“Smart Eye”

A project report of Phase-II submitted to

Rajiv Gandhi Proudyogiki Vishwavidhyalaya, Bhopal

towards partial fulfillment of

the degree of

Bachelor of Engineering in Computer Engineering

Guided By:

Dr. Anuradha Purohit
Associate Professor
Department of Computer Engg.

Mrs. Ritambhara Patidar
Assistant Professor
Department of Computer Engg.

Submitted By:

0801CS173D06 - Harish Lohare

0801CS173D18 - Yash Pathak

0801CS161067 - Pushpendra Sastiya

0801CS173D13 - Shalini Singh

0801CS161039 - Lokendra Singh

DEPARTMENT OF COMPUTER ENGINEERING

SHRI G.S. INSTITUTE OF TECHNOLOGY AND SCIENCE, INDORE(M.P.)

2019-2020

SHRI G.S. INSTITUTE OF TECHNOLOGY AND SCIENCE, INDORE(M.P.)

RECOMMENDATION

The project report of Phase-II entitled ***"Smart Eye"*** submitted by: **0801CS173D06 - Harish Lohare, 0801CS173D18 - Yash Pathak, 0801CS161067 - Pushpendra Sastiya, 0801CS173D13 - Shalini Singh, 0801CS161039 - Lokendra Singh**, students of B.E. IV year in the session 2019-2020, towards partial fulfillment of the degree of **Bachelor of Engineering in Computer Engineering** of Rajiv Gandhi Proudyogiki VishwaVidhyalaya, Bhopal is a satisfactory account of their work.

Dr. Anuradha Purohit
Project Guide
Department of Computer Engg.

Dr. Urjita Thakar
Head of Dept.
Department of Computer Engg.

Mrs. Ritambhara Patidar
Project Co-guide
Department of Computer Engg.

Dean (Academics)
S.G.S.I.T.S. Indore

SHRI G.S. INSTITUTE OF TECHNOLOGY AND SCIENCE, INDORE(M.P.)

CERTIFICATE

The project report of Phase-II entitled ***"Smart Eye"*** submitted by: **0801CS173D06 - Harish Lohare, 0801CS173D18 - Yash Pathak, 0801CS161067 - Pushpendra Sastiya, 0801CS173D13 - Shalini Singh, 0801CS161039 - Lokendra Singh**, students of B.E. IV year in the session 2019-2020, towards partial fulfillment of the degree of **Bachelor of Engineering in Computer Engineering** of Rajiv Gandhi Proudyogiki VishwaVidhyalaya, Bhopal is a satisfactory account of their work.

Internal Examiner **External Examiner**

Date:

SHRI G. S. INSTITUTE OF TECHNOLOGY AND SCIENCE, INDORE (M.P)

DECLARATION

We, **0801CS173D06 - Harish Lohare**, **0801CS161039 - Lokendra Singh**, **0801CS161067 - Pushpendrasingh Sastiya**, **0801CS173D13 - Shalini Singh**, **0801CS173D18 - Yash Pathak**, hereby declare that the work presented in the B.E project report has been carried out by us. We further declare that the work submitted for the award of the degree doesn't contain any part of the work which has been submitted for the award of any degree either in this university or any other University without proper citation.

Date:

0801CS173D06 - Harish Lohare

0801CS161039 - Lokendra Singh

0801CS161067 - Pushpendrasingh Sastiya

0801CS173D13 - Shalini Singh

0801CS173D18 - Yash Pathak

ACKNOWLEDGEMENT

We express our profound sense of gratitude to our project guide **Dr. Anuradha Purohit** and co-guide **Mrs. Ritambhara Patidar** who had advised us to do project work entitled **"Smart Eye"**. Their continuous support and motivation always made us deliver our best. Presence of such guides have always been an amazing experience, which is also a valuable gift for an engineer to progress in his/her life.

We are also grateful to **Dr. Urjita Thakar**, Head, Department of Computer Engineering and **Prof. Rakesh Saxena**, Director, S.G.S.I.T.S Indore, for providing us with numerous facilities and academic environment during the course of study.

We sincerely wish to express, our gratefulness to all the members of staff of Department of Computer Engineering, S.G.S.I.T.S Indore, who have extended their cooperation at all times and have contributed in their own way in developing the project.

The successful completion of the project is not an individual effort. It is an outcome of the cumulative effort of a number of people, each having their own importance to the objective. We express love and respect towards our parents and the entire family member who are our strength in everything we do. We are thankful to them for their constant support and motivation.

With a blend of gratitude, pleasure and great satisfaction we convey our indebtedness to all those who have directly or indirectly contributed to the successful completion of our project work.

0801CS173D06 - Harish Lohare

0801CS173D18 - Yash Pathak

0801CS161067 - Pushpendra Sastiya

0801CS173D13 - Shalini Singh

0801CS161039 - Lokendra Singh

ABSTRACT

Home security has become the prime concern for everyone in recent times. Current systems available are unable to prevent the unwanted activities from happening. Expensive setup and high maintenance are common characteristics of these systems. In this project a system is designed to keep track of all the activities happening in the vicinity of the house. An attempt has been made to develop a home security system which is accessible, affordable and yet effective.

This project is a modification on the systems based on CCTV cameras to achieve the ability for keeping the target location (Home, Office, School etc.) safe from unwanted activities by implementing IoT technologies. This project utilises the multi functionality of Raspberry pi such that after adding a camera module, it works as a security system. Raspberry pi comes with enough processing power to implement both face detection and recognition and hence used for the same motive. These component are implemented together in a way to minimise human intervention and therefore fulfils the overall objective of this project.

Proposed system has major functionalities like detecting the changes in the surveillance area and sending notification accordingly, face detection, live video streaming and an alarm system. Proposed system utilises functionalities of Raspberry pi and Python.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

Introduction

1.1 Preamble

Increased frequency of theft and burglary is a major concern for city dwellers. The facilities provided by the current system are not enough to prevent the crime from happening. The current system doesn't have a feedback loop to keep the concerned person aware of all the activities. Secondly, it is not capable in itself to detect any misconduct. Similarly high cost of installation make the concerned person with limited budget reluctant to have any kind of security system. As a result large portion of the city end up being a target area for thieves and mischief's. Hence it is very important to upgrade the security level whether it is at home, office or school and to make it affordable.

Recent advancements in communication technologies or more specifically in IoT (Internet of Things) have opened a large spectrum of opportunities in the field of real-time systems and streamline processes. These advancement have direct application in the security sector. With these advancements the user now can get the notification instantly which ultimately empowers the user to take measures proactively. With latest technology coming in handy for real time applications, an effective, accessible and affordable home security system is need of the hour.

The internet of things, or IoT, is a system of interrelated computing devices with the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction. IoT devices share the data they collect by connecting to an IoT gateway or other edge device where data is either sent to the cloud to be analysed or analysed locally. Sometimes, these devices communicate with other related devices and act on the information they get from one another. The devices do most of the

work without human intervention, although people can interact with the devices – for instance, to set them up, give them instructions or access the data.

1.2 Need of the Project

Increased frequency of theft or burglary is a major problem for residents of any locality. In many instances the information about theft comes long after the incident, more importantly it requires a constant watch by a person in order to prevent it beforehand. This is not a feasible solution if all the family members are working professionals. Available system do not provide any kind of alert if the suspicious activity is going on. Secondly, In daytime there are many visitors of which some are familiar and some are not available system do not provide any distinction mechanism to recognise family members.

1.3 Problem Statement

Fear of theft and burglary is a menace for any city dweller. In absence of any preventive measures, cases of robbery are frequent. Ongoing use of an obsolete technology for security purposes is ineffective. There is a clear incentive for the improvement. The inefficiencies of the currently used systems can be rectified by using available technologies. In such times there is a need for an effective, accessible and affordable security system to overcome these threats.

1.4 Objectives

The main objective is to design, develop and implement a smart security system-“Smart Eye” to provide a real-time system which is capable of detecting unwanted activities and alert the user accordingly.

1.5 Organization of the Report

Chapter 1 introduces the context of the project giving an insight of the motivation, need and scope of the project. Chapter 2 describes the literature survey done to study the concepts of the existing technologies. It also elucidates the area of concern where the “smart eye security system” is of great importance and also describes in detail the background survey conducted to understand the hardware components required to design the system. It also includes the tools and technologies used in developing the system.
Chapter 3 contains the literature review written after reading several research paper on relevant topics. It also analyse the progress done till date and the improvements made during the project. Chapter 4

deals with the functional and the non-functional requirements, system components, use case diagram, use case scenario, data flow diagram and feasibility study. Chapter 5 focuses on design details comprising of the basic architecture of the system, activity diagram and state chart diagram. Chapter 6 details the programming code of the project. Chapter 7 is the record of all the test done on the system and it's subsequent results. Chapter 8 concludes and summarizes the work done. The appendix and references are listed in a separate section.

Chapter 2

Background Study

2.1 Area of Concern

The common conception about theft is that there is more possibility of theft at a place isolated from the dense locality or that it occurs more frequently in night then in day time. Similarly it is largely accepted that CCTV's are the best possible solution for this problem. In contrary, this complacency works as a positive feedback for the thieves and mischief. The current system does not act as a deterrent and the frequency of theft increases instead of decreasing.

For the family where all the members are working professionals, it is a major concern to install a security system for the home. Secondly, if the system does not actively detect any misconduct then it is of little or no use.

The available systems are expensive and still without any facility to prevent the crime. Sometimes there is a need to hire a person to regularly check the camera feed. If the surveillance area is vast this solution also fails.

2.2 Tools and Technologies

In this section, description of all the tools- software or hardware, is given. In depth details of their functionality and market price is also included in this section.

2.2.1 Raspberry pi

The Raspberry Pi is a tiny credit card size computer. Just by adding a keyboard, mouse, display, power supply, micro SD card with installed Linux Distribution it gets converted into a fully-fledged

computer that can run applications from word processors and spreadsheets to games.Technical specification of the board are illustrated in table 2.1 .

- It is a mini computer with Raspbian OS. It can run multiple programs at a time.
- It requires complex task like installing libraries and software for interfacing sensors and other components.

Figure 2.1 shows components of a Raspberry Pi Zero W board and their location. Main components like Processor, HDMI, MicroSD, GPIO, power supply ports, USB port (In clockwise direction).

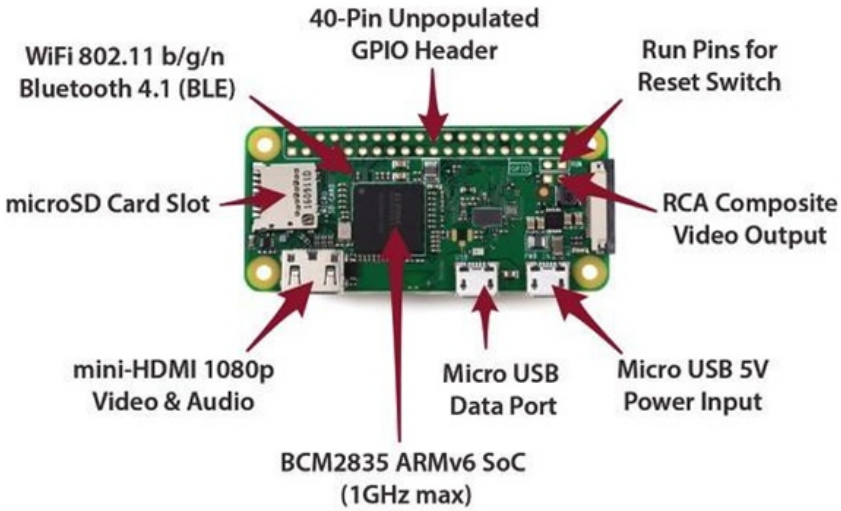


Figure 2.1: **Raspberry pi zero W board**

I. Technical Specification of Raspberry pi zero W

Table 2.1: **Specifications of Raspberry Pi Components**

Processor	ARMv6 CPU (BCM2835) (Single-core@ 1GHz)
Graphics	VideoCore IV GPU
Memory	512 MB RAM
Port	Mini-HDMI Micro USB For Data/Peripheral Micro USB power MicroSD slot
Connectivity	Wireless LAN Wi-Fi Bluetooth
Interface	CSI(Camera)
Price	2250 Rs

II. Raspberry pi NoIR camera module

- The Raspberry Pi NoIR Camera V2 is the “night vision” version of the official camera board released by the Raspberry Pi Foundation.
- The Raspberry Pi NoIR Camera Board V2 features an ultra-high quality 8 megapixel Sony IMX219 image sensor (up from 5MP on the V1 camera board), and a fixed focus camera lens.
- It is capable of 3280 x 2464 pixel static images, and also supports 1080p30, 720p60 and 640x480p90 video.
- The module attaches to Raspberry Pi by way of a 15 Pin Ribbon Cable, to the dedicated 15-pin MIPI Camera Serial Interface (CSI), which was designed especially for interfacing to cameras.

Figure. 2.2 shows Raspberry Pi NoIR Camera Module v2 with a high quality 8 megapixel Sony IMX219 image sensor custom designed add-on board for Raspberry Pi, featuring a fixed focus lens.

