# Image Processing (Face Recognition) Based Door Lock

## A Major Project Report Submitted to


## Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal
## Towards Partial Fulfillment for the Award of

## Bachelor of Engineering

## (Information Technology)

**Under the Supervision of**

**Prof. Pawan K Gupta**

**Submitted By**

**Raksha Sankhala(0827IT161086)**

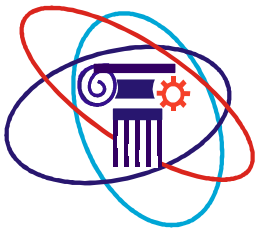
**Saiel Wadwekar(0827IT161094)**

**Saquib Qureshi(0827IT161098)**

**Yashasvi Sharma(0827IT161121)**

**Yashraj Sharma(0827IT161122)**

**Yuvraj Panwar(0827It161123)**



**Department of Information Technology**

**Acropolis Institute of Technology & Research, Indore**

**June 2020**

# EXAMINER APPROVAL

*The Project entitled "***Image Processing (Face Recognition) Based Door Lock***"* **submitted by Raksha Sankhala**(0827IT161086)**, Saiel Wadwekar** (0827IT161094)**, Saquib Qureshi** (0827IT161098), **Yashasvi Sharma** (0827IT161121), **Yashraj Sharma** (0827IT161122), **Yuvraj Singh Panwar** (0827IT161123) has been examined and is hereby approved towards partial fulfilment for the award of ***Bachelor of Engineering degree in Information Technology*** discipline, for which it has been submitted. It understood that by this approval the undersigned do not necessarily endorse or approve any statement made, opinion expressed or conclusion drawn therein, but approve the project only for the purpose for which it has been submitted.

**(Internal Examiner)** **(External Examiner)**

**Date:** **Date:**

# GUIDE RECOMMENDATION

This is to certify that the work embodied in this project entitled **"Image Processing (Face Recognition) Based Door Lock"** Submitted by **Raksha Sankhala**(0827IT161086), **Saquib Qureshi**(0827IT161098), **Saiel Wadwekar** (0827IT16094), **Yashasvi Sharma**(0827IT161121), **Yashraj Sharma** (0827IT161122), **Yuvraj Singh Panwar** (0827It161123) is a satisfactory account of the bonafide work, done under the supervision of **Prof. Pawan Gupta**, is recommended towards partial fulfillment for the award of the **Bachelor of Engineering (Information Technology)** degree by **Rajiv Gandhi Proudyogiki Vishwavidhyalaya, Bhopal.**

**(Project Guide)** **(Project Coordinator)**

# STUDENTS UNDERTAKING

This is to certify that project entitled "**Image Processing (Face Recognition) Based Door Lock**" has developed by us in the supervision of **Prof. Pawan K Gupta**. The whole responsibility of work done in this project is ours. The sole intension of this work is only for practical learning and research. However, we put proper citation remarks in our work. If the same work found then we are liable for explanation to this.

**Raksha Sankhala**(0827IT161086)

**Saiel Wadwekar** (0827IT161094)

**Saquib Qureshi** (0827IT161098)

**Yashasvi Sharma** (0827IT161121)

**Yashraj Sharma** (0827IT161122)

**Yuvraj Singh Panwar** (0827IT161123).

# ACKNOWLEDGEMENT

We thank the almighty Lord for giving us the strength and courage to sail out through the tough and reach on shore safely.

There are number of people without whom this projects work would not have been feasible. Their high academic standards and personal integrity provided me with continuous guidance and support.

We owe a debt of sincere gratitude, deep sense of reverence and respect to our guide and mentor **Prof. Pawan K Gupta,** for his motivation, sagacious guidance, constant encouragement, vigilant supervision and valuable critical appreciation throughout this project work, which helped us to successfully complete the project on time.

We express profound gratitude and heartfelt thanks to **Prof. Pawan K Gupta**, for his support, suggestion and inspiration for carrying out this project. I am very much thankful to other faculty and staff members of IT Dept, AITR Indore for providing me all support, help and advice during the project. We would be failing in our duty if do not acknowledge the support and guidance received from **Dr. S C Sharma**, Director, AITR, Indore whenever needed. We take opportunity to convey my regards to the management of Acropolis Institute, Indore for extending academic and administrative support and providing me all necessary facilities for project to achieve our objectives.

We are grateful to **our parent** and **family members** who have always loved and supported us unconditionally. To all of them, we want to say "Thank you", for being the best family that one could ever have and without whom none of this would have been possible.

**Raksha Sankhala**(0827IT161086)**, Saiel Wadwekar** (0827IT161094)**, Saquib Qureshi** (0827IT161098), **Yashasvi Sharma** (0827IT161121), **Yashraj Sharma** (0827IT161122), **Yuvraj Singh Panwar** (0827IT161123).

# SUMMARY

***Image Processing (Face Recognition) Based Door Lock***

This project is submitted to Rajiv Gandhi Proudyogiki Vishwavidhyalaya, Bhopal (MP), India for partial fulfilment of Bachelor of Engineering in Information Technology branch under the sagacious guidance and vigilant supervision of Prof. **Pawan Kumar Gupta**.

Authentication is one of the significant issues in the era of the information system. Among other things, human face recognition (HFR) is one of the known techniques which can be used for user authentication. As an important branch of biometric verification, HFR has been widely used in many applications, such as video monitoring/surveillance system, human-computer interaction. This project proposes a method for automatic door access system using face recognition technique by using python programming and from OpenCV library Haar cascade method. Object Detection using Haar feature-based cascade classifiers is an effective object detection method proposed by Paul Viola and Michael Jones. This is the standalone security device has been developed by electronic development board and operated on Battery power supply, wireless internet connectivity. Automatic notification has been achieved by sending security alert notification to the user app. This proposed is more effective, reliable, and this system consumes very less data and power compared to the other existing systems.

# List of Figures

# Table of Contents

# Chapter 1: INTRODUCTION

In today's world of connectivity and smart devices there is an urgent need to modify our existing day to day objects and make them smart, also it is not the era when we can blindly trust the old and conventional security measures, specifically speaking is our door locks. To change and modernize any object we need to eliminate its existing drawbacks and add extra functionality. Face detection is more challenging because of some unstable characteristics, for example, glasses and beard will impact the detecting effectiveness. Moreover, different kinds and angles of lighting will make detecting face generate uneven brightness on the face, which will have an influence on the detection process. An intensive study of OpenCV platform and its inbuilt libraries has been conducted to generate a code, which does correct and reliable facial recognition with new and efficient use of hardware.

This proposed system also acts as a home security system for both Person detection and provide security for door access control by using facial recognition for the home environment. The human body is identified as an intruder within a home environment achieved by capturing live video from web camera and processing will be done on captured.

Security deft has suggested various preferred approaches like biometric and password to enhance security. But the technology is developed and growing with the usage of different equipment's. The trend's moved from fingerprint to face recognition. So, we prefer a face recognition system for unlocking the door. Facial recognition is widely used in various industries and corporate sectors.
This door unlocking system mainly uses facial recognition.

The latest camera is used to detect the images and the images are sending to the database. If the image matches with the admin's image then the door is unlocked and an acknowledgement is sent via Zigbee as "y" if it does not match the image an acknowledgement is sent as "n". A pass code column is shown which takes values from the keypad to unlock the door.

## 1.1 Project Aim

We propose Smart Door Lock System to overcome the present obstacles in door locks. The system replaces traditional door locks which uses mechanical key. These traditional ways can be threatening if the keys or card is misplaced. Hence to avoid such incidents, we use face recognition technology to give access of the house to the person who is trying to enter the house. Everyone has a unique face and the chances that some other twin can open the door is very less.

Also, we implement intrusion detection in our door locks. If someone tries to enter the house in unauthorized way, owner of the house will be notified and he can make necessary actions against it.

We choose face as our key because it is more reliable and hassle-free way to open the door. The system is user friendly and more secure in terms of intrusion.

The most important of feature of any home security system is to detect the people who enter or leave the house. Instead of monitoring that through passwords or pins unique faces can be made use of as they are one's biometric trait. These are innate and cannot be modified or stolen easily. The level of security can be raised by using face detection. The proposed face recognition door lock security system has been developed to prevent robbery in highly secure areas like home environment with lesser power consumption and more reliable standalone security device for both Intruder detection and for door security.

**Objective**

The objective of this project is to use traditional Computer Vision techniques to develop an advanced and robust algorithm that can detect and track lane boundaries in a video. The pipeline highlighted below was designed to operate under the following scenarios:

- To detect the face of an individual so that system could verify the identity.
- To store a database of faces that can be identified by the system with the help of a camera.
- To verify whether the face present on the camera actually matches the face that is already processed into the system.
- To indicate a sign of negation or affirmation when the face readings are taken as an input by the computer.
- To set up a camera in an angle such that the individual entering the premises can be clearly observed.
- To implement a more reliable way in door lock system.
- To eliminate intrusion threats by making the user aware about them.
- To provide hassle-free and user-friendly way to access the door.

## 1.2 Project Description

### 1.2.1 Goal

The goal of this project is to build up a door lock (take a frame from camera as an input, do something, return a modified version of the frame), which allows detecting faces in simple conditions: sunny weather, good visibility, only people. One more thing*:* the face detected should match the database.

### 1.2.2 Dependencies

- Python 3.7
- NumPy
- Matplotlib (for charting and visualising images)
- OpenCV 4.1
- MoviePy (to process video files)

### 1.2.3 Project structure

- dataset/

  : Contains our face images organized into subfolders by name.

- images/

  : Contains three test images that we'll use to verify the operation of our model.

- face_detection_model/

  : Contains a pre-trained Caffe deep learning model provided by OpenCV to *detect* faces. This model detects and localizes faces in an image.

- output/

  : Contains my output pickle files. If you're working with your own dataset, you can store your output files here as well. The output files include:

- embeddings.pickle

  : A serialized facial embeddings file. Embeddings have been computed for every face in the dataset and are stored in this file.

- le.pickle

  : Our label encoder. Contains the name labels for the people that our model can recognize.

- recognizer.pickle

  : Our Linear Support Vector Machine (SVM) model. This is a machine learning model rather than a deep learning model and it is responsible for actually recognizing faces.

### 1.2.4 Pipeline

This project uses LBPH (Local Binary Patterns Histograms) Algorithm to detect faces. It labels the pixels of an image by thresholding the neighborhood of each pixel and considers the result as a binary number.

