa. The subject is waring spectacles and the facial recognition algorithm seems to be able to extract matching facial feature since the eyes are clearly visible in this scenario. Overall 3.128 seconds taken to analyse and output results.

5. Image recognition in sunlight and the face is covered partially

In this test case, the person is standing directly in the sun light and the face is partially covered from part of a cloth. And yet the algorithm successfully managed to identify the person.

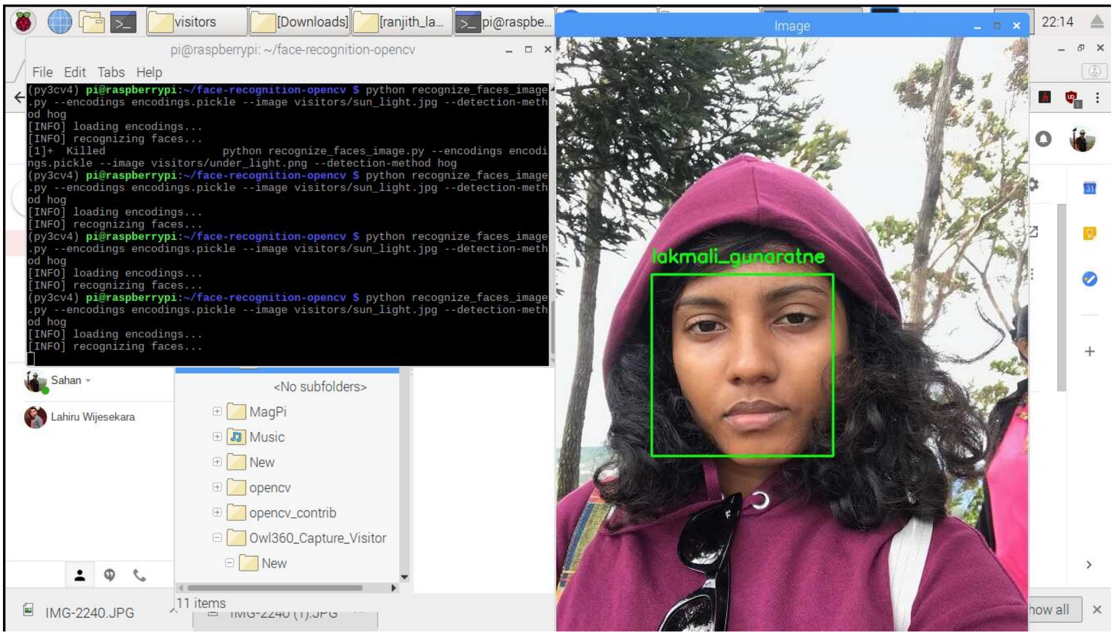

Figure 7.2.28 - Image recognition in sunlight and the face is covered partially

6. Image recognition when the person is wearing an accessory

In the following test case scenario user is wearing sunglasses and once the facial recognition algorithm ran, it managed to identify the faces even when the person was wearing sunglasses.