Figure 2.2: **Raspberry Pi Camera Module v2**

III. The Raspberry Pi NoIR Camera Board V2 Features:

- No IR Filter – Great for Low Light Conditions
- Includes Blue Filter
- Fixed focus lens on-board
- 150mm CSI camera cable included
- 8 megapixel native resolution sensor-capable of 3280 x 2464 pixel static images

- Supports 1080p30, 720p60 and 640x480p90 video
- Size 25mm x 23mm x 9mm
- Weight just over 3g
- Connects to the Raspberry Pi board via a short ribbon cable (supplied)
- NoIR Camera v2 is supported in the latest version of Raspbian, Raspberry Pi's preferred operating system

2.2.2 Python

- Python is a general purpose programming language started by Guido van Rossum that became very popular very quickly, mainly because of its simplicity and code readability. It enables the programmer to express ideas in fewer lines of code without reducing readability.
- Compared to languages like C/C++, Python is slower.
- Python wrappers that can be used as Python modules.
- The code is as fast as the original C/C++ code (since it is the actual C++ code working in background).
- It is easier to code in Python than C/C++. OpenCV-Python is a Python wrapper for the original OpenCV C++ implementation.

(I) OpenCV-Python

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

Open CV has a modular structure, which means that the package includes several shared or static libraries. The following modules are available:

- **Core** - a compact module defining basic data structures, including the dense multi-dimensional array0 Mat and basic functions used by all other modules
- **Image processing** - an image processing module that includes linear and non-linear image filtering, geometrical image transformations (resize, affine and perspective warping, generic table-based remapping), color space conversion, histograms, and so on.

- **Video** - a video analysis module that includes motion estimation, background subtraction, and object tracking algorithms.
- **Calib3d**- basic multiple-view geometry algorithms, single and stereo camera calibration, object pose estimation, stereo correspondence algorithms, and elements of 3D reconstruction.
- **Features2d**-salient feature detectors, descriptors, and descriptor matchers.
- **Object detection** - detection of objects and instances of the predefined classes (for example, faces, eyes, mugs, people, cars, and so on).
- **High GUI** - an easy-to-use interface to video capturing, image and video codecs, as well as simple UI capabilities.
- **GPU --**accelerated algorithms from different Open CV modules.

Chapter 3

Literature Review

3.1 Inception

A review of completed and ongoing research has been conducted to identify current knowledge or methodologies in the field of face detection and recognition. Several efforts were made in the past to implement such systems, which have motivated and guided us throughout the process. After undertaking the project there were four big questions to answer. This review will answer these questions sequentially.
What tools should be used to implement the project?
What approach should be taken for face recognition and face detection?
What are the drawbacks of the earlier systems?
What are the improvements?

3.2 Tools Required for Implementation

The Internet of things (IoT) is a system of interrelated computing devices, mechanical and digital machines provided with unique identifiers (UIDs) and the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction. Such devices can be modelled into a security system. Steven Fernandez et al. [1], suggested a face detection system using Raspberry Pi. However authors did not approach face recognition implementation due to complexity of the recognition process, since recognition process would need more powerful resources to accomplish better results. Fernan Gallego et al. in [2] were able to switch from the closed-circuit television CCTV graphical processor to a computer graphical processing unit GPU, and embed the security cameras into the computer GPU. “Smart Eye” uses Raspberry pi zero W board which can perform both face recognition and detection. It also uses Raspberry Pi NoIR

Camera Module v2 with a high quality 8 megapixel Sony IMX219 image sensor custom designed add-on board for Raspberry Pi.

3.3 A Study of Available Approaches

Ping Hsin lee et al. in [3], attempted to detect faces in a digital image using various techniques such as skin colour segmentation, morphological processing, template matching, Fisher linear discriminant (FLD), and Eigen face decomposition. P. Viola et al. in [4] discussed rapid object detection method using a boosted cascade of simple features.

There are three main contributions of object detection framework. The first contribution of this paper is a new image representation called an integral imagethat allows for very fast feature evaluation. This detection system does not work directly with image intensities. Like these authors we use a set of features which are reminiscent of Haar Basis functions (though we will also use related filters which are more complex than Haar filters). In order to compute these features very rapidly at many scales they introduce the integral image representation for images. The integral image can be computed from an image using a few operations per pixel. Once computed, any one of these Harr-like features can be computed at any scale or location in constant time.

The second contribution of this paper is a method for constructing a classifier by selecting a small number of important features using AdaBoost. Within any image sub window the total number of Harr-like features is very large, far larger than the number of pixels. In order to ensure fast classification, the learning process must exclude a large majority of the available features, and focus on a small set of critical features. Motivated by the work of Tieu and Viola, feature selection is achieved through a simple modification of the AdaBoost procedure: the weak learner is constrained so that each weak classifier returned can depend on only a 1 single feature. As a result each stage of the boosting process, which selects a new weak classifier, can be viewed as a feature selection process. AdaBoost provides an effective learning algorithm and strong bounds on generalization performance.

The third major contribution of this paper is a method for combining successively more complex classifiers in a cascade structure which dramatically increases the speed of the detector by focusing attention on promising regions of the image. The notion behind focus of attention approaches is that it is often possible to rapidly determine where in an image an object might occur. More complex processing is reserved only for these promising regions. The key measure of such an approach is the “false negative” rate of the attentional process. It must be the case that all, or almost all, object instances are selected by the attentional filter.

3.4 Analysis of Drawbacks & Improvements

Face recognition is one of the most important applications of image analysis, pattern recognition and image processing. Especially in the past two decades, many researchers have focused on it to find ways to improve the accuracy of recognition. Identification and verification are two main use of face recognition. General identity verification, Image database investigations, "Smart Card" applications and video indexing are some of applications of these two purpose.

V. Liong et al in [5] suggest LBP as a solution. Local Binary Pattern (LBP) is known as a powerful tool for texture classification and face recognition due to lack of sensitivity to light variation. Local binary pattern histogram (LBPH) calculates the histogram of all the LBP images. Then compares the histogram vector of each test images with all the train images and finds the best fit. Gray Level Co-occurrence Matrix (GLCM) has already been used for a face recognition system and it has good accuracy on ORL database. The result of direct GLCM have been compared with the results of Haralick's features and it was found the direct GLCM is more accurate for face recognition system. The co-occurrence matrix of local average binary pattern has also been used for face recognition system. In this technique, first, the average of gray-values of pixels are computed and then LBP operator is applied. Then, co-occurrence matrix of processed image is computed. Databases that are used to evaluate the algorithms have different features such as illumination variation, rotation, age variation and so on.

An algorithm cannot recognize faces of different databases with either high or the same accuracy. Each algorithm is appropriate for a specific face database according to its function. For this reason, a method that combines algorithms with different usages can be used to recognize the faces of databases with different characteristics. Based on that they proposed a feature extraction algorithm based on Co-occurrence Matrix of Local Median Binary Pattern (CMLMBP) to improve the accuracy of face recognition system. In the proposed technique, first Local Median Binary Pattern of the input image is computed. Then co-occurrence matrix of the processed image is calculated. This proper combination of these techniques improves the accuracy of ORL and Yale databases 96.25% and 100%, respectively.

Generally, there are four stages in a conventional face recognition system: face detection, face alignment, face representation, and face matching. As a representative pattern recognition problem, face representation and face matching are the most two key stages in a face recognition system.

For face representation, the objective is to extract discriminative features to make face images more separable. For face matching, the goal is to design effective classifiers to differentiate different face patterns. Compared with face matching, face representation significantly affects the performance of a face

recognition system because face images captured in real world environments are usually affected by many variations such as varying poses, expressions, illuminations, occlusions, resolutions, and backgrounds. These variations reduce the similarity of face samples from the same person and increase the similarity of face samples from different persons, which is one of the key challenges in face recognition. In recent years, a number of face representation methods have been proposed, and they can be mainly classified into two categories: holistic features, and local features. Representative holistic features include principal component analysis (PCA) and linear discriminant analysis (LDA), and typical local features are local binary pattern (LBP) and Gabor wavelets.

While these face representation methods have achieved encouraging recognition performance in controlled environments, their performance is still far from satisfactory in unconstrained environments. Moreover, most of them are hand-crafted, which usually require strong priors to engineer them by hand.

Hence, how to extract robust and discriminative features to enlarge the inter-personal margins and reduce the intra-personal variations simultaneously remains a central and challenging problem in face recognition. In this paper, they propose a compact binary face descriptor (CBFD) feature learning method for face representation. Inspired by the fact that binary codes are robust to local changes such as varying illuminations and expressions, they aim to learn compact binary codes directly from raw pixels for face representation. Specifically, they learn a feature mapping to project each local pixel difference vector (PDV) into a low-dimensional binary vector, where the variance of all binary codes in the training set is maximized so that the redundancy information in PDVs is removed. To make the learned binary codes compact, they expect that the loss between original PDVs and learned binary codes is minimized and the learned binary codes are evenly distributed at each bin. Then, they cluster and pool these compact binary codes to obtain a histogram representation of each face image.

Moreover, they propose a coupled CBFD (C-CBFD) method to reduce the modality gap at the feature level for heterogeneous face matching. The contributions of this work are summarized as follows:

1. An unsupervised feature learning method to learn compact binary feature descriptor for face representation. With the learned feature filter, the redundancy information of the original raw pixels is removed in the obtained binary codes.

2. A coupled learning method to learn compact binary face descriptor for heterogeneous face matching. With the learned coupled filters, a common discriminative binary feature space is obtained and the modality gap of heterogeneous faces is greatly reduced at the feature level.

3. Apply CBFD and C-CBFD to learn face features in a local manner so that position-specific

information is exploited in the learned features.

4. Conducted extensive face recognition experiments on five widely used face datasets to demonstrate the efficacy of our proposed methods. Experimental results show that our methods are superior to most state-of-the-art face descriptors in both homogeneous and heterogeneous face recognition.

"Smart Eye" attempts to overcome above mentioned short comings by implementing a framework using Local Binary Pattern algorithm. V. Liong et al. [5] discussed such framework. With the advancement in the field of the face recognition it will help to identify the person. The facial recognition system presented in this review article contributes a resilient face recognition model based on the Local Binary pattern which captures the behavioural characteristics with the physiological biometric characteristics. The local binary pattern and its various types can be very helpful in the ongoing development on this topic forward. Over the decade the LBP algorithm has evolved faster as compared to the other face recognition algorithm which makes it perfect and efficient one. Also the advancement of hardware technology is making the execution of this algorithm faster as compared to its predecessors.

3.5 Summary

Steven Fernandez et al. [1], suggested a face detection system using Raspberry Pi which is compatible for Smart Eye. Fernan Gallego et al. in [2] embedded the security cameras into the computer GPU, but were unable to recognise faces.. "Smart Eye" uses Raspberry pi zero W board which can perform both face recognition and detection.Ping Hsin lee et al. in [3], attempted to use Fisher linear discriminant (FLD), and Eigen face decomposition whereas P. Viola et al. in [4] discussed rapid object detection method using a boosted cascade of simple features to detect face. Smart Eye uses LBP classifier for reasons such as time and resource constraints. V. Liong et al in [5] suggested LBP as a solution for face recognition. Given the constraints, this algorithm is best suited for Smart Eye.

Chapter 4

Analysis

4.1 Detailed Problem Statement

Home security has become a prime concern for everyone in recent times. The issue is not only to report the theft after it has occurred but to proactively stop it from happening.The facilities provided by the current system are not enough to prevent the crime from happening. The current system doesn't have a feedback loop to keep the concerned person aware of all the activities. Secondly, it is not capable in itself to detect any misconduct. It is very important to upgrade the security level whether it is at home, office or school.

With latest technology coming in handy for real time applications, an effective, accessible and affordable home security system is need of the hour.

4.2 Requirement Analysis

This section covers various function that are the main promises of the system. These functionalities constitute the minimum offerings of a complete system. There are two types of requirement for any system functional and non-functional requirement.

4.2.1 Functional Requirements

In software engineering and systems engineering, a functional requirement defines a function of a system or its component, where a function is described as a specification of behaviour between outputs and inputs. Following are the functional requirements of the system :

(I) Detects change in surveillance area

System is capable of detecting any change in the premises of user thorough comparison of images before and after any change occurs.

(II) Detects face in the image

System can differentiate between a random visitor and a family member by comparing face with the database image and accordingly notify the user through E-mail in case the person is unknown.

(III) Notification of the changes via email to end user

Whenever the system will find any random visitor instead of family members then system will capture the frame of detected human face and send it to authorized person via an email.

(IV) Prevent detection of family members as intruder

System is capable of detecting human face in the video frame and it also compare the image with images of family members. Whenever the system finds the captured image similar to the family member, it will not send it as notification to the authorised person.

4.2.2 Non Functional Requirement

This section specifies the required system quality factors that are not related to specific functional requirements documented in the use case module. These requirements are always meant to be fulfilled. Major non-functional requirements of the system are :

(I) Performance requirement: proper internet connectivity required for sending emails and receiving inputs by user.

(II) Adaptability: system will be able to adjust according the condition (like setup time-span, on off alarm) mention by user.

(III) Availability: system will be available 24x7 in working condition.

(IV) Correctness: System will find appropriate output of error detection, based on that it will notify user.

(V) Maintainability: User need to maintain (upload/remove) image folders whenever required.

(VI) Usability: System will be GUI based, therefore anybody can use it easily.

4.3 Use Case Analysis

A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved.
Use case diagram represented by Figure. 4.1 illustrates major functionalities like- Notification, Live Video Streaming, Database Management.