LBPH uses 4 parameters :

(i) Radius: the radius is used to build the circular local binary pattern and represents the radius around the central pixel.

(ii) Neighbors : the number of sample points to build the circular local binary pattern.

(iii) Grid X : the number of cells in the horizontal direction.

(iv) Grid Y : the number of cells in the vertical direction.

The model built is trained with the faces with tag given to them, and later on, the machine is given a test data and machine decides the correct label for it.

**Software and Techniques for Face Recognition:**

This System is used Python 3.5+ software and python programming. With OpenCV (*Open Source Computer Vision*) is a library of programming functions mainly aimed at real- time computer vision.

**Haar Cascade Classifier:**

Object Detection using Haar feature-based cascade classifiers is an effective object detection method proposed by Paul Viola and Michael Jones in their paper, "Rapid Object Detection using a Boosted Cascade of Simple Features" in 2001. It is a machine learning based approach where a cascade function is trained from a lot of positive and negative images. It is then used to detect objects in other images.

**Local Binary Patterns Histograms Technique:**

The Local Binary Patterns methodology has its roots in 2D texture analysis. The basic idea of Local Binary Patterns is to summarize the local structure in an image by comparing each pixel with its neighborhood. Take a pixel as center and threshold its neighbors against. If the intensity of the center pixel is greater-equal its neighbor, then denote it with 1 and 0 if not. You'll end up with a binary number for each pixel, just like 11001111. So with 8 surrounding pixels, you'll end up with 2^8 possible combinations, called *Local Binary Patterns* or sometimes referred to as *LBP codes*. The first LBP operator described in literature actually used a fixed 3 x 3 neighborhood just like this.

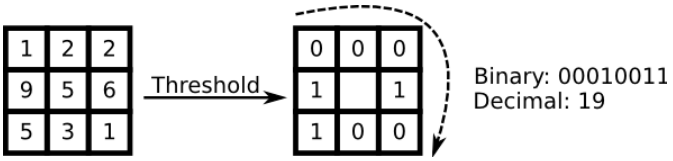
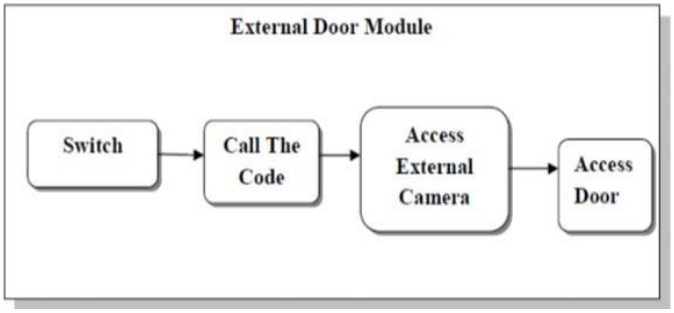



Fig 1.1: Local Binary Histogram Technique

**External Door module:**

The following block diagram shows the external module of door unlocking system. Which is have the switch, Led by indication & camera. In this module when the person presses the switch, then the code will run from the virtual environment of the raspberry pi board. Then the camera is started, and from camera capture, the images and it will process in raspberry pi. After matching and verifying the images the Led will glow it means the door is open.

Fig 1.2: External Door Module



First we write the code for detecting and recognition of image through the camera.

Then start the training code, and the camera will open up, Accuracy depends on the number of data sets as well as the quality and lighting conditions.

Take pictures of the person for face recognition after running create_database.py script. It automatically creates Train folder in Database folder containing the face to be recognized. You can change the name from Train to the person's name. The result in the creation of the real-time database is recorded. The real-time database is created by using python. While executing it 100 images of each subject. Likewise, databases should be created a dictionary and it creates each image size of variable pixels of height and width. While creating the database, the face images must have different expressions, which is why a 0.38-second delay is given in the code for creating the dataset. In this example, we take about 45 pictures/images and extract the face, convert it into greyscale and save it to the database folder with its name.

After program execution, the database is created and then run face recognition program. And the database is accessed by the program and it will match the live stream video image with it.

The proposed system works are as follows:

- Interface the camera to capture live face images.
- Creating a database of an authorized person by using Graphical user interface shown in below image.