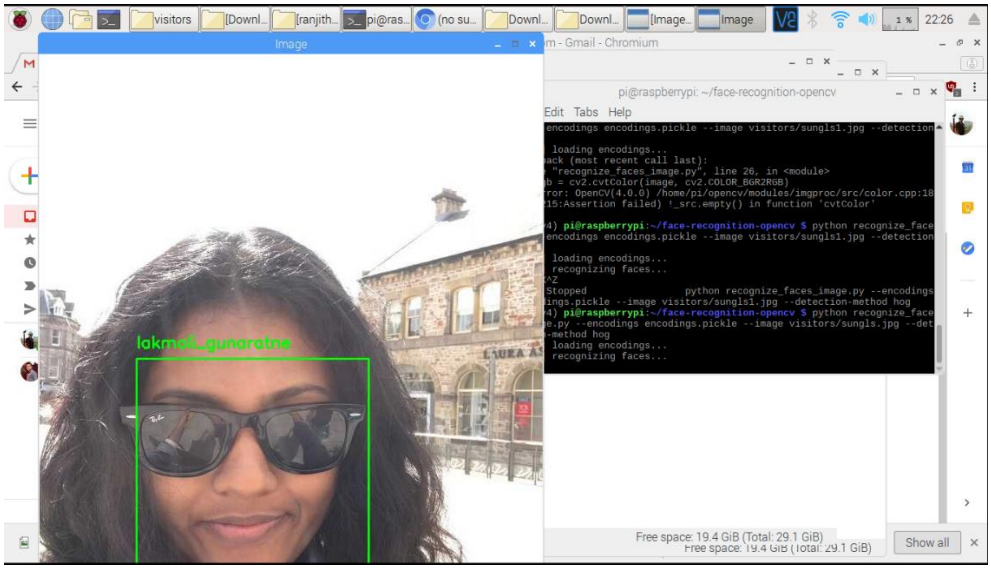

Figure 7.2.29 - Image recognition when the person is wearing an accessory

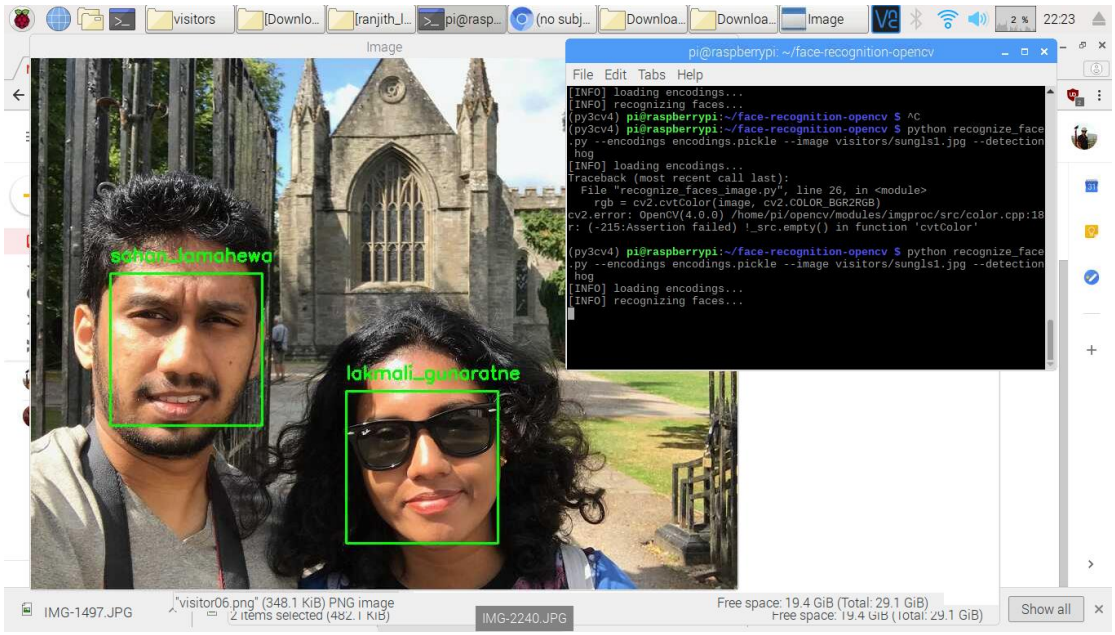

Figure 7.2.30 - Image recognition when the person is wearing an accessory

7. Image recognition when there's an unknown person

When there's an unknown person, the facial recognition tags the unknown person as "Unknown". As shown in the following test scenario, when the image is consisted of both unknown and known persons, the algorithm can successfully identify both persons belonging to each category.