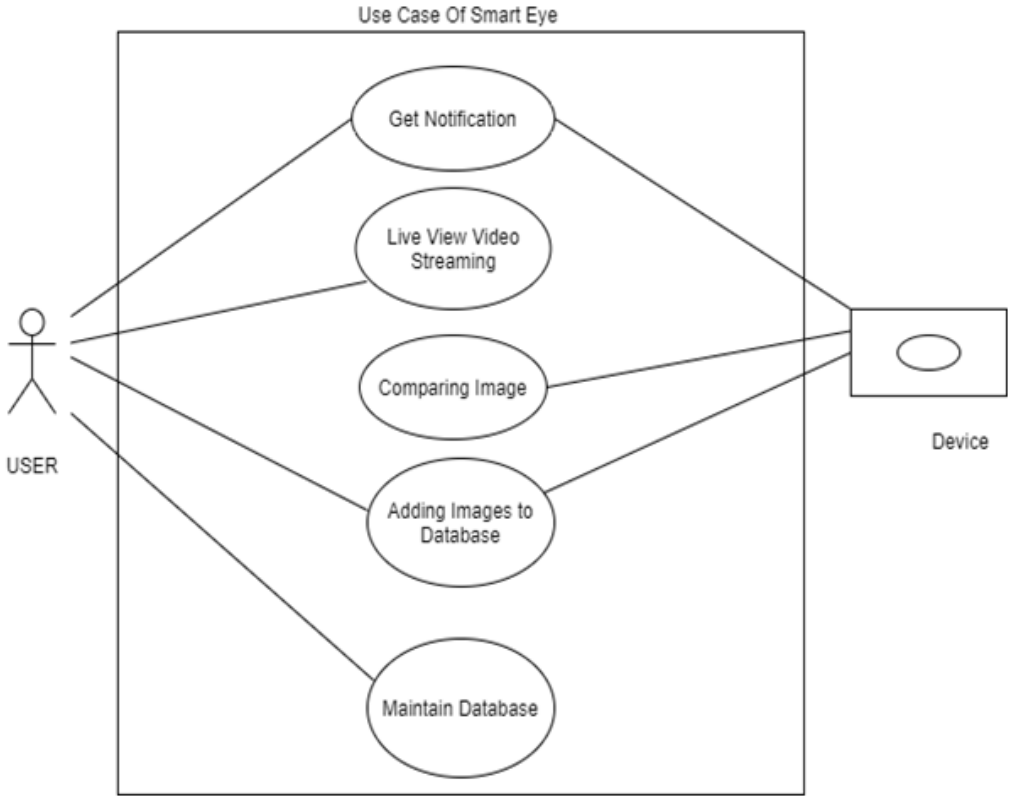


Figure 4.1: **Use Case Diagram of Smart Eye Security System**

(I) Actors

- User
- Device

(II) Roles and Responsibilities

Role: Device

Responsibility:

- Detect change in the surveillance area.
- Send notification.

Role: User

Responsibility:

- Get notification.

(III) Use Case Scenario

i. Get Notification

Use Case Name: Get Notification

Primary Actor: End User

Secondary Actor: Device

Pre-Condition: User and device must have internet connectivity.

Post-Condition: User will take decision on the basis of notification..

Flow of Event: System first detects change in surveillance area and clicks image then it detects face from image and compare it with base image if alteration is found then it will send notification to authorized user.

ii. View Live Video Streaming

Use Case Name: View Live Video Streaming.

Primary Actor: End User

Secondary Actor: Device

Pre-Condition: User must have access to watch live video stream.

Post-Condition: User will be able to watch live video streaming anytime and from anywhere.

Flow of Event: System takes images frequently which then convert into continues video signal which passes through http live streaming protocol which in turn enable user to watch live video stream.

iii. Comparing Images

Use Case Name: Comparing images.

Primary Actor: Device

Pre-Condition: There must be a random person in the video frame to capture the image.

Post-Condition: System should have images in the database to compare.

Flow of Event: System will compare the captured image with the database image and when it find it different it will send it to authorised person.

iv. Adding Image to Database

Use Case Name: Adding image to database.

Primary Actor: End User

Pre-Condition: There must have enough memory in the system to store image.

Post-Condition: User will be able to add image of their family member.

Flow of Event: User first registers himself/herself into the database then he/she login into the database then he/she select set of images of their family members and upload it into the database with proper naming convention.

v. Maintain Database

Use Case Name: Maintain Database.

Primary Actor: End User

Pre-Condition: User must authorized to maintain database.

Post-Condition: User will be able to upload, delete and modify database.

Flow of Event: User first login into the database/dashboard after entering valid user credentials then he/she will perform operation like adding, deleting or modification in the database.

4.4 System Requirements

Resources are the means that are used to achieve project objectives. The main grouping of resources needed in this system is HARDWARE and SOFTWARE requirement.

(I) LIST OF HARDWARE USED

- Raspberry pi zero w
- Raspberry pi NoIR camera Module
- Ada fruit raspberry pi zero w camera cable

(II) LIST OF SOFTWARE USED

- Operating System (Linux 5.3.1).
- Raspbian pi zero w.
- Open CV 4.1.1.

4.5 Data Flow Diagram

A data-flow diagram is a way of representing a flow of a data of a process or a system. The DFD also provides information about the outputs and inputs of each entity and the process itself. A data-flow diagram has no control flow, there are no decision rules and no loops.

4.5.1 Data Flow Diagram(Level 0)

Level 0 is a top level (also known as “context diagram”) data flow diagram. It only contains one process node ("Process 0") that generalizes the function of the entire system in relationship to external entities.

Figure. 4.2 is the top level data flow diagram of the entire system (Level 0). This figure shows the flow of data between device and user. Device sends notification after detection of unusual activities.

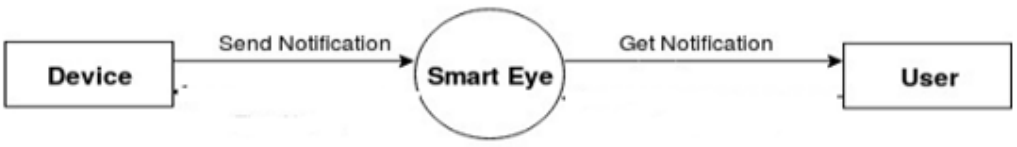


Figure 4.2: **Top level data flow diagram of the entire system (Level 0)**

4.5.2 Data Flow Diagram(Level 1)

A level 1 data flow diagram (DFD) is more detailed than a level 0 DFD. It breaks down the main processes into sub-processes that can then be analysed and improved on a more intimate level.
Figure 4.3 is the detailed Data flow diagram of the system with sub-processes (DFD Level 1).It breaks down the main processes into sub-processes namely detection, comparison and sending notification that can be analysed and improved on a more intimate level.

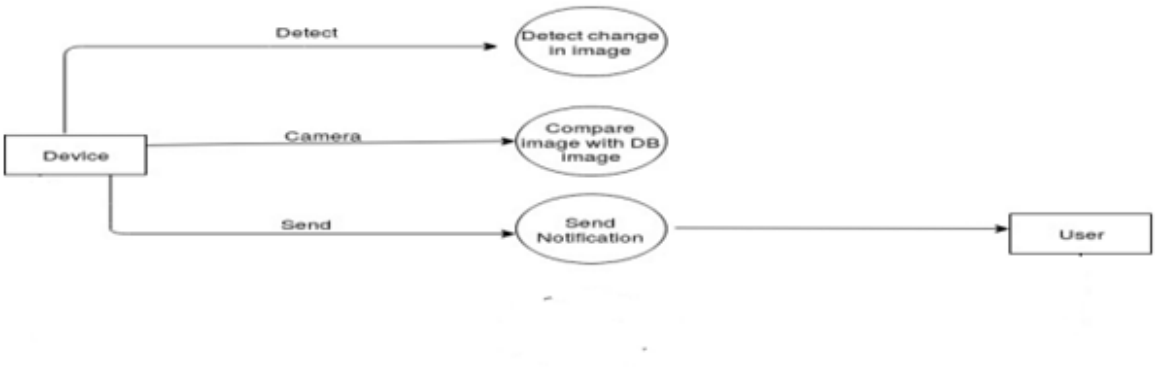


Figure 4.3: **Detailed Data flow diagram of the system with sub-processes (DFD Level 1)**

4.6 Feasibility Study

Feasibility study is an assessment of the practicality of a proposed project or system. A feasibility study aims to objectively and rationally uncover the strengths and weaknesses of an existing or proposed solutions, opportunities and threats present in the natural environment, the resources required to carry through, and ultimately the prospects for success. In its simplest terms, the two criteria to judge feasibility are cost required and value to be attained.

- **Technical Feasibility** -It is an evaluation of the hardware and software and how it meets the need of the proposed system.
  All the hardware and software tools used in this system are easily available. They are easy to understand and easy to get expertise.

- **Operational Feasibility** -Operational feasibility is the measure of how well a proposed system solves the problems, and takes advantage of the opportunities identified during scope definition.
  All the components used in this system are maintainable, reliable and replaceable at nominal cost.

- **Economic Feasibility** -The purpose of an economic feasibility study is to demonstrate the net benefit of a proposed project for accepting or disbursing funds or benefits.

  A simple economic analysis which gives the actual comparison of costs and benefit are much meaningful in this case. The total cost of smart eye comes out to be in the range of Rs.4000-5000.

Chapter 5

Design

5.1 System Architecture

A system architecture or systems architecture is the conceptual model that defines the structure, behaviour, and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviours of the system.
Figure 5.1 is the Architecture Diagram of the System, the major components shown in this figure are Raspberry Pi NoIR camera Module, Raspberry Pi Zero W Board and User.

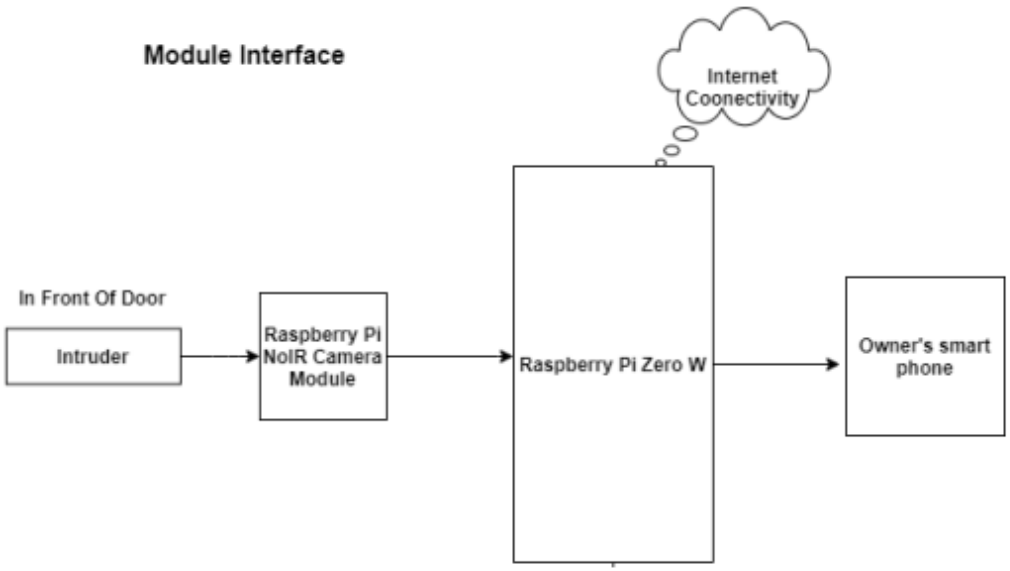


Figure 5.1: **Architecture diagram of the System**

The architecture diagram describes the system as whenever any intruder arrives in front of door then system capture the frames of detected face by Raspberry pi NoIR Camera module which is attached with raspberry pi Zero W and process the image by comparing it with family member's image. When system detects any difference between the images, it will send that image to owner's smart phone which should be connect through the internet.

5.2 Activity Diagram

Activity diagram is used to describe the dynamic aspects of the system. Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. The control flow is drawn from one operation to another.

Figure 5.2 is the Activity Diagram of the System. This figure shows the flow of activities respectively detecting an unknown person, comparing clipped image and then sending notification.

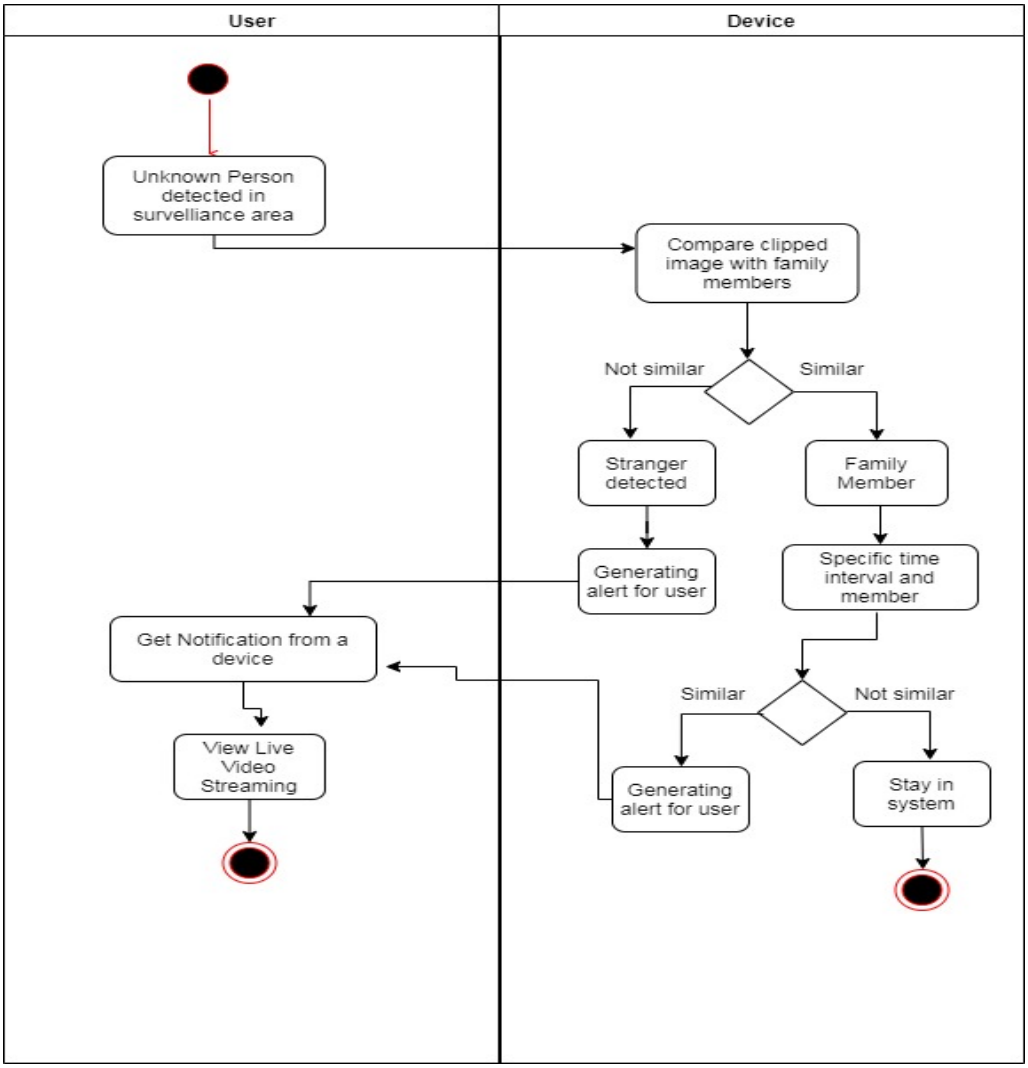


Figure 5.2: **Activity Diagram of System**

The above diagram represents the flow from one activity to another activity.Initially an unknown person arrives in front of camera. Camera captures the image, and compares clipped images with family member's image.