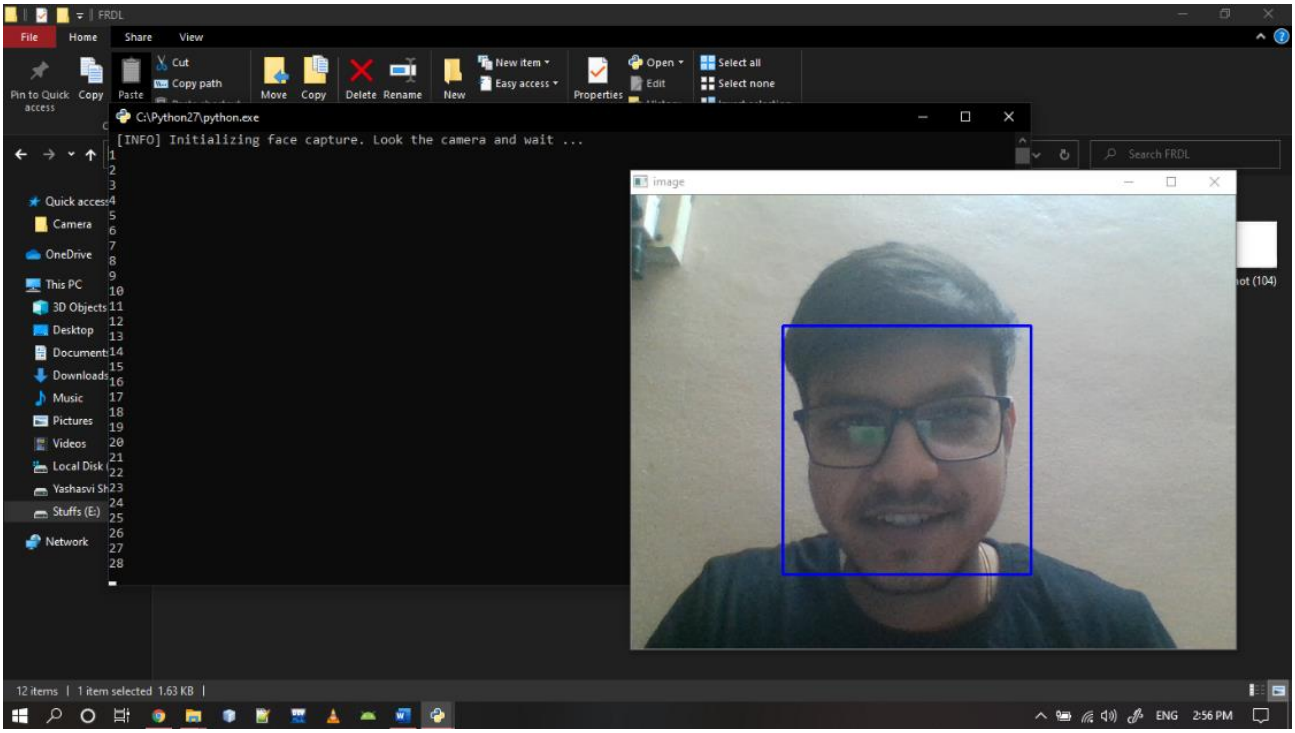



Fig 1.3: Frame of Face Detected

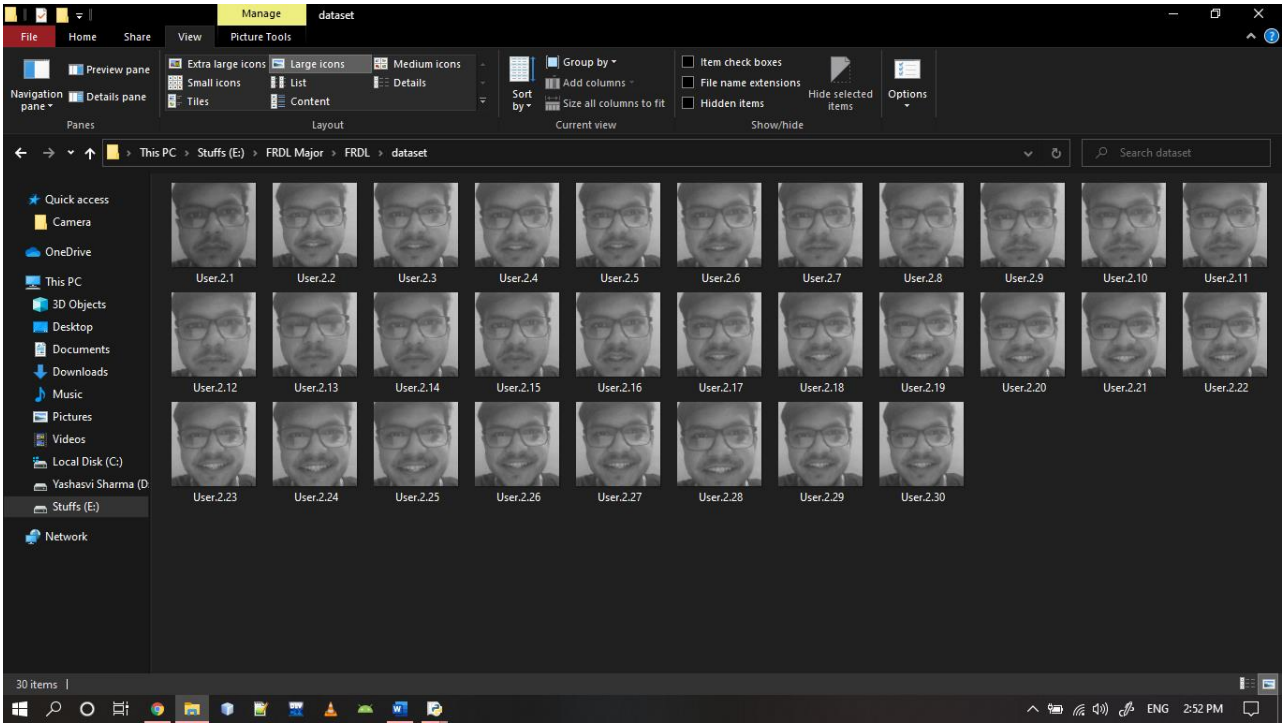



Fig 1.4: Face Dataset Created

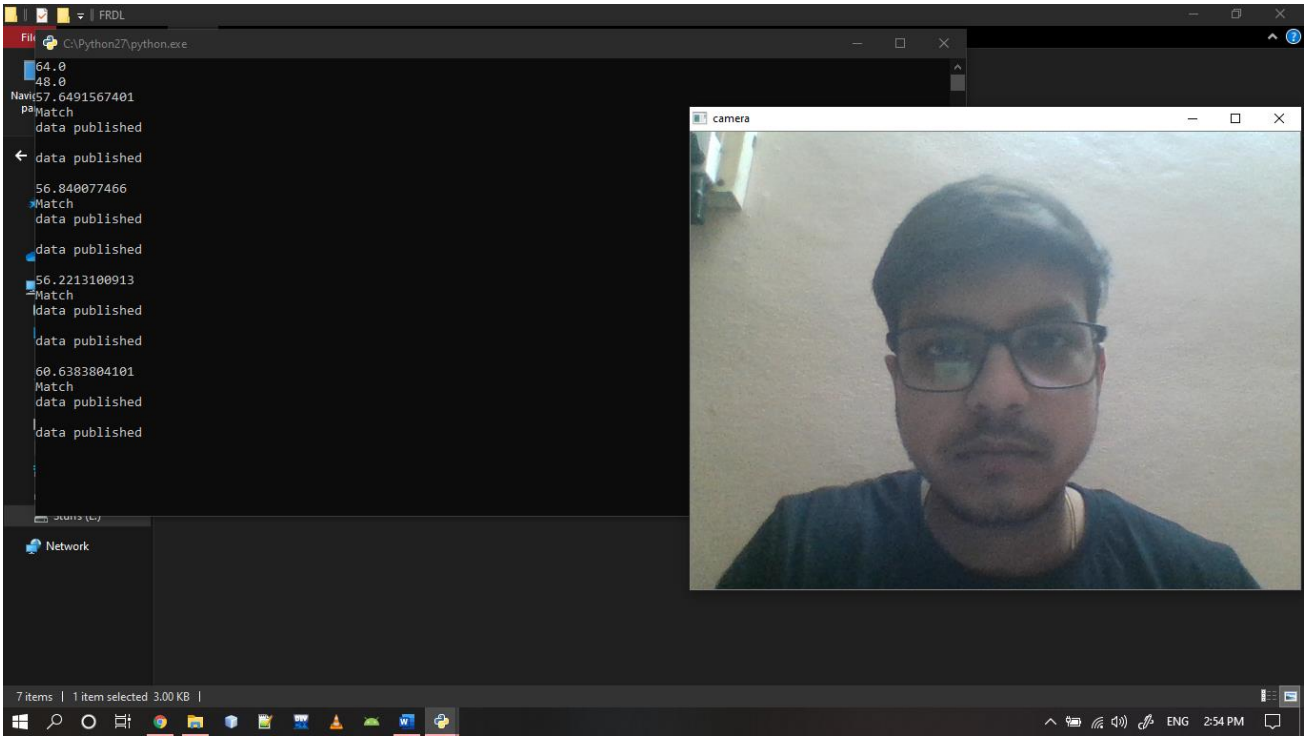



Fig 1.5: Face Detected

When the face is recognized then an alert to an authorized person is sent through the app while unlocking the locked door.

## 1.3 Scope

Authentication is one of the significant issues in the era of the information system. Among other things, human face recognition (HFR) is one of the known techniques which can be used for user authentication. As an important branch of biometric verification, HFR has been widely used in many applications, such as video monitoring/surveillance system, human-computer interaction. This project proposes a method for automatic door access system using face recognition technique by using python programming and from OpenCV library Haar cascade method. Object Detection using Haar feature-based cascade classifiers is an effective object detection method proposed by Paul Viola and Michael Jones.

Automatic alert notification has been achieved by sending security alert to the user app. This proposed is more effective, reliable, and this system consumes very less data and power compared to the other existing systems.

In today's world by using smart devices we are make our needs smart. By following trends and updates we have to consider and remove drawbacks in existing system and add more features and updates. Face detection system is more complex because of unstable characteristics. Example: let us consider glasses and beard will show some impact to detect the faces. So by considering the different angles and multiple images of faces and it will influence on detection process. The study of OpenCV and its inbuilt library functions helps to generate a code will do correct and authentic facial recognition system with new and more efficient use of hardware. Human body will identified as an input within environment by capturing live video from a web camera and the process will be done on captured video frames.

To run this model there are different algorithms in that we took because it will provide more accuracy results when compared to other algorithms. By this, we can say that door locking and unlocking by detecting faces is a new model which includes an alternate manual pass code unlocking system by using keypad which helps to gain access to that door in necessary situations.

## 1.4 Background of Project

The goal of this project is to build up a simple image pipeline (take a frame from camera as an input, do something, return a modified version of the frame), which allows detecting faces in simple conditions: sunny weather, good visibility.

## 1.5 Operation Environment

The Facial Recognition using door lock is a project and shall operate in all systems, for a particular system we are taking a Windows 10 laptop with Python Version 3.7.4, OpenCV and its packages ssuch as Matplotlib, Pillow, Numpy, PIP. For text editor we are using Jetbrains by Pycharm community edition 2019.2.1.

## 1.6 Report Structure

The project Real-time Object detection and Recognition is primarily concerned with the Image processing in real-time and whole project report is categorized into five chapters.

Chapter 1: Introduction- introduces the background of the problem followed by rationale for the project undertaken. The chapter describes the objectives, scope and applications of the project. Further, the chapter gives the details of team members and their contribution in development of project which is then subsequently ended with report outline.

Chapter 2: Review of Literature- explores the work done in the area of Project undertaken and discusses the limitations of existing system and highlights the issues and challenges of project area. The chapter finally ends up with the requirement identification for present project work based on findings drawn from reviewed literature and end user interactions.

Chapter 3: Proposed System - starts with the project proposal based on requirement identified, followed by benefits of the project. The chapter also illustrate software engineering paradigm used along with different design representation. The chapter also includes block diagram and details of major modules of the project. Chapter also gives insights of different type of feasibility study carried out for the project undertaken. Later it gives details of the different deployment requirements for the developed project.

Chapter 4: Implementation - includes the details of different Technology/ Techniques/ Tools/ Programming Languages used in developing the Project. The chapter also includes the different user interface designed in project along with their functionality. Further it discusses the experiment results along with testing of the project. The chapter ends with evaluation of project on different parameters like accuracy and efficiency.

Chapter 5: Conclusion - Concludes with objective wise analysis of results and limitation of present work which is then followed by suggestions and recommendations for further improvement.

# Chapter 2: Review of Literature

## 2.1 Related Work

In previous works, they deal with different algorithms technologies and equipment for unlocking the door. In reference article 8 studies by Somjit Nath, Paramita Banerjee proposed "Arduino Based Door Unlocking System with Real Time Control".This approach implemented with RFID codes to scan for unlocking the door. So when a person wants to enter the door he needs to scan the card then he gains access to that door. If he misplaces that card he cannot access through that door and there is a chance of insecurity that anyone can access to the door with that misplaced card. This stands as a drawback to this system.

In reference article 7 studies by Charoen Vongchumyen, Pakorn Watanachaturaporn, pattaya proposed "Door locking system via web application". In this approach, a web application is designed to monitor the door so that user can easily access door by his mobile and he can also check the status whether it is locked or not. There is a drawback in this system: when someone hacked and got security code then hackers can easily access to that room.

In reference article 2 study by Suchit Shavi proposed "Secured Room Access Module".In this approach a keyboard based door unlocking system is implemented with a micro controller. So the user needs to enter his password to unlock the door. In this approach is secured when compared to the previous models and well used in today's world. Even though this system is secured there might be a drawback: If someone observes your password, they can gain access to that door by using the same password you have entered.

In reference article 1 study by Muhammad Kashif Shaikh, Syedannas Bin Mazhar proposed "Comparative Analysis for a Real Time Face Recognition System Using Raspberry Pi". This approach did an analysis of various algorithms on the face recognition system. This analysis took LBPH (Local Binary Pattern Histogram), Fisher Faces, Eigen Faces Algorithms for comparison and checked with different processors to know time complexity and accuracy of various algorithms. This approach is most secured and without matching face no one can access that door.

Hence this survey work proves that face recognition system is the best to approach for using in present days by referencing this approach we designed a new type of door locking and unlocking system.

## 2.2 Requirement Analysis

Requirements analysis in systems engineering and software engineering, encompasses those tasks that go into determining the needs or conditions to meet for a new or altered product, taking account of the possibly conflicting requirements of the various stakeholders, such as beneficiaries or users. It is an early stage in the more general activity of requirements engineering which encompasses all activities concerned with eliciting, analyzing, documenting, validating and managing software or system requirements.

Requirements analysis is critical to the success of a systems or software project. The Requirements should be documented, actionable, measurable, testable, traceable, related to identified business needs or opportunities, and defined to a level of detail sufficient for system design.

## 2.3 Problem Definition

In the world of emerging technology, security became an essential component in day to day life. Information theft, lack of security and violation of privacy etc. are the essential components which are needed to be protected. Using smart secure systems for door lock and unlocking became popular nowadays. This is system is being adapted by many countries and first grade countries such as USA, Japan etc. already makes use of this system. This system provides either a facial recognition security feature or a keypad is provided to enter the pass code to unlock the door. Although it provides security to the doors, it also has its own drawbacks:

Firstly, if the system mainly uses a facial recognition module, there might be a slight chance that sometimes the face may not be detected and hence the door cannot be unlocked.

Secondly, if the system uses a keypad to enter the pass code to unlock the door, there might be a chance that the key maybe is recorded or can be observed by others without users consent. Hence, two-step verification is developed which makes use of facial recognition as the first step and pass code as its following step. But the same issues pertain in the newly developed system.

Thus, a new model which rectifies all the above issues is developed.

## .2.4 Requirement Specification

The most important of feature of any home security system is to detect the people who enter or leave the house. Instead of monitoring that through passwords or pins unique faces can be made use of as they are one's biometric trait. These are innate and cannot be modified or stolen easily. The level of security can be raised by using face detection. The proposed face recognition door lock security system has been developed to prevent robbery in highly secure areas like home environment with lesser power consumption and more reliable standalone security device for both Intruder detection and for door security.

Whenever the person comes in front of the door, it recognizes the face and if it is registered then it unlocks the door, if the face is not registered it will raise an alarm and clicks a picture and send it on the registered number. This is how the system works.