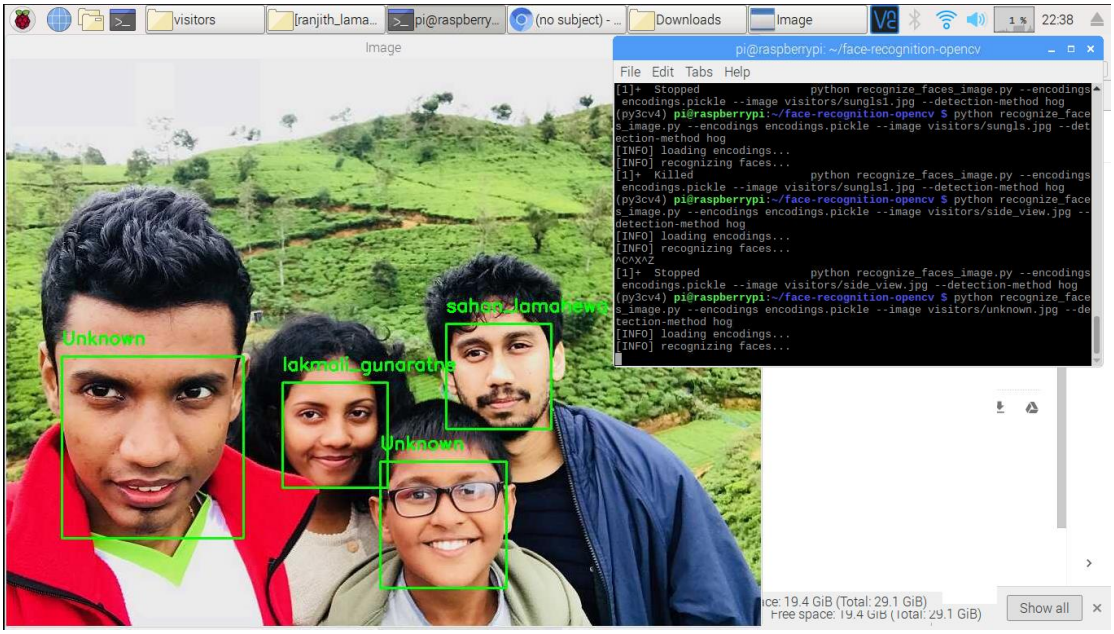

Figure 7.2.31 - Image recognition when there's an unknown person

8. Image recognition when there's a side view of a face

As shown in the image below, the facial recognition algorithm is unable to identify a face, when a side-view of the face is presented.