If son/daughter of the user comes in the specific time interval, send notification to the user. If the image doesn't match, appropriate notification is sent.

5.3 State Chart Diagram

A state diagram is a type of diagram used to describe the behaviour of systems. State diagrams require that the system described is composed of a finite number of states, sometimes, this is indeed the case, while at other times this is a reasonable abstraction.

Figure 5.3 is the State Chart Diagram of the system. This figure represents various states of system and their transition. These states are: taking initial image, detecting, comparing and sending notification.

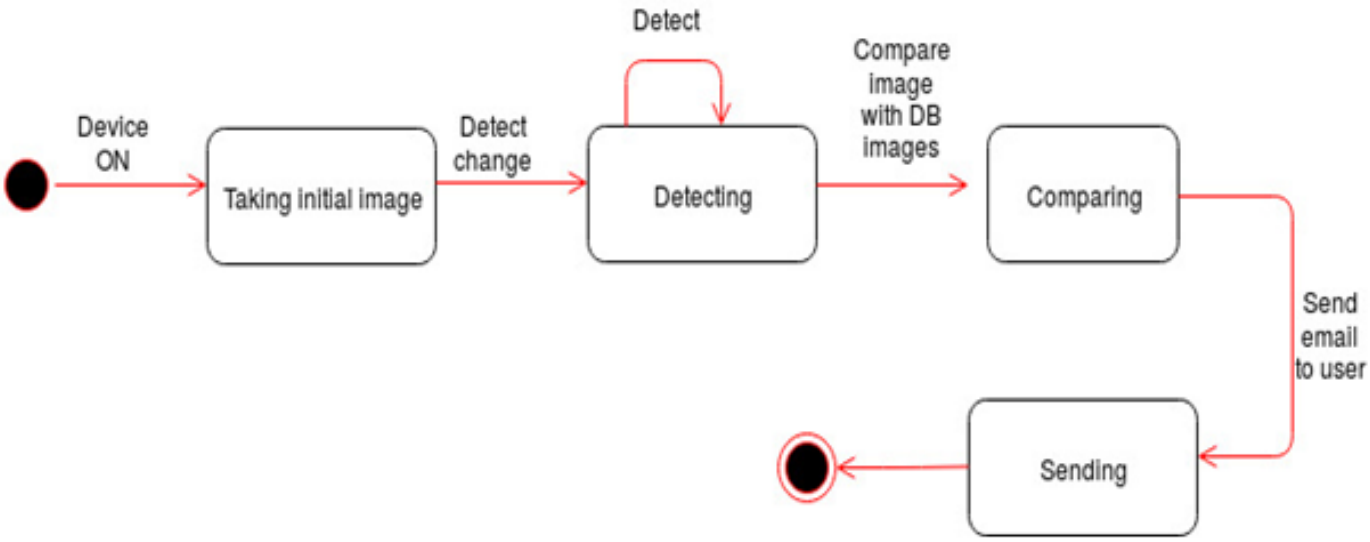


Figure 5.3: **State Chart Diagram of System**

The above diagram shows the behaviour of systems.There are a total of 4 states,and the first state indicates that the device is ready to take an image and on the next stage it detects an image and compares with family member's image on at third state. In last state it sends notification if it detects an intruder or if the son/daughter of the user arrives in a specific time interval.

5.4 Algorithm

Local Binary Pattern classifier algorithm is used for face detection. Detailed description of the algorithm is given below.

Face Detection

- Face detection is a technique that identifies or locates human faces in digital images.
- Face detection is different from Face recognition.
- Face detection detects merely the presence of faces in an image while facial recognition involves identifying whose face it is.
- Face detection is performed by using classifiers.
- A classifier is essentially an algorithm that decides whether a given image is positive(face) or negative(not a face).
- OpenCV already has two pre-trained face detection classifiers, which can readily be used in a program.
- The two classifiers are:

  i Haar Classifier

  ii Local Binary Pattern classifier.

Local Binary Pattern classifier Local Binary Pattern (LBP) is a simple yet very efficient texture operator which labels the pixels of an image by threshold the neighbourhood of each pixel and considers the result as a binary number.

1. **Parameters:** The LBPH uses 4 parameters

   - **Radius**:The radius is used to build the circular local binary pattern and represents the radius around the central pixel. It is usually set to 1.
   - **Neighbours**:The number of sample points to build the circular local binary pattern. Keep in mind: the more sample points you include, the higher the computational cost. It is usually set to 8.
   - **Grid X**: The number of cells in the horizontal direction. The more cells, the finer the grid, the higher the dimensionality of the resulting feature vector. It is usually set to 8.

- **Grid Y**: The number of cells in the vertical direction. The more cells, the finer the grid, the higher the dimensionality of the resulting feature vector. It is usually set to 8

2. **Training the Algorithm**

   First, training of the algorithm is done. To do so, a dataset is used with the facial images of the people we want to recognize. Then set an ID (it may be a number or the name of the person) for each image, so the algorithm will use this information to recognize an input image and give an output. Images of the same person must have same ID.

3. **Applying the LBP operation**

   The first computational step of the LBPH is to create an intermediate image that describes the original image in a better way, by highlighting the facial characteristics. The algorithm uses a concept of a sliding window, based on the parameters radius and neighbours.

   Figure 5.4 shows intermediate image of the original image in which decimal value of the generated binary number is then used for labeling the given pixel.

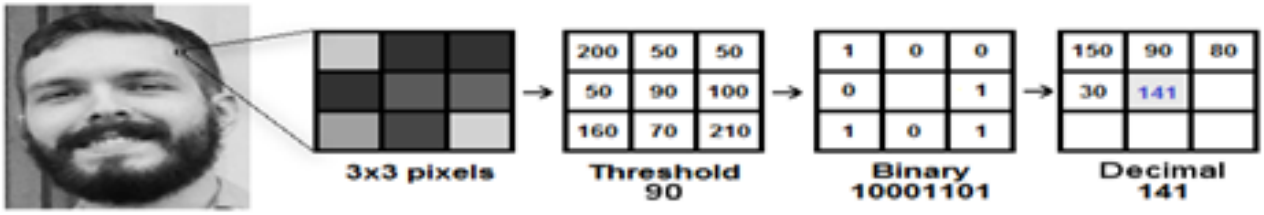


Figure 5.4: **Intermediate image of the original image**

Figure 5.5 shows pixels from captured image.Based on the image above, breaking functions into several small steps make it comprehensive:

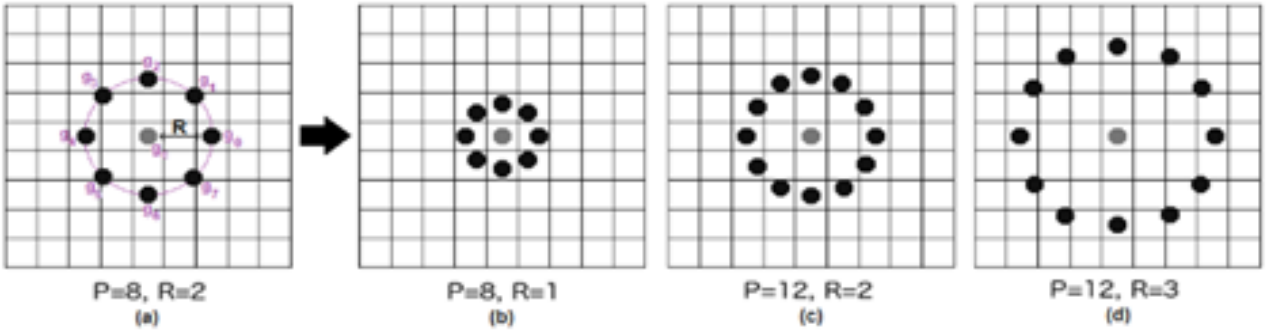


Figure 5.5: **Represent pixel from the original image**

- Suppose there is a facial image in gray-scale.
- Get part of this image as a window of 3x3 pixels.
- It can also be represented as a 3x3 matrix containing the intensity of each pixel (0 255).
- Then, take the central value of the matrix to be used as the threshold.
- This value will be used to define the new values from the 8 neighbours.
- For each neighbour of the central value (threshold), set a new binary value. Set 1 for values equal or higher than the threshold and 0 for values lower than the threshold.
- Now, the matrix will contain only binary values (ignoring the central value). Concatenate each binary value from each position from the matrix line by line into a new binary value (e.g. 10001101). Note: some authors use other approaches to concatenate the binary values (e.g. clockwise direction), but the final result will be the same.
- Then, convert this binary value to a decimal value and set it to the central value of the matrix, which is actually a pixel from the original image.
- At the end of this procedure (LBP procedure), there is a new image which represents better the characteristics of the original image.

4. Extracting the Histograms

Now, using the image generated in the last step, use the Grid X and Grid Y parameters to divide the image into multiple grids.

Figure 5.6 shows histogram representation of the images as can be seen in the above image Based on the image above, extract the histogram of each region as follows:

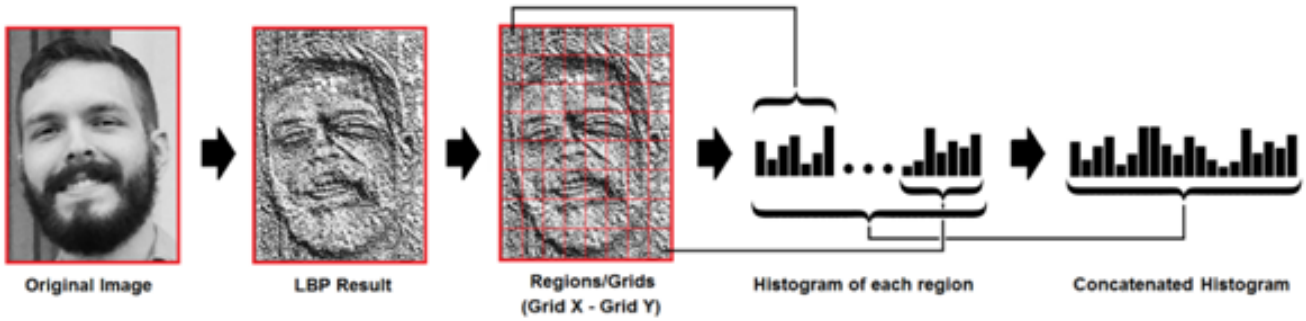


Figure 5.6: **Histogram of the original image**

- There is an image in gray-scale, each histogram (from each grid) will contain only 256 positions (0 255) representing the occurrences of each pixel intensity.
- Then, concatenate each histogram to create a new and bigger histogram. Supposing we have 8x8 grids, we will have 8x8x256=16.384 positions in the final histogram. The final histogram represents the characteristics of the image original image.

5. Performing the Face Recognition

In this step, the algorithm is already trained. Each histogram created is used to represent each image from the training dataset. So, given an input image, perform the steps again for this new image and create a histogram which represents the image.

- So to find the image that matches the input image we just need to compare two histograms and return the image with the closest histogram.
- Various approaches can be used to compare the histograms (calculate the distance between two histograms), for example: **euclidean distance, chi-square, absolute value**, etc. In this example, the Euclidean distance is used (which is quite known) based on the following formula:

$$D = \sqrt{\sum_{i=1}^{n}(hist1_i - hist2_i)^2}$$

- So the algorithm output is the ID from the image with the closest histogram. The algorithm should also return the calculated distance, which can be used as a **'confidence'** measurement. **Note:**

don't be fooled about the 'confidence' name, as lower confidences are better because it means the distance between the two histograms is closer.

- Then use a threshold and the 'confidence' to automatically estimate if the algorithm has correctly recognized the image. Assume that the algorithm has successfully recognized if the confidence is lower than the threshold defined.