Functional Requirements

- The lines are mapped from the face which are a result of determination of pixels which would be a part of the right and left lines of the structure of eyes.
- A way which would help to reduce security problems as lesser chances of robbery would take place.
- A method which would combine for the concept of facial recognition through a hardware.

## 2.5 Limitation

- It provides less security as there might be issue in recognizing similar faces.
- Every time password entering required for unlocking Micro controller cannot interface high power devices directly.

# Chapter 3: Proposed System

## 3.1 Proposal

The proposed works are as follows:

- Interfacing of camera to capture live face images.
- Create a database of authorized person if they exist.
- Capturing current image, save it and compare with the database image.
- Interface application module to send alert to authorized person while unlocking the locked door in the form of notification.
- The project can also be used for surveillance. For instance, it can capture the images of unidentified individuals and store it which can later be used to determine the impostors who tried to gain illegitimate access.
- Interface relay as an output.

The system will work in two different parts. The first part is for capturing and creating a database by storing the image. And the second one is to compare the image with the stored images in the database. For feature extraction we will use Eigen faces methodology and Euclidian distances will used for recognition of the face.

Camera module: Camera module is pi camera interfacing to the raspberry pi module. It is used to capture images and send the clicked images to the raspberry pi module. Camera contains LEDs and flashes to handle that light condition that is not explicitly supplied by the environment and these light conditions are known as ambient light conditions.

There are two parts in this section. The first is the implementation of Door lock access by using Face Recognition and the second is the implementation of person detection along with auto alert sending.

### Implementation of Door Lock Access by using Face Recognition

This project work proposes an idea of for face reorganization concept for accessing the door lock system and it implemented with the help of OpenCV, which is a popular computer vision library. Face recognition is an important application of image processing owing to its use in many fields. An effective face recognition system based on OpenCV is developed in the project. Face recognition has been the best choice for the problem of biometrics and it has a various type of applications in our present life. An efficient face recognition system can be of great help in forensic sciences, identification for law enforcement, authentication for banking and security system, and giving preferential access to authorized users i.e. access control for secured areas etc.

A real-time door lock access system by face recognition system. The algorithm used here is Local Binary Patterns Histograms (LBPH), based on Haar Feature-based Cascade Classifiers is presented in the project. The technique used here we will work with face detection. Initially, the algorithm needs a lot of positive images (images of faces) and negative images (images without faces) to train the classifier. Then we need to extract features from it. For this, Haar features shown in the below Fig-3.1 are used. They are just like our convolution kernel. Each feature is a single value obtained by subtracting the sum of pixels under the white rectangle from the sum of pixels under the black rectangle.

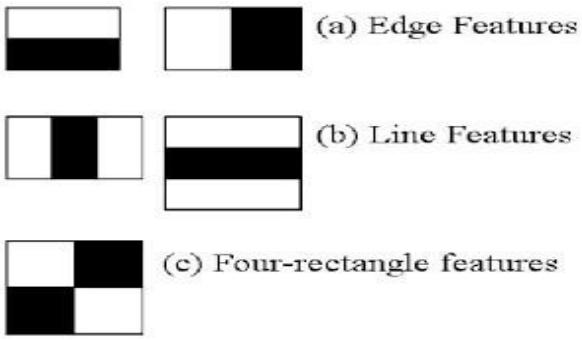



Fig 3.1: Haar Features

**Implementation of Person Detection along with Auto Alert Sending.**

The proposed home security system is a machine learning based approach where a cascade function is trained from a lot of positive and negative images. It is then used to detect objects in other images. A procedural view of how this person detection works is shown in below flowchart.

In this, we are using the LBPH (Local Binary Pattern Histogram) Algorithm. This algorithm will give us more accurate results when we compare to other types of algorithms such as Fisher Face, Eigen Faces Algorithms in base paper. This LBPH Algorithm will take number of images as you wish in different angles and check those all images at the time of face no recognition. In our case, we are taking 20 images of a person with different angles and it will be stored in our date base. For this algorithm, we are using VNC viewer to run raspbian os for detecting images from the data base. At first, we have to save images by using data sets and after that, we will train that faces to algorithm then it stores into the data base. At first, it converts color images to gray scale images and then it converts into pixels for detecting this will divides the image into various pieces then it stores the values of each pixel. If pixels are less then it will be represented as 0 and pixels which are high will be 1 then it will be arranged in 3 x 3 matrix format for recognizing the new images on screen compared to data base stored images. Here are some different variations of faces that is capture.


Fig 3.2: Data set training images

### 3.2 Feasibility

Feasibility study is the process of determination of whether or not a project is worth doing. Feasibility studies are undertaken within tight time constraints and normally culminate in a written and oral feasibility report. The contents and recommendations of this feasibility study helped us as a sound basis for deciding how to precede the project. It helped in taking decisions such as which software to use, hardware combinations, etc.

**Operational Feasibility:**

Operation feasibility is a measure of how people feel about the system. Operational Feasibility criteria measure the urgency of the problem or the acceptability of a solution. Operational Feasibility is dependent upon determining human resources for the project. It refers to projecting whether the system will operate and be used once it is installed.
The essential questions that help in testing the operational feasibility of a system are following.
Does management support the project?

Are the users not happy with current business practices? Will it reduce the time (operation) considerably? If yes, then they will welcome the change and the new system.

Have the users been involved in the planning and development of the project? Early involvement reduces the probability of resistance towards the new system.

Our proposed project "Face Recognition using Door Lock" is operationally feasible since there is no need for special training of staff member and whatever little instructing on this system is required can be done so quite easily and quickly as it is essentially This project is being developed keeping in mind the general people who one have very little knowledge of computer operation, but can easily access their required database and other related information. The redundancies can be decreased to a large extent as the system will be fully automated.

**Technical Feasibility:**

Technical feasibility determines whether the work for the project can be done with the existing equipment, software technology and available personnel. Technical feasibility is concerned with specifying equipment and software that will satisfy the user requirement.

- In technical feasibility the following issues are taken into consideration.
- Whether the required technology is available or not
- Whether the required resources are available
- Manpower- programmers, testers &debuggers
- Software and hardware

**Economical Feasibility:**

Economical feasibility has great importance as it can outweigh other feasibilities because costs affect organization decisions. The concept of Economic Feasibility deals with the fact that a system that can be developed and will be used on installation must be profitable for the Organization. The cost to conduct a full system investigation, the cost of hardware and software, the benefits in the form of reduced expenditure are all discussed during the economic feasibility.

During the economical feasibility test we maintained the balance between the Operational and Economical feasibilities, as the two were the conflicting. For example the solution that provides the

best operational impact for the end-users may also be the most expensive and, therefore, the least economically feasible.

We classified the costs of our proposed project "Face Recognition using Door Lock" according to the phase in which they occur. As we know that the system development costs are usually one-time costs that will not recur after the project has been completed. For calculating the Development costs we evaluated certain cost categories.

- Personnel costs
- Computer usage costs
- Training costs
- Supply and equipments costs
- Cost of any new computer equipments and software.

### 3.5 Diagrams

### Block Diagram

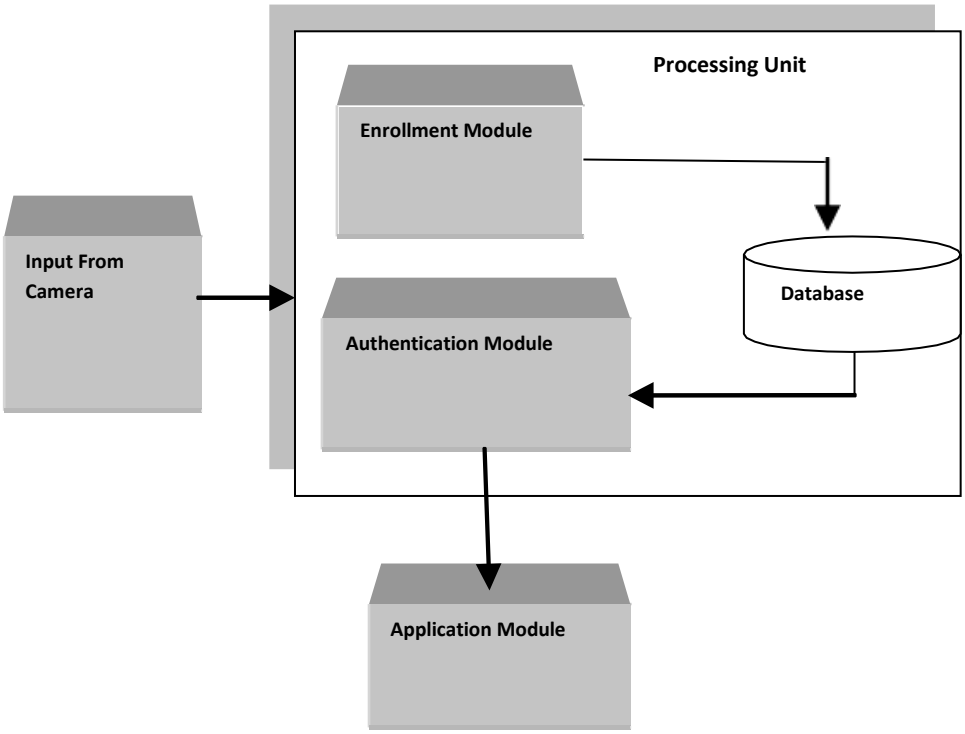



Fig 3.3: Block Diagram

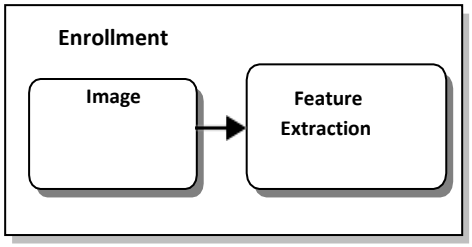
In input unit, the Facial images for Face Recognition and Video frames for person detection are captured from camera input devices respectively i.e. from Raspberry Pi Camera or Web camera.

The data which is collected from Input unit that is captured Image and Video frames input is fed into the processing unit in which the processing or calculations are performed on the proposed person detection and door lock system module, Here the processing unit is nothing but a Raspberry Pi board, along with code scripts of the implemented modules.

**Enrollment Module**

In the enrollment module, the data which collected from input camera means person face image is stored in the database. Before storing the image it will use feature extraction means it is converted in Haar Feature-based Cascade Classifiers.