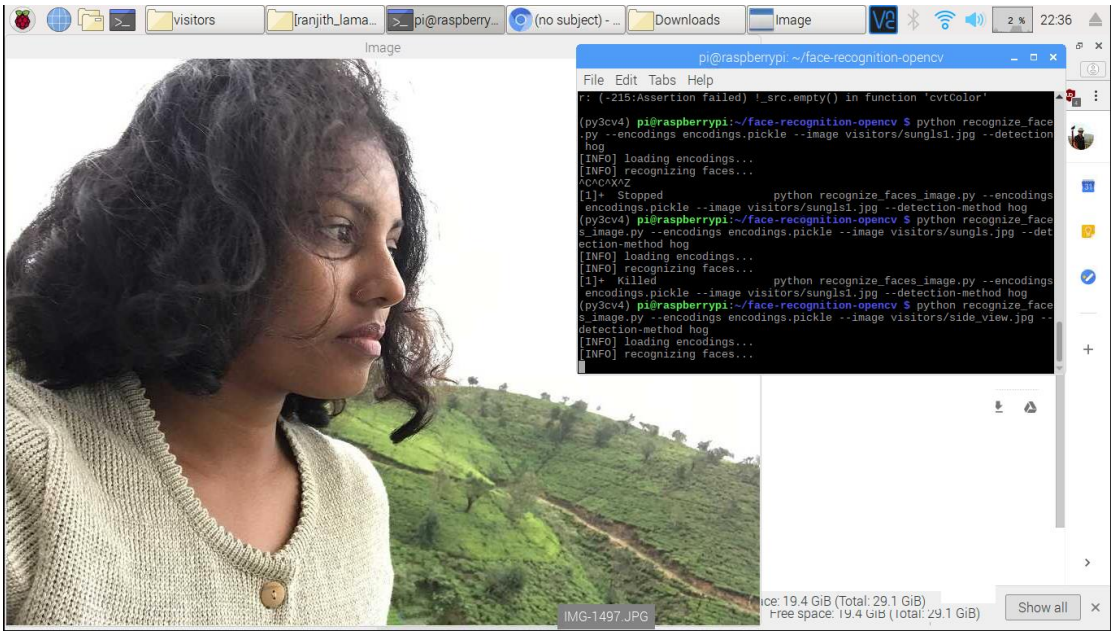

Figure 7.2.32 - Image recognition when there's a side view of a face

7.2.2. WebRTC one-way video conference call

Using the WebRTC technology, visitor can have a one-way video conference call with the home owner. This scenario was tested and the following test results were collected. From the test results it can be concluded that one-way video conference in supported by different networks.

1. iOS

In Safari browser, the connection was successfully established on the Apple iOS when visitor and user are on same network and different networks.

Figure 7.2.33 – iOS one-way video conference

2. Android

In Google Chrome and Firefox browser, the connection was successfully established on the Android OS when visitor and user are on same network and different networks.

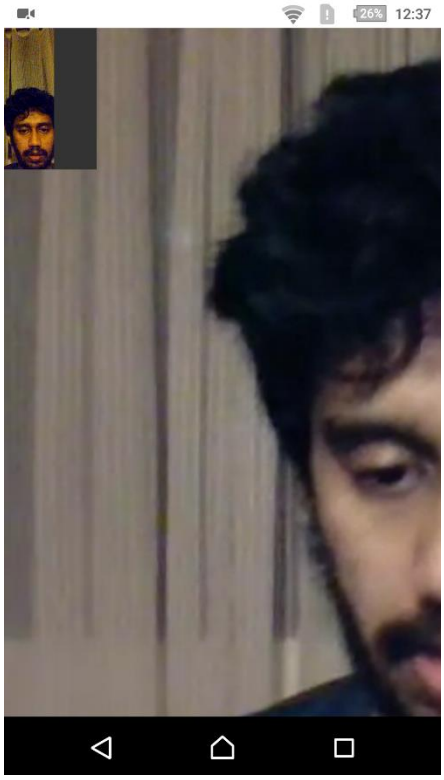

Figure 7.2.34 - Android one-way video conference

The connection between visitor and user can be established in different networks. The following table displays how this function works on different browsers and computer and mobile platforms.

Desktop		Mobile		
		iPhone	Android	
Chrome	Firefox	Safari	Chrome	Firefox
✓	✓	✓	✓	✓

Table 7.1 - Browser compatibility

7.3. Summary

In this chapter, test cases of all major modules of the system were discussed and necessary test data and findings were provided to support the expected functionalities

of the said major modules of the implemented system. In the next chapter, an overall discussion on the proposed system, its limitations and future work will be discussed.

Chapter 8

8. Conclusion and future work

8.1. Introduction

In the previous, evaluation of the entire system was done by performing several test cases and listing out the findings according to the test results. In the current chapter, entire research work which has been conducted up to now, any limitations of the system and how this proposed home security system can be improved, integrated or customized in the future will be thoroughly discussed.

8.2. Discussion

By observing the results obtained, it can be concluded that most of the time, the facial recognition algorithm has the ability to identify the trained datasets properly if the subject's face is properly captured regardless of the distance, lighting condition or facial accessory availability. The algorithm is more likely to identify faces if the subjects are directly facing the camera rather than facing the camera in different angles.

It was also observed that more than 4 seconds were taken to identify if three subjects were identified in a single frame. Further, the time taken to identify only two subjects out of three was always less than 4 seconds. Table 8.1 shows the summarized analysis that was derived from the different test scenarios.