Chapter 6

Implementation

6.1 Module / Package Description

Major modules of the system is written in Python. Detailed description of each module is given below. System majorly contains four modules named as:

- Main.py
- Camera.py
- Compare Transition.py
- Mail.py

1. **Main.py**

   This module initiates the system. All the major libraries are imported here. This libraries are used by the system for sending e-mail, comparing face, video streaming. All major functions of the system will run by main.py module.

2. **Camera.py**

   After the system is initiated by the main.py module, the remaining libraries used by the camera is imported here like **PiVideoStream** from **imutils.video.pivideostream** package. Now camera starts streaming.

   - **imutils.video.pivideostream**: This package is used from the OpenCV in python to perform the video streaming.

3. **Compare Transition**:

   Now that the system is working, transitions occurring in the frame are caught by the camera. For this task module uses FacePP, exceptions imported from facepplib package. For comparing the transition, a transfer function is defined for transferring images to ftp server and then a face comparing function was defined for comparison.

   - **Facepplib**: This module is for communicating with Face++ facial recognition service. It is used in python to find the transition in the face.

4. **Mail.py**

   After comparing between the image of family member and clipped image whenever the system detects any difference it sends the clipped image as a notification to the authorized person. It requires send mail package from the mail module, which was imported in the main.py, to perform this task. Here set the Mail id and password of authorized person to receive mail from the system.

   - **Mail**: It is a package used in python to get the mail.

6.2 Deployment Diagram

A deployment diagram is a UML diagram type that shows the execution architecture of a system, including nodes such as hardware or software execution environments, and the middle ware connecting them. Deployment diagrams are typically used to visualize the physical hardware and software of a system.
Figure 6.1 shows deployment diagram of the system.This figure represents various nodes of system and their artifacts. These nodes are: FTP server, Smart eye (hardware node), Remote File System, Wireless Router etc.

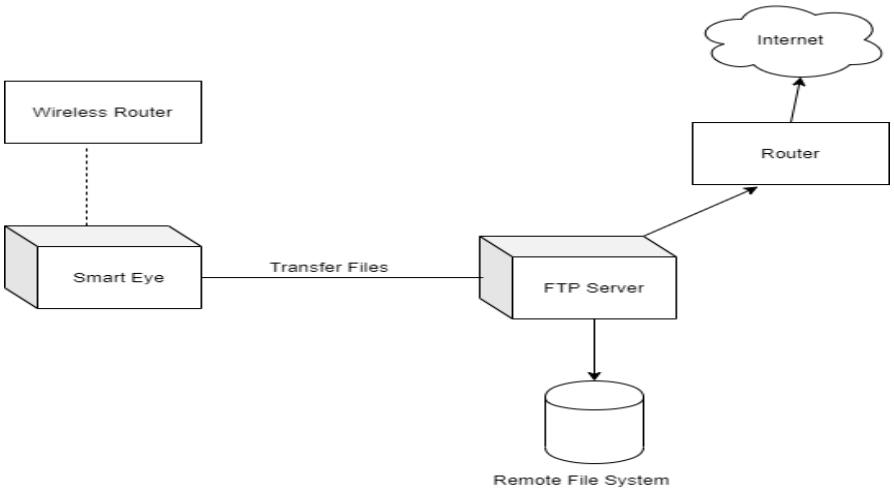


Figure 6.1: **Deployment diagram of system**

6.3 Implementation Diagram

Implementation diagram allows to model the flow of various component or modules involved in the system. It decomposes system into various modules which shows how these modules are interacting with each other. Figure 6.2 showing the flow of various modules included in smart eye.

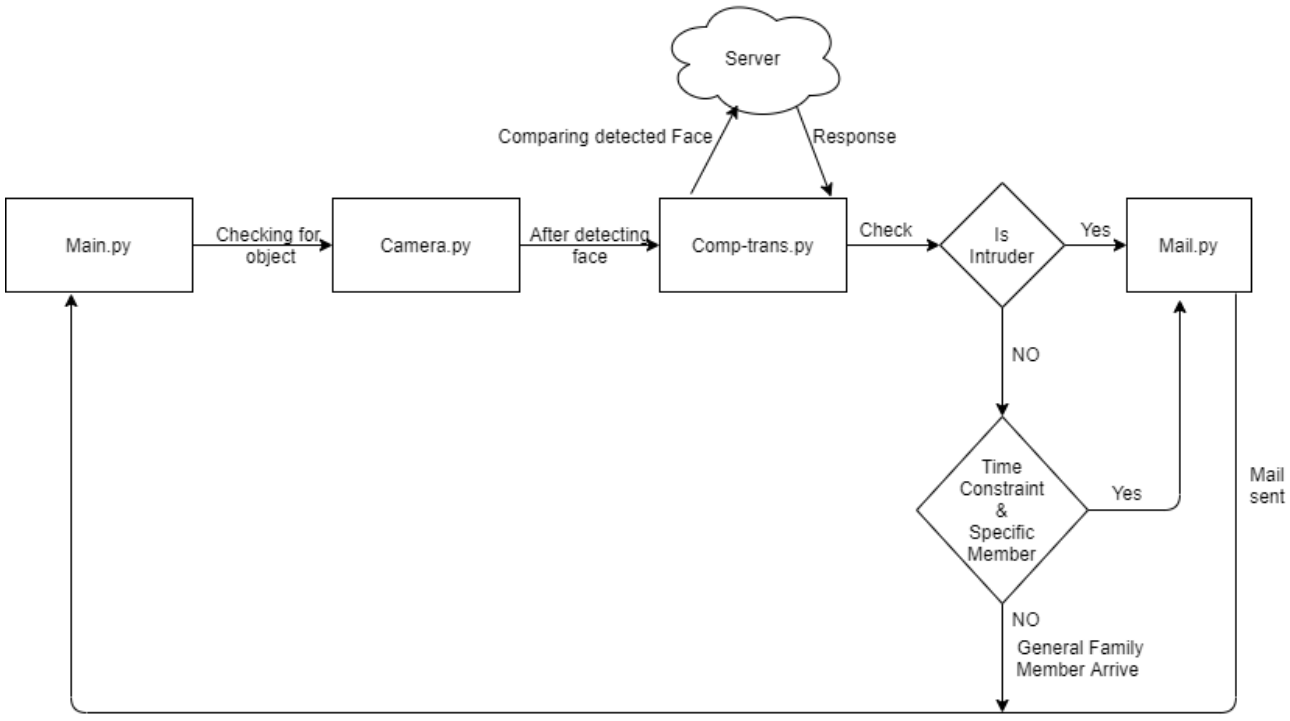


Figure 6.2: **Implementation diagram of system**

Intially main.py initialises the camera.py file by calling `check_for_object()` method, which after detecting object, call `comp_trans.py` file, which will transfer detected file and compare its face from the images stored in ftp server. Ftp server give response to `comp_trans.py` file which then check condition for intruder and specific family member. If these condition found true then `comp_trans.py` file gives its control to mail.py which will sent mail to authorise user.

And if the above condition are false that means the member is family member so no mail will be sent. After completing the cycle whole process control comes back to main.py and the iteration starts again

6.4 Code Description

Python is used to implement the major modules of the project. Step by step description of the code is given below.

Accessing the Raspberry Pi Camera with OpenCV and Python

1. **Start Raspberry Pi camera board module.**

2. **Enable your camera module.**

   $ sudo raspi-config

3. **Test out the camera module.**

   $ raspistill -o output.j

4. **Install picamera.**

   # Check for OpenCV

   $ source /.profile

   $ workon cv

   # Install picamera

   $ pip install "picamera[array]"

   $ pip install "picamera[array]"

5. **Accessing a single image of your Raspberry Pi zero W using Python and OpenCv.**

   # import the necessary packages

   import cv2

```
import sys
from mail import sendEmail
from flask import Flask, render_template, Response
from camera import VideoCamera
from flask_basicauth import BasicAuth
import time
import threading
```

#Function to initialize the camera setup and checking for the object.

```
def check_for_objects():
global last_epoch
while True:
try:
frame, found_obj = video_camera.get_object(object_classifier)
print("smart eye watching.....")
if found_obj and (time.time() - last_epoch) > email_update_interval:
last_epoch = time.time()
print("Smart Eye found object")
print ("Sending email to user...")
sendEmail(frame)
print ("done!")
except:
print ("Error sending email: ", sys.exc_info()[0])
```

#Function to get the response as a image from video camera.

```
@app.route('/')
@basic_auth.required
def index():
return render_template('index.html')
def gen(camera):
while True:
frame = camera.get_frame()
@app.route('/video_feed')
def video_feed():
```

return Response(gen(video_camera),

mimetype='multipart/xmixedreplace; boundary=frame')

if __name__ == '__main__':

t = threading.Thread(target=check_for_objects, args=())

t.daemon = True

t.start()

app.run(host='0.0.0.0', debug=False)

6. Accessing the camera for live video streaming.

#import the necessary packages

import cv2

from imutils.video.pivideostream import PiVideoStream

import imutils

import time

import numpy as np

 # function for video streaming.

class VideoCamera(object):

def __init__(self, flip = False):

self.vs = PiVideoStream().start()

self.flip = flip

time.sleep(2.0)

def __del__(self):

self.vs.stop()

def flip_if_needed(self, frame):

if self.flip:

return np.flip(frame, 0)

return frame

def get_frame(self):

frame = self.flip_if_needed(self.vs.read())

ret, jpeg = cv2.imencode('.jpg', frame)

return jpeg.tobytes()

```
def get_object(self, classifier):
found_objects = False
frame = self.flip_if_needed(self.vs.read()).copy()
gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
objects = classifier.detectMultiScale
(
gray,
scaleFactor=1.1,
minNeighbors=5,
minSize=(30, 30),
flags=cv2.CASCADE_SCALE_IMAGE
)
if len(objects) > 0:
found_objects = True


for (x, y, w, h) in objects:
cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 255, 0), 2)
ret, jpeg = cv2.imencode('.jpg', frame)
return (jpeg.tobytes(), found_objects)
```

7. Face Comparison

```
# define transfer function for transfer images to ftp server
def transfer(frame):
dir_list=[]
try:
print("...enter into file transfer mode.....")
ftp=FTP('files.000webhost.com',timeout=120)
ftp.login(user='smarteyesgsits',passwd='Root@1234')
ftp.cwd("/public_html/images/")
ftp.dir()
ftp.retrlines('NLST',dir_list.append)
print(dir_list)
```

```
with open("a.jpg", "wb") as img:
img.write(frame)
local=open('/home/pi/Downloads/SmartEye/a.jpg','rb')
ftp.storbinary('STOR a.jpg',local)
c=face_comparing(app_,dir_list)
ftp.close()
if c==True:
return True
else:
return False
except:
#print("error in comp_trans file")
print("Oops!",sys.exc_info()[0],"occured.")
# define face comparing function
def face_comparing(app,dir_list):
print()
print('-'*30)
print('Comparing Photographs......')
print('-'*30)
img='https://smarteye-sgsits.000webhostapp.com/images/'
i=3
while(i<len(dir_list)):
image1=img +''.join(dir_list[i])
print(image1)
#image stores
image2='https://smarteye-sgsits.000webhostapp.com/images/a.jpg'
cmp_ = app.compare.get(image_url1 = image1,image_url2 = image2)
print('Photo1', '=', cmp_.image1)
print('Photo2', '=', cmp_.image2)
i=i+1
# Comparing Photos
if cmp_.confidence > 70:
```

print('Both photographs are of same person.....')

return True

else:

pass

print('Both photographs are of two different persons.....')

return False

8. Sending Email.

#import the necessary packages

import smtplib

from email.mime.multipart import MIMEMultipart

from email.mime.text import MIMEText

from email.mime.image import MIMEImage

fromEmail = 'harish.lohare1998@gmail.com'

fromEmailPassword = 'deepchand1998'

toEmail = 'hdlohare98@gmail.com'

#toEmail = 'yashpathakmmps1@gmail.com'

 # Function to send an Email.

def sendEmail(image):

msgRoot = MIMEMultipart('related')

msgRoot['Subject'] = 'Security Update'

msgRoot['From'] = fromEmail

msgRoot['To'] = toEmail

msgRoot.preamble = 'Smart Eye Security Camera Update'

msgAlternative = MIMEMultipart('alternative')

msgRoot.attach(msgAlternative)

msgText = MIMEText('Smart security camera found object') #convert plain text into mime text

msgAlternative.attach(msgText) #attach msg within a mail msgText = MIMEText('<img src="cid:image1">', 'html')

#image address converted as mime type

msgAlternative.attach(msgText) #attach image address within a mail

msgImage = MIMEImage(image)

```
msgImage.add_header('Content-ID', '<image1>')
msgRoot.attach(msgImage)
smtp = smtplib.SMTP('smtp.gmail.com', 587)
smtp.starttls()
smtp.login(fromEmail, fromEmailPassword)
smtp.sendmail(fromEmail, toEmail, msgRoot.as_string())
smtp.quit()
```

Chapter 7

Testing and Results

7.1 Testing

Testing is a process, to evaluate the functionality of a system or software application with an intent to find whether the developed system or software met the specified requirements or not and to identify the defects to ensure that the product is defect-free in order to produce the quality product. The subsequent test cases of our system are as follows:

(I) Test Case - I

Test Case Name : Face detection.

System : Raspberry pi zero w board and Raspberry pi NoIR camera module.

Description : Test the system's capability of detecting face from multiple angle.

Pre condition :

(a) System should be on

(b) Presence of human to detect face

Table 7.1: **Specifications of Test Case-I**

Step	Action	Expected System	Pass/Fail
1	Move camera in front of face	Detect full view of face	Pass
2	Move camera at an angle of 45	Detect 45 degree view of face	Pass
3	Move camera at an angle of 90	Detect 90 degree view of face	Pass

Post condition : Faces from different angle captured.