Fig 3.3.1: Enrollment Module




**Authentication Module**

In this module, we recognize and detect the input images. This module is connected to the outer side of the door, where the captured image converted into Haar Feature- based Cascade Classifiers. And matching this feature extraction image with the database.

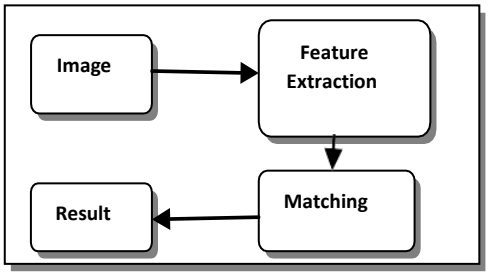



Fig 3.3.2: Authentication Module

**Application Module**

The Application specific unit which consists of Door lock circuitry, it is associated with Door lock system module of authentication module and it starts functioning according to results of the module to perform door lock open/close operation based on Face Recognition.

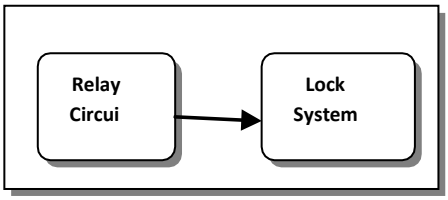



Fig 3.3.3: Application Module

**Flow Diagram**

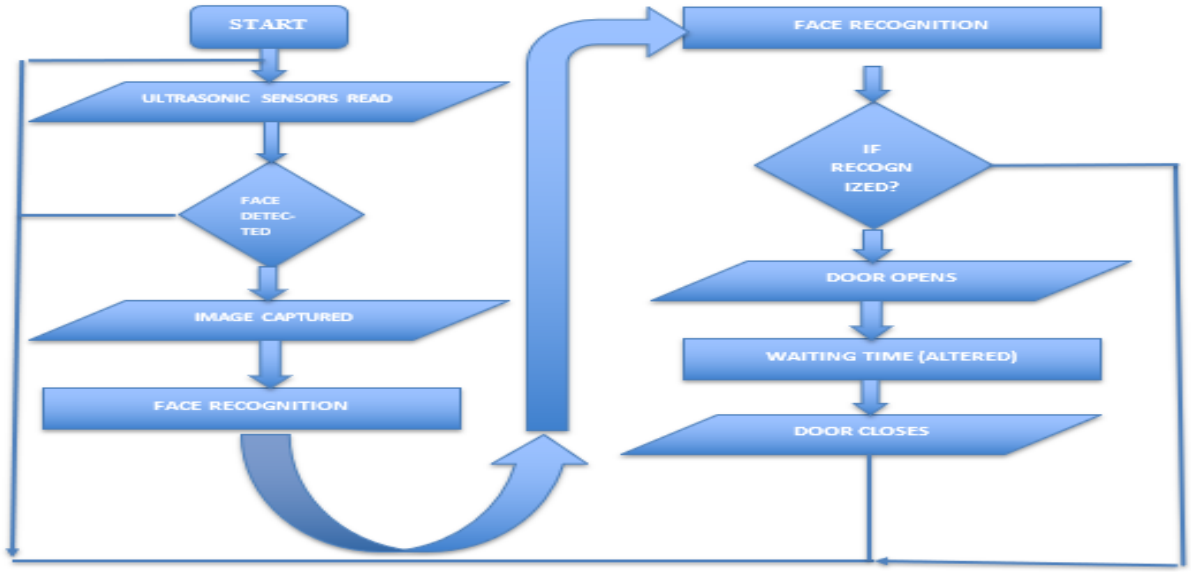



Fig 3.4 Flow Diagram of Person Detection

### 3.4 Deployment Requirement

Functional Requirements

- The lines are mapped from the face which are a result of determination of pixels which would be a part of the right and left lines of the structure of eyes.
- A way which would help to reduce security problems as lesser chances of robbery would take place.
- A method which would combine for the concept of facial recognition through a hardware.

Non-Functional Requirements

- Reliability:

  The capability to maintain the specified level of performance is called reliability. This application is a web based application that runs on any device that has a browser.

- Availability:
  The project will be available for 99% of the time.

- Security:
  The business logic is hidden from the users and is much safer and thus avoids un authorised or illegal access or database corruption. Security of the user's information is also safe as there is a login facility.

- Maintainability:
  Maintenance is typically done after the software development has been completed. As the time evolves, so do the requirements and needs. It revolves around the understanding of the existing software and the effects of the change.

- Portability:
  Portability is the ability of the system or application that can run in various environments. As the web application is based on the java language, the application is portable.

### 3.5 Software and Hardware Requirements

Software Requirements

- Jetbrains by Pycharm community edition 2019.2.1
- Python 3.7.4
- OpenCV
- Important packages such as Matplotlib
- Pillow
- Pip
- Numpy
- Opencv-python
- PyBundle
- Kiwisolver and many more library packages.

Hardware Requirements

- Intel Core i3 or above
- Windows 10
- RAM 4 GB or above
- Memory 500 GB or above
- Arduino Uno Board
- Capacitor
- Voltage Regulator
- Esp6266
- Servo Motor

# Chapter 4: Implementation

## 4.1 Software Tools

### Python

Python is a multi-paradigm programming language. Object-oriented programming and structured programming are fully supported, and many of its features support functional programming and aspect-oriented programming (including by metaprogramming and metaobjects (magic methods)). Many other paradigms are supported via extensions, including design by contract and logic programming.

Python uses dynamic typing, and a combination of reference counting and a cycle-detecting garbage collector for memory management. It also features dynamic name resolution (late binding), which binds method and variable names during program execution.

Python's design offers some support for functional programming in the Lisp tradition. It has `filter`, `map`, and `reduce` functions; list comprehensions, dictionaries, sets and generator expressions. The standard library has two modules (itertools and functools) that implement functional tools borrowed from Haskell and Standard ML.

The language's core philosophy is summarized in the document The Zen of Python (PEP 20), which includes aphorisms such as:

- Beautiful is better than ugly
- Explicit is better than implicit
- Simple is better than complex
- Complex is better than complicated
- Readability counts

Rather than having all of its functionality built into its core, Python was designed to be highly extensible. This compact modularity has made it particularly popular as a means of adding programmable interfaces to existing applications. Van Rossum's vision of a small core language with a large standard library and easily extensible interpreter stemmed from his frustrations with ABC, which espoused the opposite approach.

Python strives for a simpler, less-cluttered syntax and grammar while giving developers a choice in their coding methodology. In contrast to Perl's "there is more than one way to do it" motto, Python embraces a "there should be one—and preferably only one—obvious way to do it" design philosophy. Alex Martelli, a Fellow at the Python Software Foundation and Python book author, writes that "To describe something as 'clever' is not considered a compliment in the Python culture."

Python's developers strive to avoid premature optimization, and reject patches to non-critical parts of the CPython reference implementation that would offer marginal increases in speed at the cost of clarity. When speed is important, a Python programmer can move time-critical functions to extension modules written in languages such as C, or use PyPy, a just-in-time compiler. Cython is also available, which translates a Python script into C and makes direct C-level API calls into the Python interpreter.

An important goal of Python's developers is keeping it fun to use. This is reflected in the language's name—a tribute to the British comedy group Monty Python—and in occasionally playful approaches to tutorials and reference materials, such as examples that refer to spam and eggs (from a famous Monty Python sketch) instead of the standard foo and bar.

A common neologism in the Python community is pythonic, which can have a wide range of meanings related to program style. To say that code is pythonic is to say that it uses Python idioms well, that it is natural or shows fluency in the language, that it conforms with Python's minimalist philosophy and emphasis on readability. In contrast, code that is difficult to understand or reads like a rough transcription from another programming language is called unpythonic.

Users and admirers of Python, especially those considered knowledgeable or experienced, are often referred to as Pythonists, Pythonistas, and Pythoneers.

**OpenCV**

OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. Being a BSD-licensed product, OpenCV makes it easy for businesses to utilize and modify the code.

The library has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms. These algorithms can be used to detect and recognize faces, identify objects, classify human actions in videos, track camera movements, track moving objects, extract 3D models of objects, produce 3D point clouds from stereo cameras, stitch images together to produce a high resolution image of an entire scene, find similar images from an image database, remove red eyes from images taken using flash, follow eye movements, recognize scenery and establish markers to overlay it with augmented reality, etc. OpenCV has more than 47 thousand people of user community and estimated number of downloads exceeding 18 million. The library is used extensively in companies, research groups and by governmental bodies.

Along with well-established companies like Google, Yahoo, Microsoft, Intel, IBM, Sony, Honda, Toyota that employ the library, there are many startups such as Applied Minds, VideoSurf, and Zeitera, that make extensive use of OpenCV. OpenCV’s deployed uses span the range from stitching street view images together, detecting intrusions in surveillance video in Israel, monitoring mine equipment in China, helping robots navigate and pick up objects at Willow Garage, detection of swimming pool drowning accidents in Europe, running interactive art in Spain and New York, checking runways for debris in Turkey, inspecting labels on products in factories around the world on to rapid face detection in Japan.

It has C++, Python, Java and MATLAB interfaces and supports Windows, Linux, Android and Mac OS. OpenCV leans mostly towards real-time vision applications and takes advantage of MMX and SSE instructions when available. A full-featured CUDA and OpenCL interfaces are being actively developed right now. There are over 500 algorithms and about 10 times as many functions that compose or support those algorithms. OpenCV is written natively in C++ and has a templated interface that works seamlessly with STL containers.

**Matplotlib**

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and IPython shells, the Jupyter notebook, web application servers, and four graphical user interface toolkits.

Matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, errorcharts, scatterplots, etc., with just a few lines of code. For examples, see the sample plots and thumbnail gallery.

For simple plotting the pyplot module provides a MATLAB-like interface, particularly when combined with IPython. For the power user, you have full control of line styles, font properties, axes properties, etc, via an object oriented interface or via a set of functions familiar to MATLAB users.

**Numpy**

NumPy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays.