Figure	Light Condition	No of Subjects	No of identified subjects	Time
7.2.4	Low Light	3	2	3.618
7.2.5	Low Light	3	3	4.174
7.2.6	Low Light	3	3	4.146
7.2.7	Low Light	3	3	4.146
7.2.8	Low Light	3	2	3.629
7.2.9	Low Light	1	1	3.142
7.2.10	Low Light	1	1	2.587
7.2.11	Low Light	1	1	2.587
7.2.13	Under light	3	3	4.179
7.2.14	Under light	3	3	4.238
7.2.15	Under light	3	2	3.633
7.2.16	Under light	3	2	3.631
7.2.17	Under light	1	1	2.549
7.2.18	Under light	1	1	2.532
7.2.19	Under light	1	1	2.553
7.2.20	Daylight	3	3	4.305
7.2.21	Daylight	3	2	3.846
7.2.22	Daylight	3	2	3.614
7.2.23	Daylight	3	1	3.079
7.2.24	Daylight	1	1	3.077
7.2.25	Daylight	1	1	2.624
7.2.26	Daylight	1	1	2.539
7.2.27	Daylight	1	1	3.128

Table 8.1 – Time taken to face recognition analysis

In conclusion, the number of identified subjects are directly proportional to the time taken to identify the subjects.

The time taken to train a dataset is also directly proportional to the number of images included in the dataset. This can be seen from the following Figure 8.1.

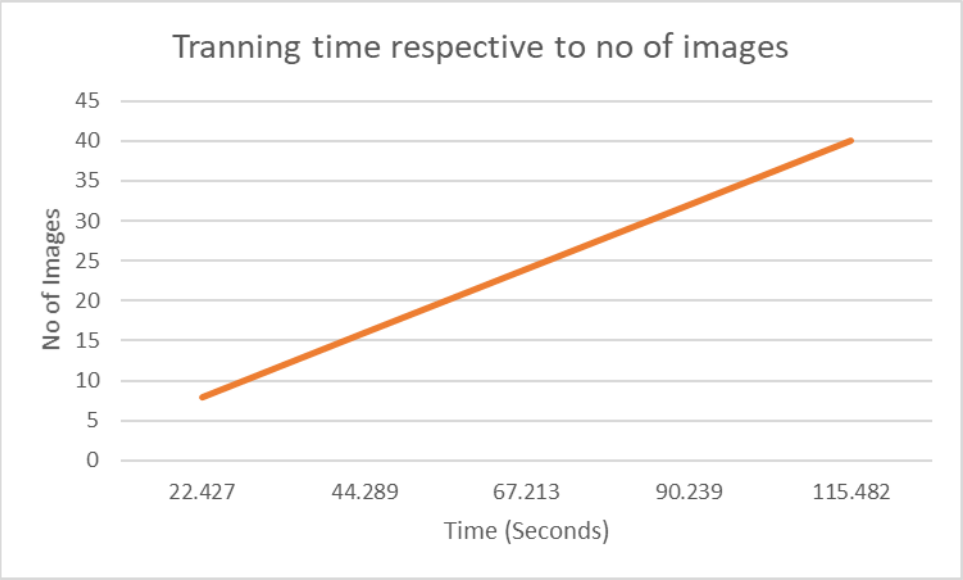

Figure 8.1 – Training time respective to no of images

8.3. Limitations

In some instances, WebRTC will not work and can be slower. There's a slight delay in video streaming as it is taken by frame by frame. Further, face recognition algorithm is unable to identify faces with side views.

8.3. Future Development

This implemented solution for home security can be enhanced and developed further as a security system for server rooms. The system can also be developed as a number plate recognition system. The system can be enhanced to automatically open & close doors.

8.4. Summary

This chapter concludes the research work done to find a low cost, compact, simple & platform independent solution to mitigate door step crimes using Raspberry Pi. The Smart Door Bell solution discussed in the thesis for a home security system can be developed for a cheaper cost unlike its competitors in the current market. Most similar researches related to doorbell security systems contain several setbacks. If it's low-cost

then the technologies used are complex. The technology is simple but the development cost is high for some projects. Most systems only contain the basic functionalities without providing any additional useful features.