(II) Test Case-II

Test Case Name : Registration.

System : 000webhost ftp server.

Description : Test login and registration of end user on 000webhost ftp server.

Pre condition :

(a) System of end user should be connected by internet

Table 7.2: **Specifications of Test Case-II**

Step	Action	Expected System	Pass/Fail
1	Login without registration	Login request will be rejected	Pass
2	Click on registration	User will be prompted to enter credentials	Pass
3	Enter credentials	Credentials will be validated by system	Pass
4	Submit credentials	Successful registration notification will be received	Pass

Post condition : End user now able to login into 000webhost ftp server.

(III) Test Case-III

Test Case Name : Upload Image

System : 000webhost ftp server

Description : Test the upload and delete feature of 000webhost server

Pre condition :

(a) Registration on 000webhost

(b) Internet connectivity

Table 7.3: **Specifications of Test Case-III**

Step	Action	Expected System	Pass/Fail
1	Login	User will be prompted to enter credentials	Pass
2	Click on Upload Image	User will be directed to upload image from system	Pass
3	Click on delete image	Selected image will be deleted	Pass

Post condition : Images of family member will be uploaded over ftp server

(IV) Test Case - IV

Test Case Name : Identify intruder/family member

System : Smart Eye

Description : Test system's capability to distinguish between intruder and family member

Pre Condition :

(a) System should be connected to power and internet

(b) Presence of user in front of camera

Table 7.4: **Specifications of Test Case-IV**

Step	Action	Expected System	Pass/Fail
1	Image of family member uploaded on ftp server then move camera over his/her face	Family member arrive	Pass
2	Image of family member deleted from ftp server then move camera over his/her face	Intruder detected, notification will be send to authorised user	Pass

Post condition : Smart Eye distinguish between family member and intruder and in case intruder, notification is sent to authorised user

(V) Test Case - V

Test Case Name : Send Notification

System : Smart Eye

Description : Test system's feature of sending notification in case intruder detected

Pre condition :

(a) System should be connected to power and internet

(b) Intruder should be detected

Table 7.5: **Specifications of Test Case-V**

Step	Action	Expected System	Pass/Fail
1	Detect face	Compare face	Pass
2	Intruder detects	Send notification	Pass

Post condition : Smart Eye detects intruder and sends notification to authorised user

(VI) Test Case - VI

Test Case Name : Get Notification

System : End User's system

Description : Test whether end user receiving notification or not

Pre condition :

(a) System should be connected to power and internet

(b) Intruder should be detected

Table 7.6: **Specifications of Test Case-VI**

Step	Action	Expected System	Pass/Fail
1	Check e-mail	Notification will be received	Pass

Post condition : Smart Eye distinguish between family member and intruder and in case intruder, notification is sent to authorised user

(VII) Test Case - VII

Test Case Name : Identify Intruder in Dim Light

System : Smart Eye

Description : Test system capability to detect human face/intruder in dim light.

Pre condition :

(a) System should be turn on

(b) Presence of human to detect face

(c) Light should be dim at night

Table 7.7: **Specifications of Test Case-VII**

Step	Action	Expected System	Pass/Fail
1	Move camera in front of face in dim light at night	Detect full view of face	pass
2	Move camera at an angle of 45 in dim light at night	Detect 45 degree view of face	pass
3	Move camera at an angle of 90 in dim light at night	Detect 90 degree view of face	pass

Post condition : Face from different angle in dim light at night is captured

7.2 Result

To check system for different tasks, different types of test cases are prepared in order to verify system compliance, for example : 90 degree view of face and differentiating between family members and intruders etc. It was found that system is defect free and producing results as expected and well under permissible time delay.

Chapter 8

Conclusion

The smart security facility developed under this project for home and other sensitive areas is manifested by using the available technologies. Hence they are easily available and are comparatively cheaper than other security systems. At an affordable price, this will ultimately empower a common city dweller against any mischief or attempt of theft on their property.

The system has achieved functionalities such as, detecting intruders and family members, sending notification and updating files in FTP server. The system detects changes within surveillance area and checks for intruders, if it is not a family member then it sends notification to the user to apprise about the situation. The whole process takes about 2-3 seconds, which are enough for taking any effective actions to stop an ongoing theft. The home owner will not remain a passive observer but will be able to act immediately to avert any misconduct on her property. These measures also works as a deterrent for mischief.

Limitations:

The system works under certain constraints such as, all time internet connectivity, uninterrupted power supply (although batteries can be used as backup), and no physical harm done to the device.

Future Scope:

Applications of the proposed security system are not limited to homes, it can be scaled up for vast surveillance area. To acquire the features of the proposed system, latest IoT technology are used which are easily available and replaceable, therefore there will be no burden on the part of the user for extensive maintenance.

References

[1] V Swathi and Steven Fernandes. “Raspberry Pi Based Human Face Detection”. In: *Journal of Advanced Research in Computer* 4.9 (2015).

[2] V. B. Saiz and F. Gallego. “GPU: Application for CCTV systems”. In: *2014 International Carnahan Conference on Security Technology (ICCST)*. 2014, pp. 1–4.

[3] Ping Hsin Lee, Vivek Srinivasan, and Arvind Sundararajan. *Face Detection, Final Year Project*. 2014.

[4] Paul Viola and Michael Jones. “Rapid object detection using a boosted cascade of simple features”. In: *Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001*. Vol. 1. IEEE. 2001, pp. I–I.

[5] J. Lu et al. “Learning Compact Binary Face Descriptor for Face Recognition”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37.10 (2015), pp. 2041–2056.