It is the fundamental package for scientific computing with Python. It contains various features including these important ones:

- A powerful N-dimensional array object
- Sophisticated (broadcasting) functions
- Tools for integrating C/C++ and Fortran code
- Useful linear algebra, Fourier transform, and random number capabilities

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

## 4.2 Screenshots

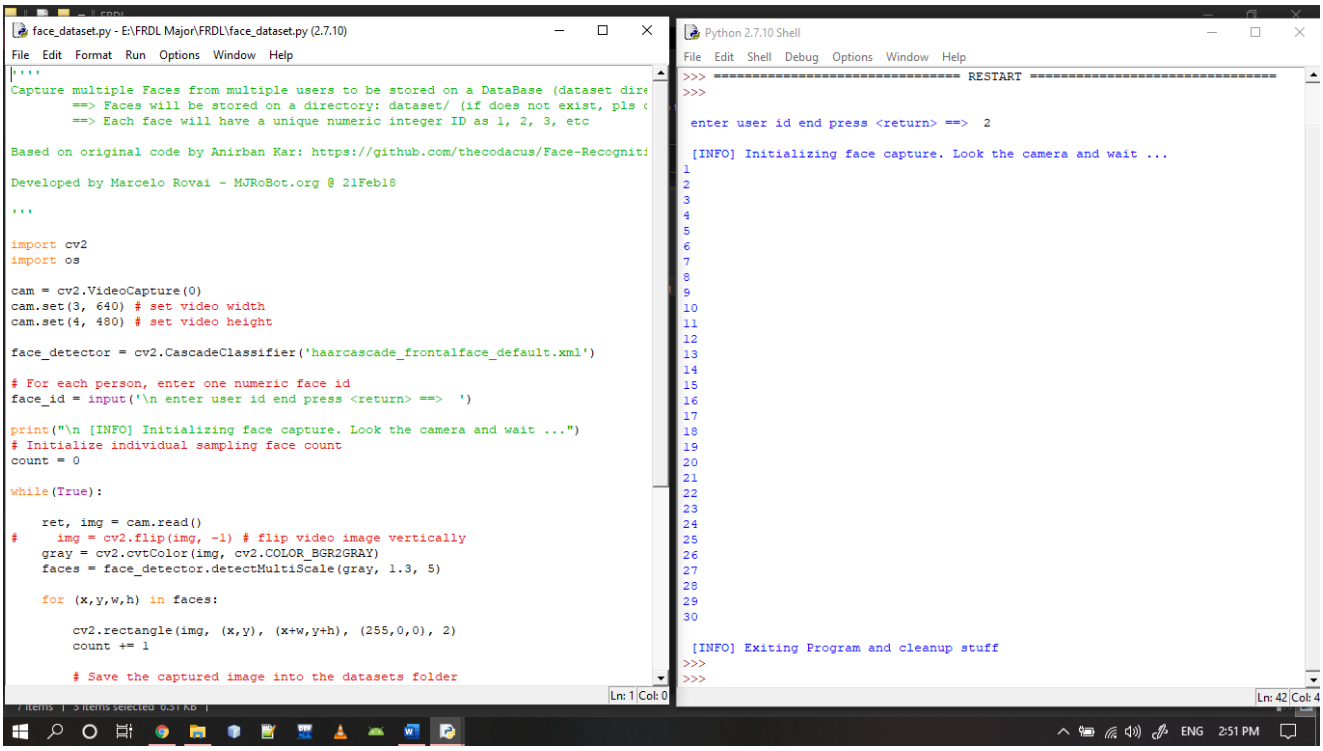



Fig. 4.1 Screenshot 1

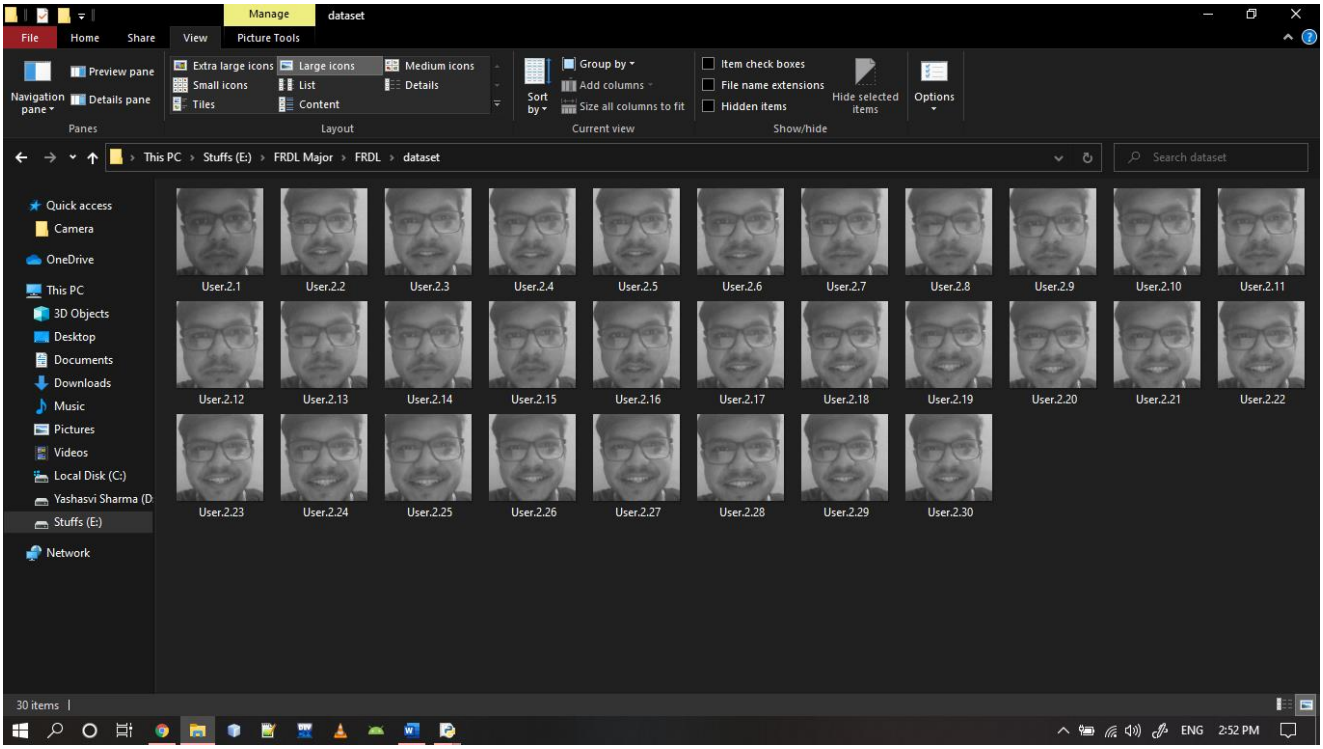



Fig. 4.2 Screenshot 2

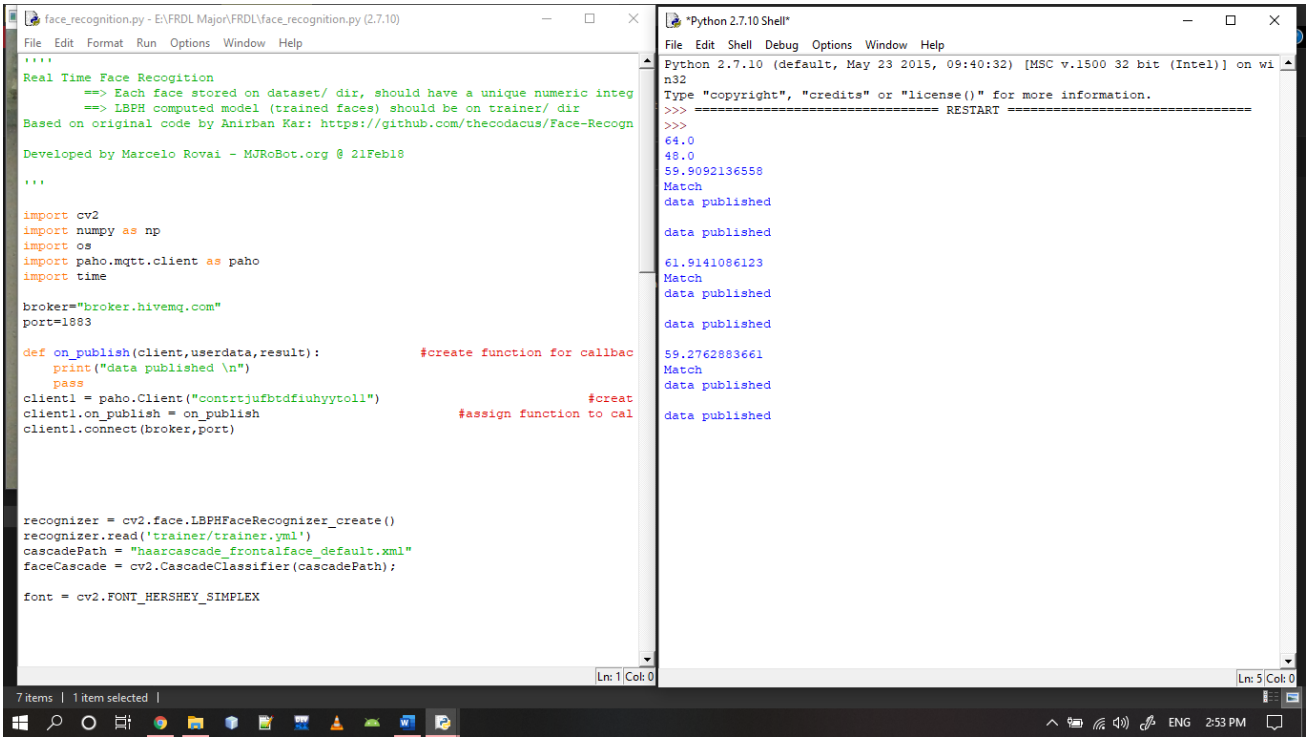



Fig. 4.3 Screenshot 3

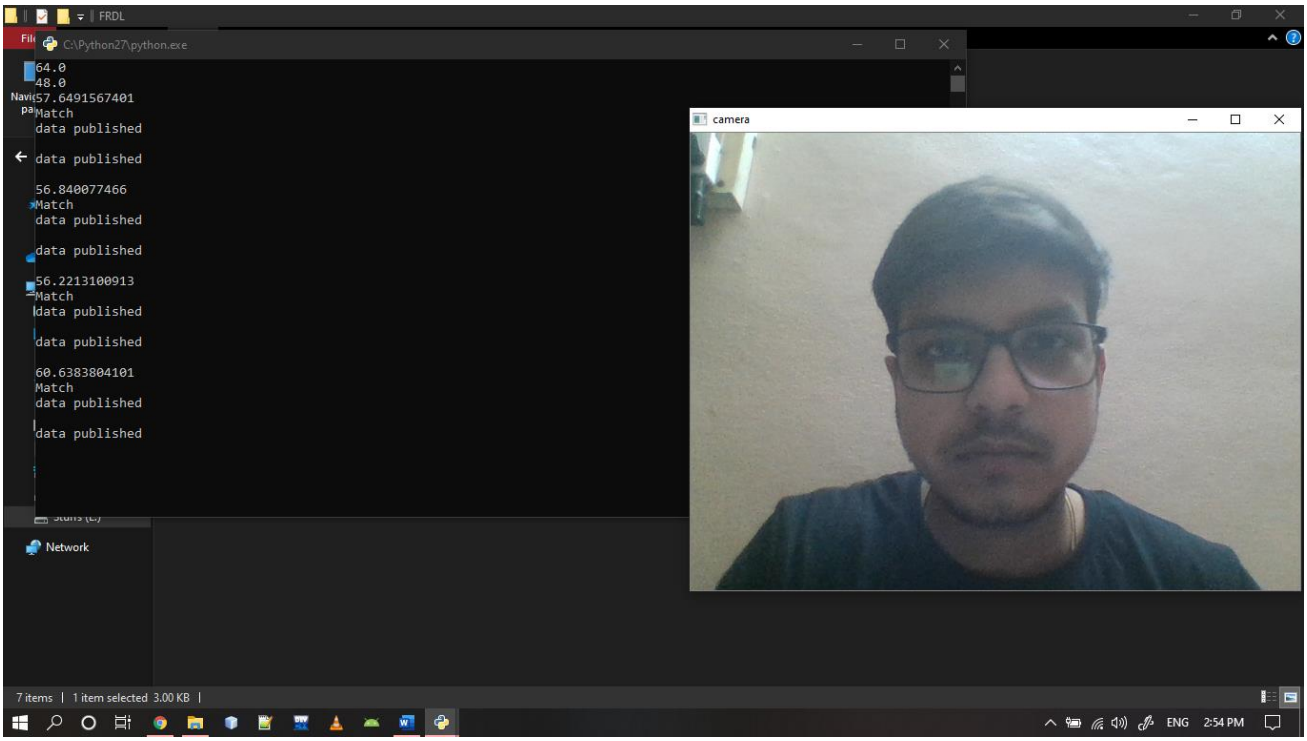



Fig. 4.4 Screenshot 4

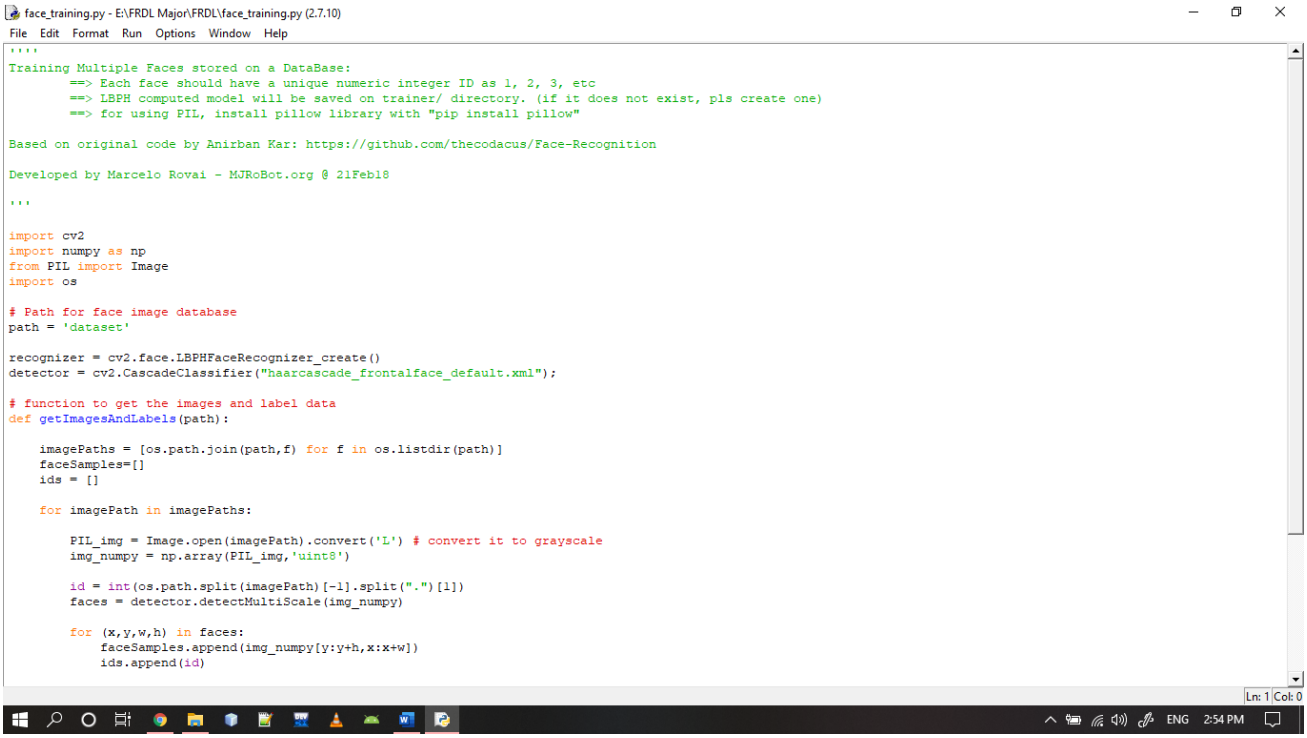



Fig. 4.5 Screenshot 5

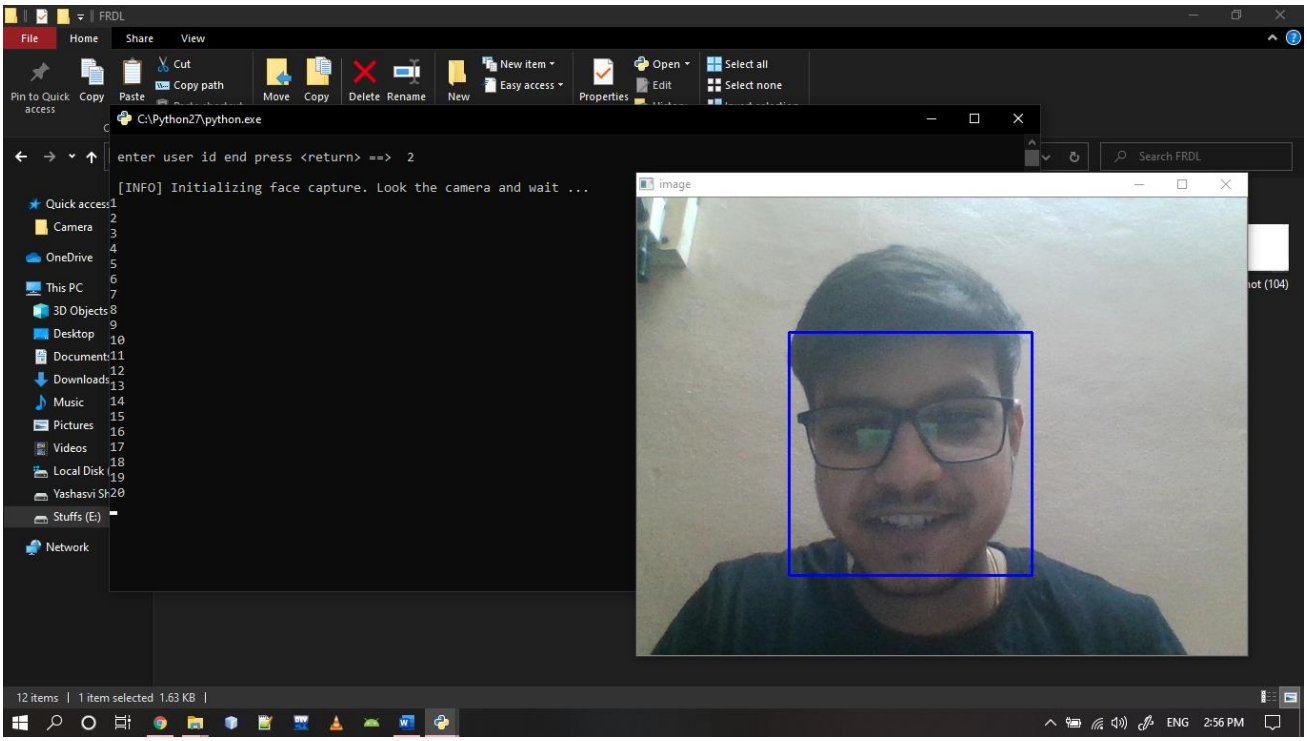



Fig. 4.6 Screenshot 6

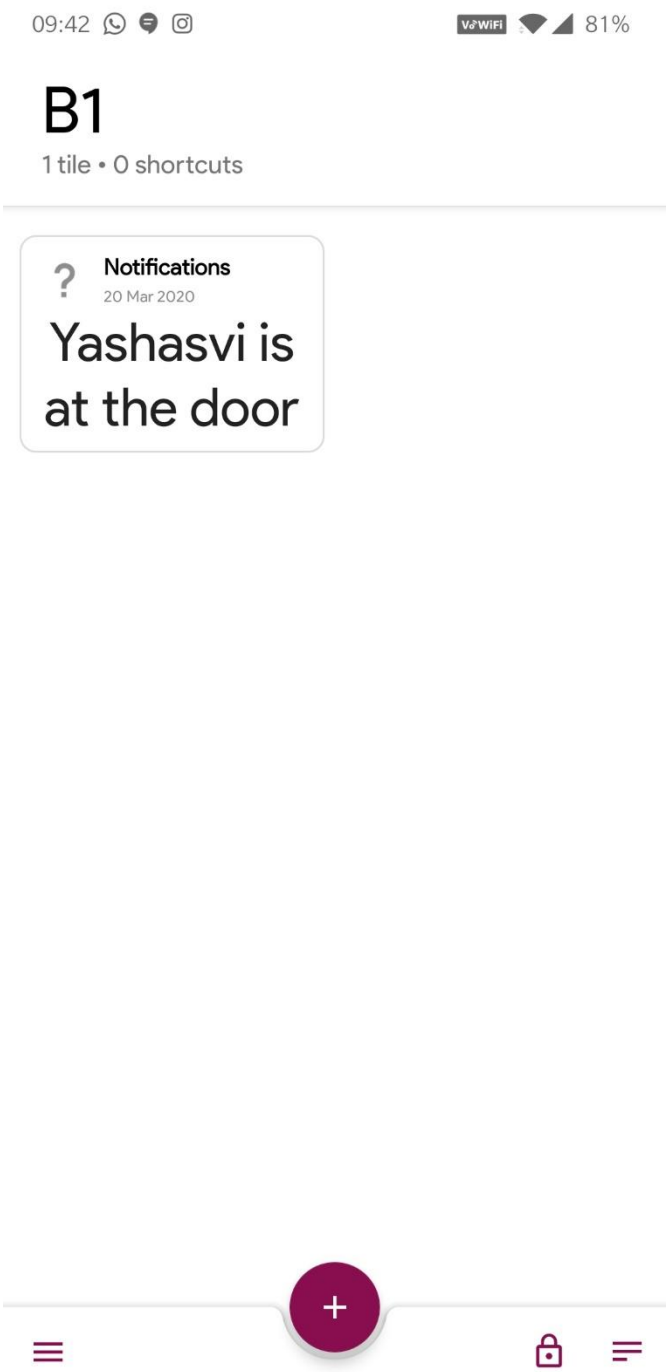



Fig 4.7 Screenshot 7

# Chapter 5: Conclusion

In this proposed door access system by using face recognition the images are stored in a data base. This system is used door lock access for Residential and Commercial Purposes. Here we have designed a highly secured door locking system by using Digital Circuit and Laptop Webcam. This system has been used with home door lock access control based on face recognition method by verifying enrolled facial images. Concern persons will be informed successfully about the person detection via e-mail alert generations along with details attached. Face recognition is one of the several techniques for recognizing people. There are several methods that can be used for that purpose. Though there are other new techniques more simple to understand the use and implement but also with very good performance.

# Bibliography


[1] (2017) “Comparative Analysis for a Real Time Face Recognition System Using Raspberry Pi” Muhammad Kashif Shaikh, Syed Annas Bin Mazhar.

[2] (2017)”Secured Room Access Module” Suchit Shavi.

[3] (2017) “Automatic Semantic Face Recognition”: Mark S. Nixon University of Southampton Southampton, United Kingdom

[4] (2017) “Real-Time Implementation of face recognition system” by Neel Ramakant Borkar and Sonia Kuwelkar,
India

[5] (2017) “IoT based Home security through Digital Image Process Algorithms” by A. Beatrice, Dr S. Britto Ramesh Kumar and J. Jerlin Sharmila, India

[6] (2017) “Secured Room Access Module” by Suchit and Shanvi, India

[7] (2017) “Door locking system via web application” Charoen Vongchumyen, Watjanapong Kasemsiri, Kiatnarong Tongprasert, Aranya Walairacht, Pattaya.
(2016) “Arduino Based Door Unlocking System with RealTime Control” Somjit Nath, Paramita Banerjee, Rathindra Nath, Biswas, Swarup Kumar, Mitra

[8] (2014) K.Gopalakrishnan, V.Sathish Kumar “embedded image capturing system using the raspberry pi system” international Journal.

[9] (2014) “Development of Intelligent Automatic Door System” Daiki Nishida, Kumiko Tsuzura1, Shunsuke Kudoh1, Kazuo Takai, Tatsuhiro Momodori.

[10] (2012) “Face Recognition Based on Magnetic Door Lock System Using Microcontroller” Harnani Hassan, Raudah, Abu Bakar Ahmad Faculty of Electrical Engineering.

[11] (2005) “Real-time Embedded Face Recognition for Smart Home” by F. Zuo and P. H. N. de.

[12] (2000) “Automatic Door Opener” Pik-Yiu Chan, John D. Enderle.