Finally, the proposed system was developed in a way to overcome issues mentioned above which were identified while studying the past similar work therefore it gives an extra value to this solution and at the same time is affordable to the general crowd.

References

[1] Andrew E., Ian C., Thmothy P., Chris D, (2017), Doorstep : a doorbell security system for the prevention of doorstep crime, *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, Orlando, FL, USA

[2]www.police.lk/images/others/crime_trends/2017/grave_crime-2017.pdf

[3]Anvekar R. G. and Banakar R. M., "IoT application development: Home security system," *2017 IEEE Technological Innovations in ICT for Agriculture and Rural Development (TIAR)*, Chennai, India, 2017.

[4] Lucas M. A. H., Luis L. A., Maria E. B., Mariano R., Juliana T. and Sergio G., "Smart Doorbell: An ICT solution enhance inclusion of disabled people," *ITU Kaleidoscope: Trust in the Information Society (K-2015)*, Barcelona, Spain, 2015.

[5] Park W. and Cheong Y., "IoT smart bell notification system: Design and implementation," *2017 19th International Conference on Advanced Communication Technology (ICACT)*, Bongpyeong, South Korea, 2017.

[6] Ennis, A., Cleland, I., Patterson, T., Nugent, C., Cruciani, F., Paggetti, C., Morrison, G. and Taylor, R., "Doorstep: A doorbell security system for the prevention of doorstep crime," *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, Orlando, FL, USA, 2016.

[7] Kumari P., Goel P., and Reddy S. R. N., "PiCam: IoT Based Wireless Alert System for Deaf and Hard of Hearing," *2015 International Conference on Advanced Computing and Communications (ADCOM)*, Chennai, India, 2015.

[8] Sahani M., Nanda C., Sahu A. and Pattnaik B., "Web-based online embedded door access control and home security system based on face recognition," *2015 International Conference on Circuits, Power and Computing Technologies [ICCPCT-2015]*, Nagercoil, India, 2015.

[9] Sruthy S. and Sudhish N., "Wifi enabled home security surveillance system using Raspberry Pi and IoT module," *2017 IEEE International Conference on Signal Processing, Informatics, Communication and Energy Systems (SPICES)*, Kollam, India, 2017.

[10] Zhao Y. and Ye Z., "A low cost GSM/GPRS based wireless home security system," *IEEE Transactions on Consumer Electronics (Volume: 54 , Issue: 2 , May 2008)*, 2008.

[11] Ayman, B. T. (2015), Enhanced smart doorbell system on face recognition, *16th international conference on Sciences and Techniques of Automatic control & computer engineering - STA'2015*, Monastir, Tunisia

[12] www.raspberrypi.org/help/videos/#what-is-a-raspberry-pi

[13] www.projects.raspberrypi.org/en/projects/physical-computing

[14] en.wikipedia.org/wiki/Apache_Cordova

[15] www.mathworks.com/discovery/deep-learning.html

[16] www.fullstackpython.com/webrtc.html

[17] www.fullstackpython.com/websockets.html

[18] www.w3schools.com/nodejs/nodejs_intro.asp

Appendix – Acronyms

API	-	Application Programming Interface
CCTV	-	Closed Circuit TeleVision
CSS	-	Cascade Style Sheets
GPIO	-	General-Purpose Input/Output
GPRS	-	General Packet Radio Service
GSM	-	Global System for Mobile Communications
HTML		Hyper Text Markup Language
IoT	-	Internet of Things
LBPH	-	Local Binary Patterns Histograms
NAT	-	Network Address Translation
OpenCV	-	Open Source Computer Vision
PIR	-	Passive Infrared Sensor
RTC	-	Real Time Communication
SDP	-	Session Description Protocol
SMS	-	Short Message Service
STUN	-	Session Traversal Utilities of NAT
URI	-	Uniform Resource Identifier
USB	-	Universal Serial Bus
WebRTC	-	Web Real-Time Communication
Wireless LAN	-	Wireless Local Area Network