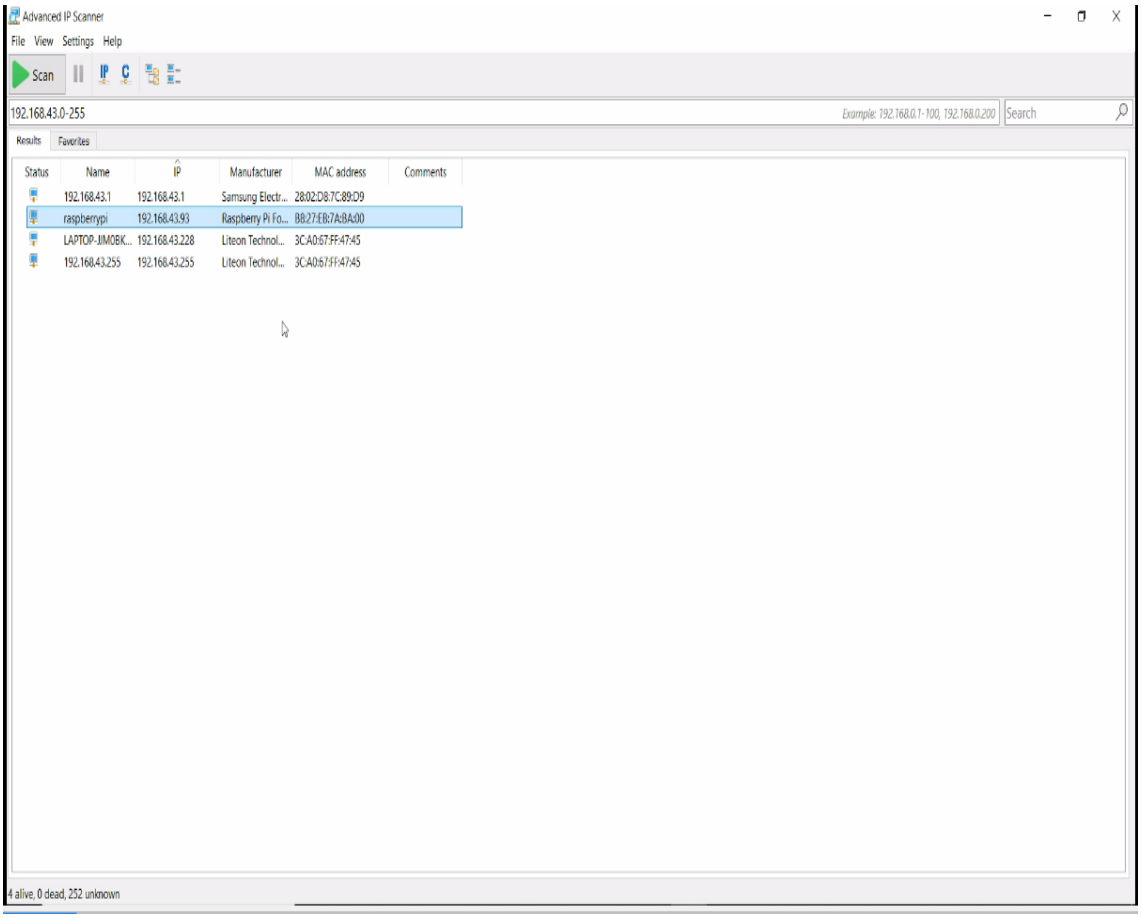
Appendix A

Screenshots

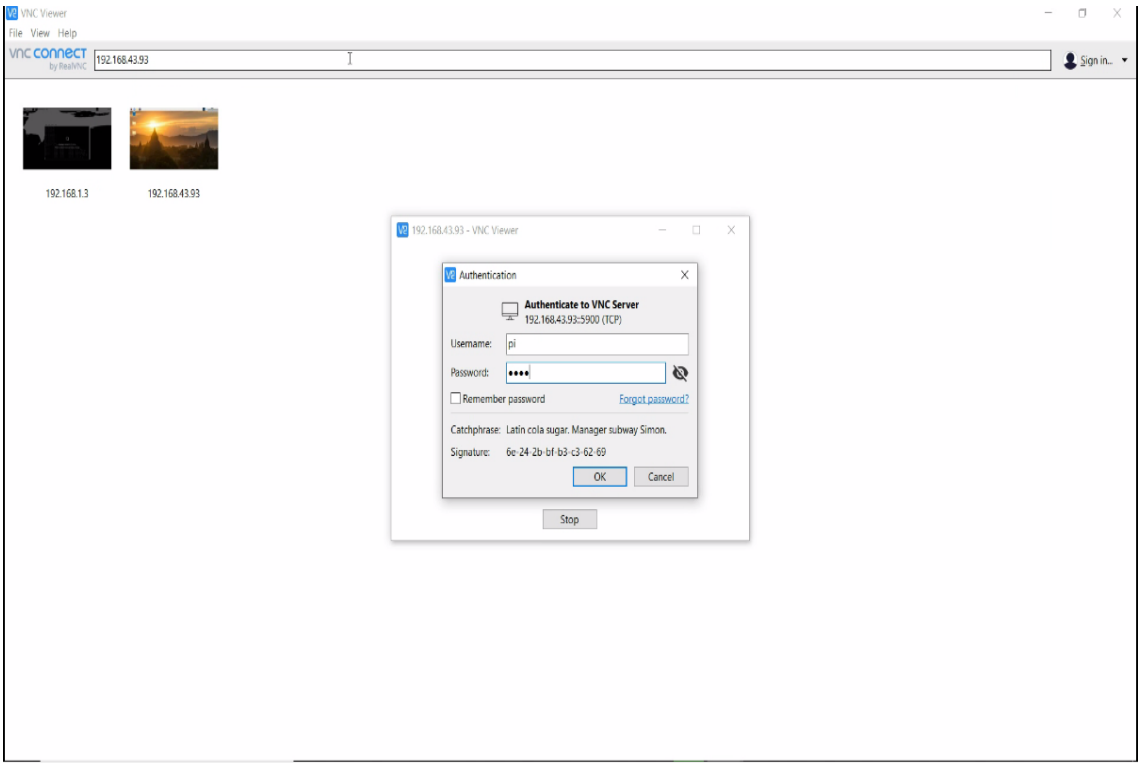
Smart Eye Camera Module

Interior of Camera Module

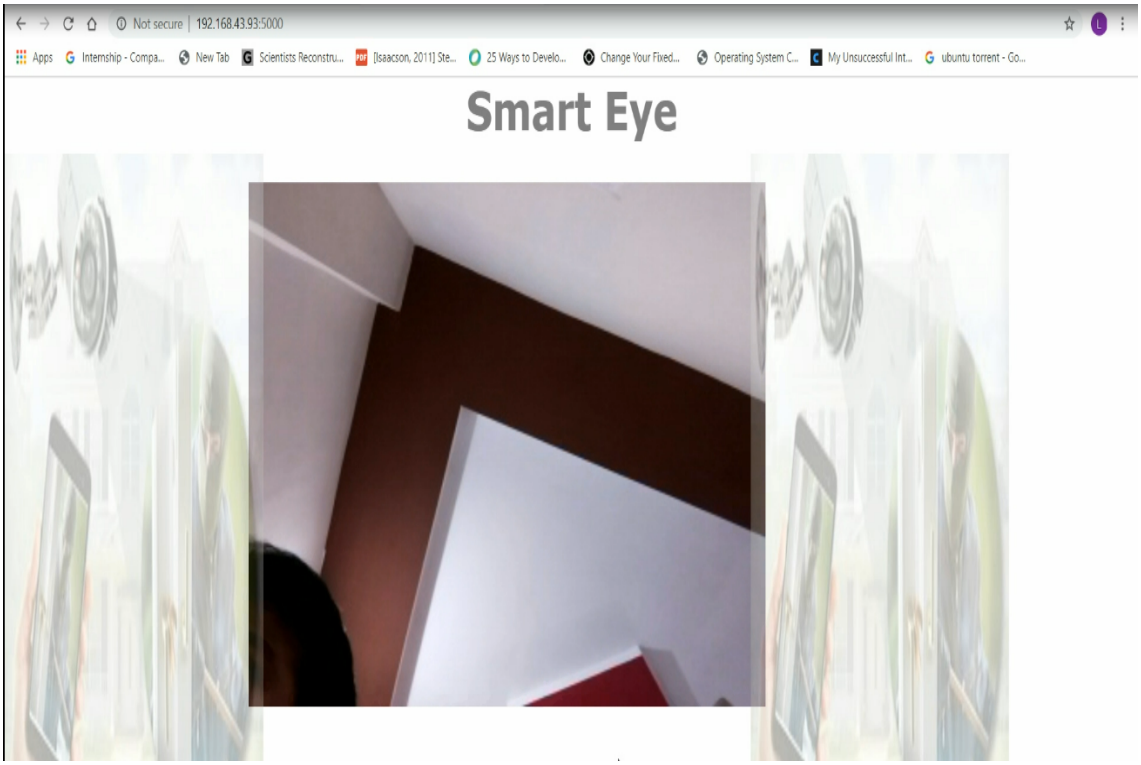
Raspberry Pi Connected to Laptop



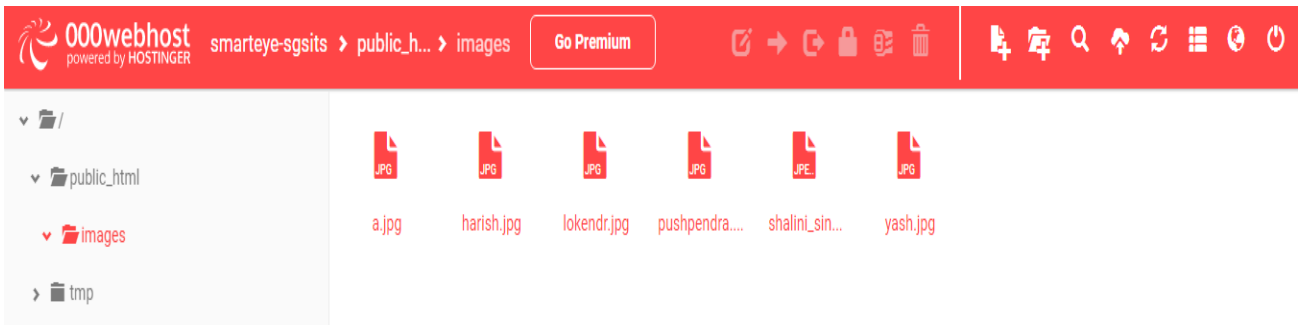
Getting PI's ip address through ip-scanner



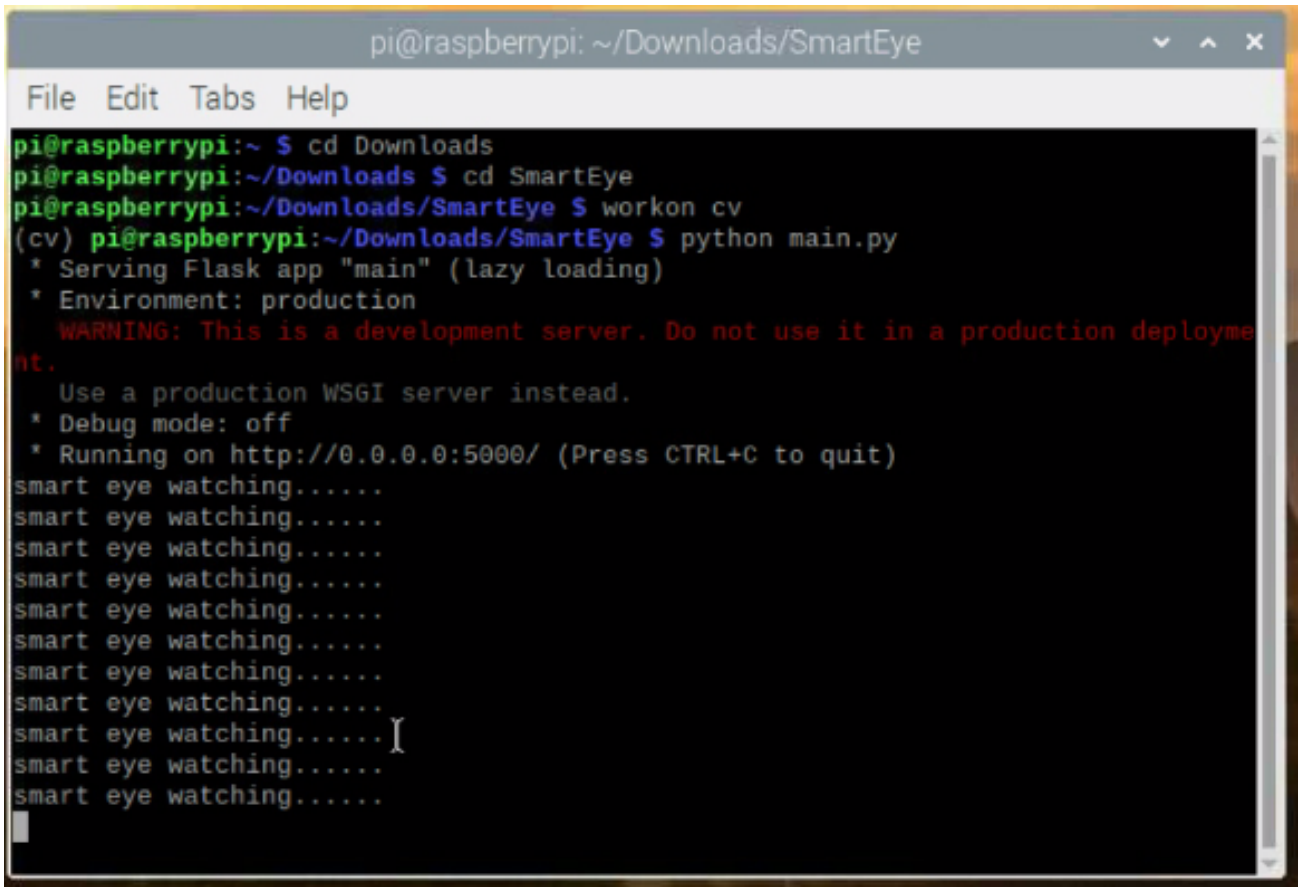
Remote Login in PI through VNC Viewer



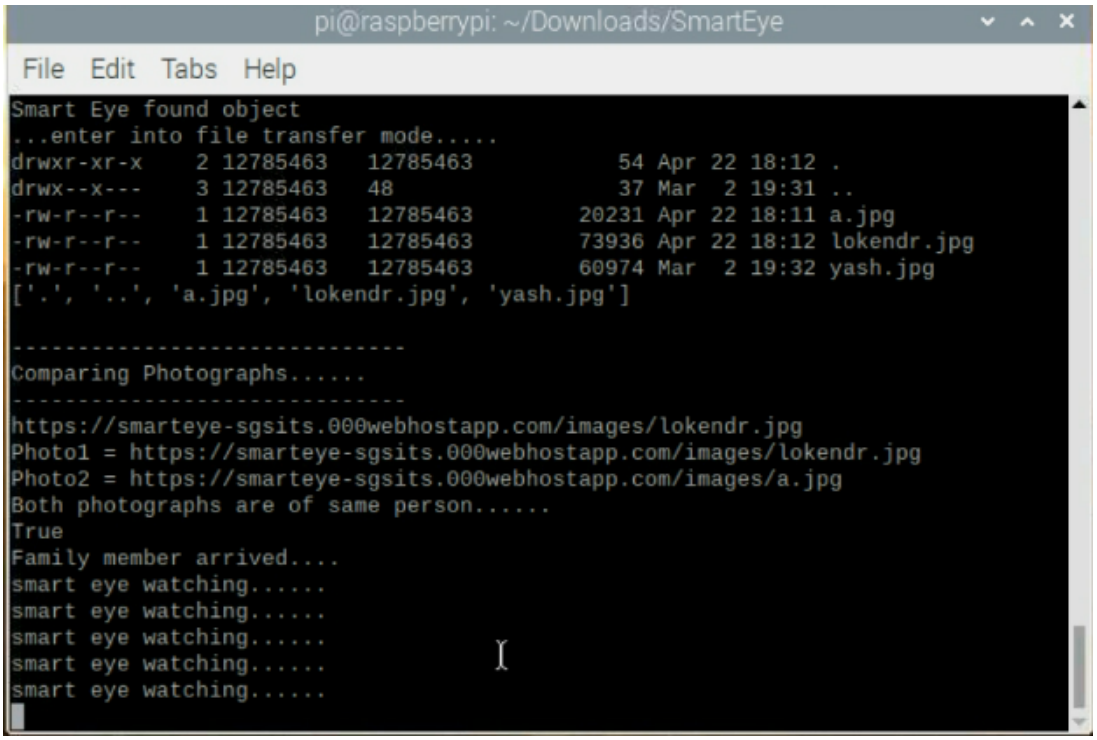
Live Video Streaming



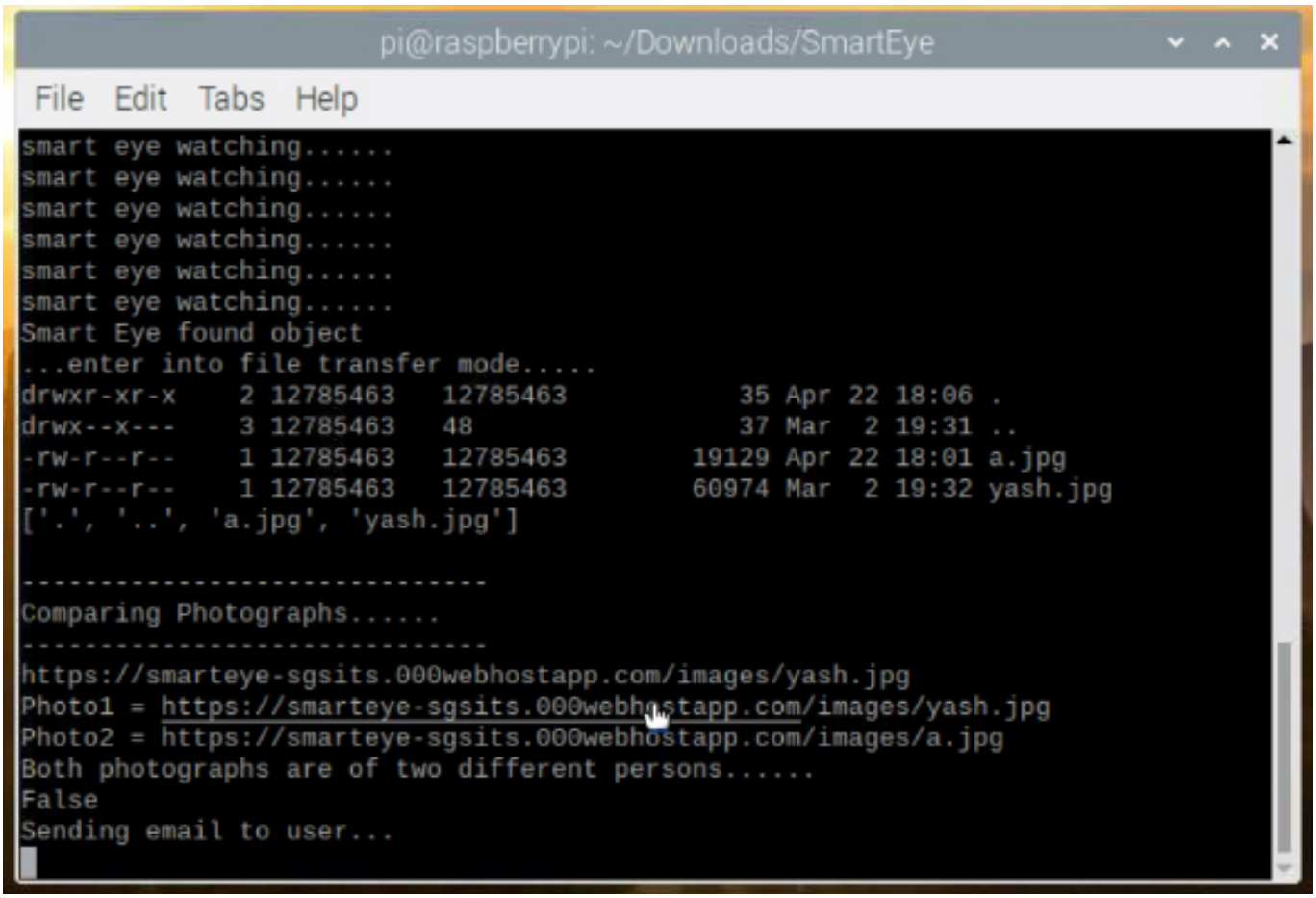
All family members image stored on FTP server