# Source Code

**SOURCE CODE FOR COLLECTION OF DATA SET:**

```
""

Capture multiple Faces from multiple users to be stored on a DataBase (dataset directory)

    ==> Faces will be stored on a directory: dataset/ (if does not exist, pls create one)

    ==> Each face will have a unique numeric integer ID as 1, 2, 3, etc


Based on original code by Anirban Kar: https://github.com/thecodacus/Face-Recognition


Developed by Marcelo Rovai - MJRoBot.org @ 21Feb18


'''


import cv2

import os


cam = cv2.VideoCapture(0)

cam.set(3, 640) # set video width

cam.set(4, 480) # set video height


face_detector = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')


# For each person, enter one numeric face id

face_id = input('\n enter user id end press <return> ==>  ')
```

```
print("\n [INFO] Initializing face capture. Look the camera and wait ...")
# Initialize individual sampling face count
count = 0

while(True):

    ret, img = cam.read()
#     img = cv2.flip(img, -1) # flip video image vertically
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    faces = face_detector.detectMultiScale(gray, 1.3, 5)

    for (x,y,w,h) in faces:

        cv2.rectangle(img, (x,y), (x+w,y+h), (255,0,0), 2)
        count += 1

        # Save the captured image into the datasets folder
        cv2.imwrite("dataset/User." + str(face_id) + '.' + str(count) + ".jpg", gray[y:y+h,x:x+w])

        cv2.imshow('image', img)
        print(count)

    k = cv2.waitKey(100) & 0xff # Press 'ESC' for exiting video
    if k == 27:
        break
```

```
    elif count >= 30: # Take 30 face sample and stop video
        break

# Do a bit of cleanup
print("\n [INFO] Exiting Program and cleanup stuff")
cam.release()
cv2.destroyAllWindows()
```

**SOURCE CODE FOR TRAINING OF DATA SET:**

```
"""
Training Multiple Faces stored on a DataBase:
    ==> Each face should have a unique numeric integer ID as 1, 2, 3, etc
    ==> LBPH computed model will be saved on trainer/ directory. (if it does not exist, pls create one)
    ==> for using PIL, install pillow library with "pip install pillow"

Based on original code by Anirban Kar: https://github.com/thecodacus/Face-Recognition

Developed by Marcelo Rovai - MJRoBot.org @ 21Feb18

"""

import cv2
import numpy as np
from PIL import Image
import os

# Path for face image database
path = 'dataset'

recognizer = cv2.face.LBPHFaceRecognizer_create()
detector = cv2.CascadeClassifier("haarcascade_frontalface_default.xml");
```

```
# function to get the images and label data
def getImagesAndLabels(path):

    imagePaths = [os.path.join(path,f) for f in os.listdir(path)]
    faceSamples=[]
    ids = []

    for imagePath in imagePaths:

        PIL_img = Image.open(imagePath).convert('L') # convert it to grayscale
        img_numpy = np.array(PIL_img,'uint8')

        id = int(os.path.split(imagePath)[-1].split(".")[1])
        faces = detector.detectMultiScale(img_numpy)

        for (x,y,w,h) in faces:
            faceSamples.append(img_numpy[y:y+h,x:x+w])
            ids.append(id)

    return faceSamples,ids

print ("\n [INFO] Training faces. It will take a few seconds. Wait ...")
faces,ids = getImagesAndLabels(path)
recognizer.train(faces, np.array(ids))
```

```