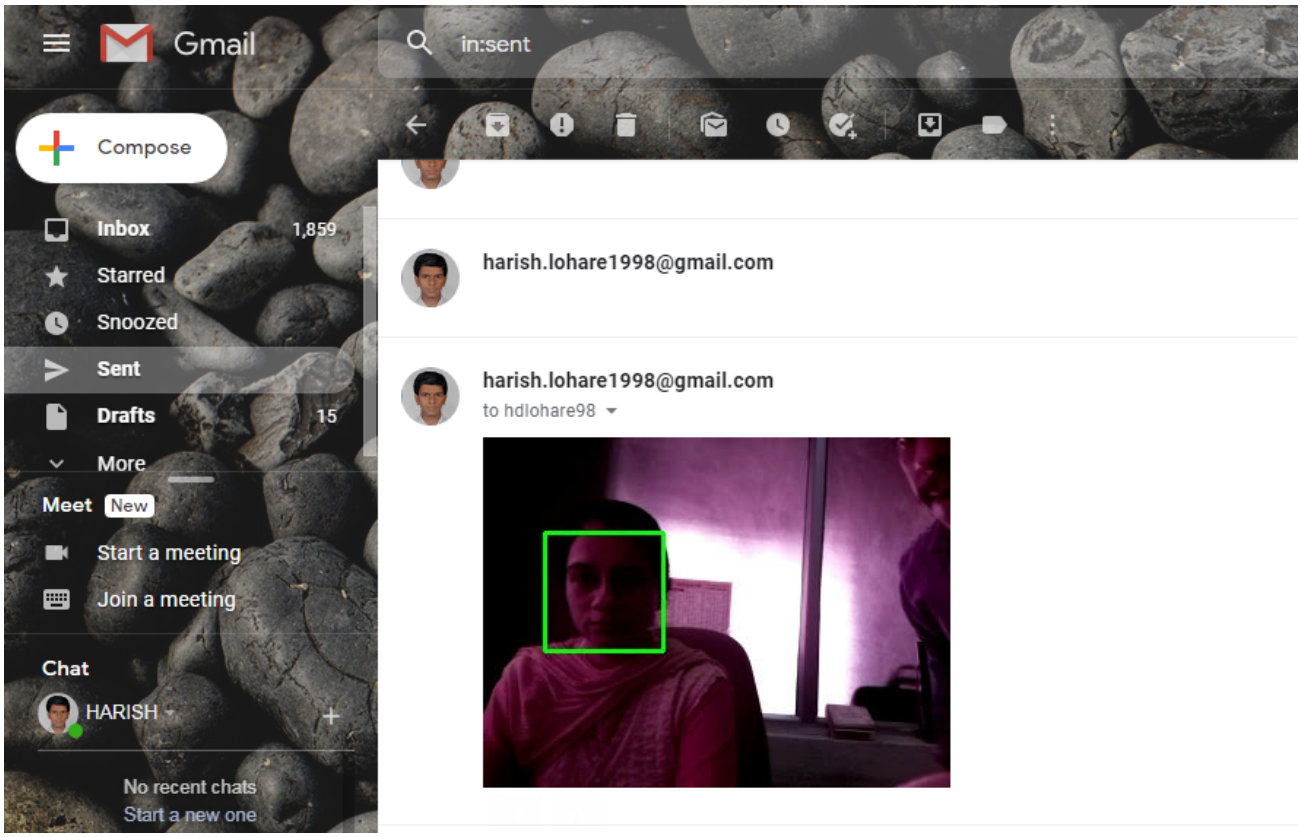
Running Main.py to Turn on the System.



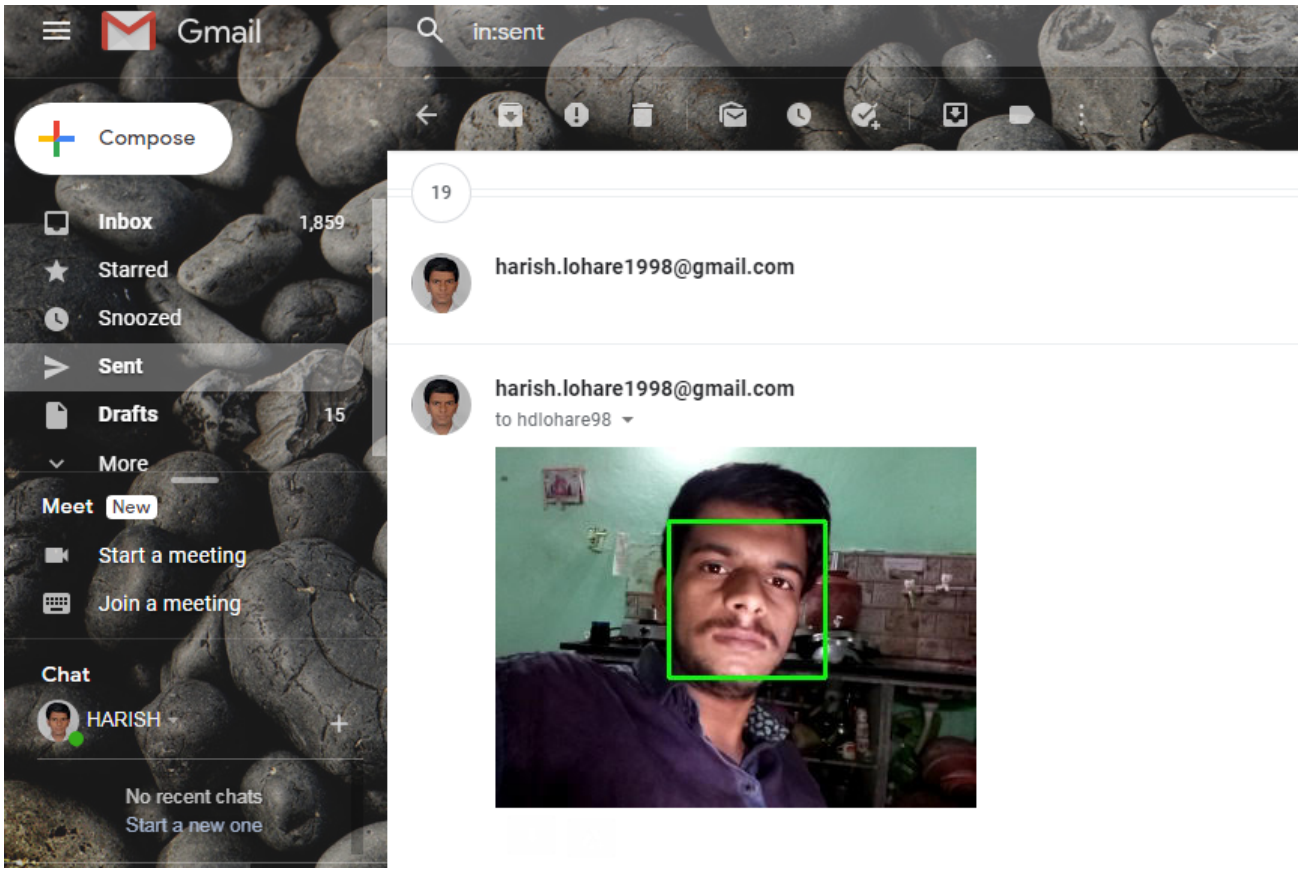
Family Member Arrives



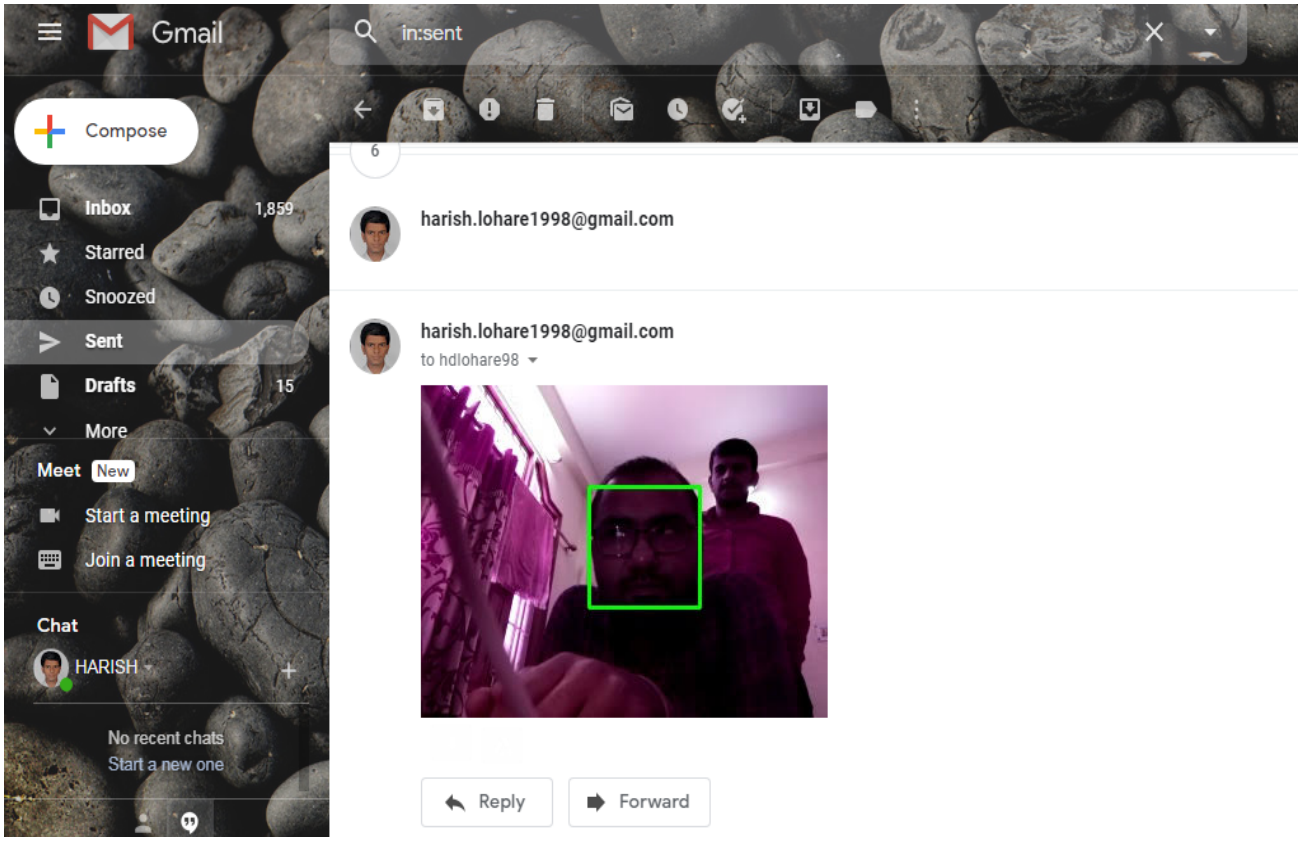
Sending email after face detection



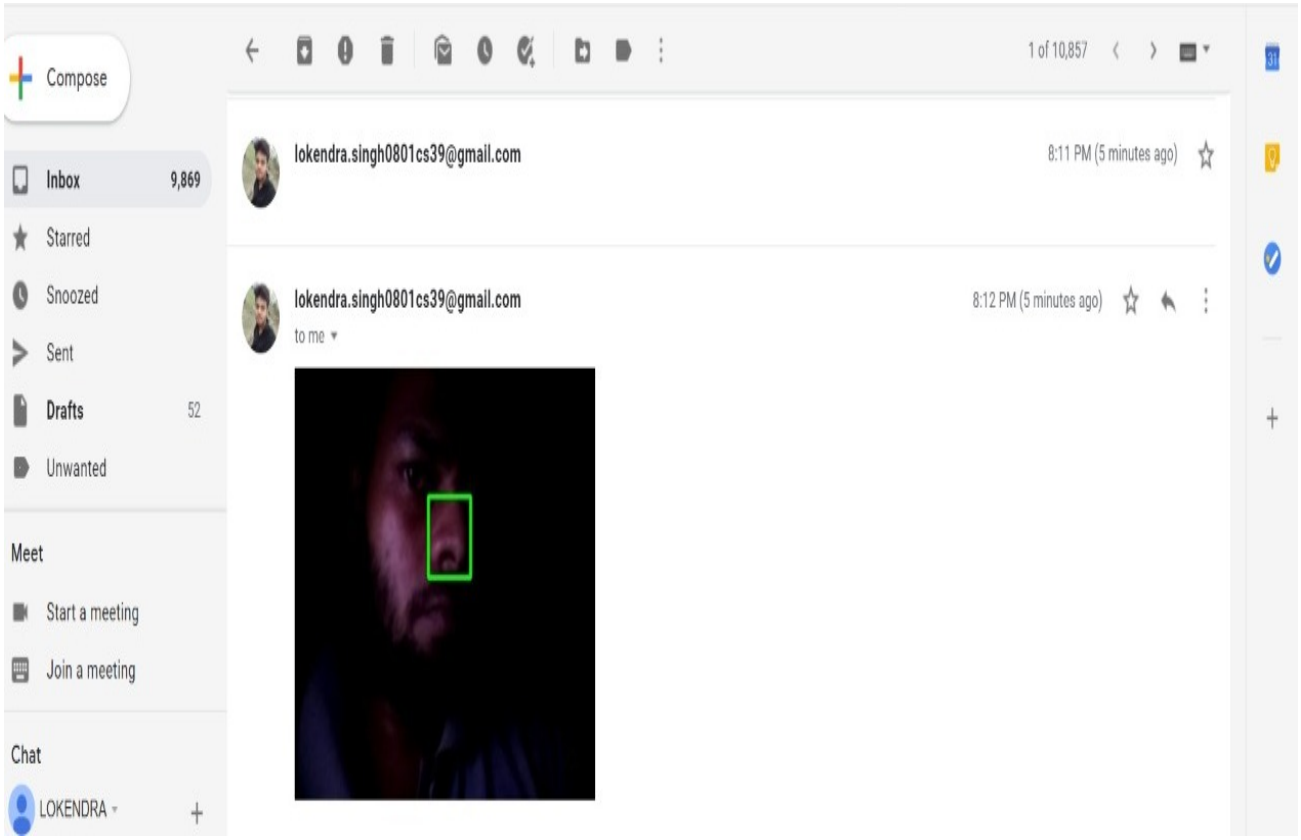
Sample Image 1



Sample Image 2



Sample Image 3



Sample Image 4