# Save the model into trainer/trainer.yml
recognizer.write('trainer/trainer.yml') # recognizer.save() worked on Mac, but not on Pi


# Print the numer of faces trained and end program
print("\n [INFO] {0} faces trained. Exiting Program".format(len(np.unique(ids))))
```

**SOURCE CODE FOR SERVER:**

```
#!/usr/bin/env python

"""Simple HTTP Server With Upload.

This module builds on BaseHTTPServer by implementing the standard GET
and HEAD requests in a fairly straightforward manner.

"""


__version__ = "0.1"
__all__ = ["SimpleHTTPRequestHandler"]
__author__ = "bones7456"
__home_page__ = "http://li2z.cn/"

import os
import posixpath
import BaseHTTPServer
import urllib
import cgi
import shutil
import mimetypes
import re
```

```
import cv2
import numpy as np
import os
from matplotlib import pyplot as plt

try:
    from cStringIO import StringIO
except ImportError:
    from StringIO import StringIO


class SimpleHTTPRequestHandler(BaseHTTPServer.BaseHTTPRequestHandler):

    """Simple HTTP request handler with GET/HEAD/POST commands.

    This serves files from the current directory and any of its
    subdirectories.  The MIME type for files is determined by
    calling the .guess_type() method. And can reveive file uploaded
    by client.

    The GET/HEAD/POST requests are identical except that the HEAD
    request omits the actual contents of the file.

    """
```

```
server_version = "SimpleHTTPWithUpload/" + __version__

def do_GET(self):
    """Serve a GET request."""
    f = self.send_head()
    if f:
        self.copyfile(f, self.wfile)
        f.close()

def do_HEAD(self):
    """Serve a HEAD request."""
    f = self.send_head()
    if f:
        f.close()

def do_POST(self):
    """Serve a POST request."""
    r, info = self.deal_post_data()
    print r, info, "by: ", self.client_address
    f = StringIO()
    f.write('<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">')
    f.write("<html>\n<title>Upload Result Page</title>\n")
    f.write("<body>\n<h2>Upload Result Page</h2>\n")
    f.write("<hr>\n")
    if r:
```

```
            f.write("<strong>Success:</strong>")
        else:
            f.write("<strong>Failed:</strong>")
        f.write(info)
        f.write("<br><a href=\"%s\">back</a>" % self.headers['referer'] if 'referer' in self.headers else
")
        print("++++++++++++++")


        fileLocation = info.strip("' upload success!")
        fileLocation = fileLocation.strip("File '")
        print(fileLocation)



        #========================================
        id = 0
        names = ['None', 'Nikhil', 'Krati', 'Ilza', 'Z', 'W']
        recognizer = cv2.face.LBPHFaceRecognizer_create()
        recognizer.read('trainer/trainer.yml')
        cascadePath = "haarcascade_frontalface_default.xml"
        faceCascade = cv2.CascadeClassifier(cascadePath);
        font = cv2.FONT_HERSHEY_SIMPLEX

        img = cv2.imread(fileLocation)
#         plt.imshow(img, cmap = 'gray', interpolation = 'bicubic')
#         plt.xticks([]), plt.yticks([])  # to hide tick values on X and Y axis
#         plt.show()
```

```
        gray = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
        print("SHOWIUNG")
#         cv2.imshow('camera',img)
        print("ASDFGHJKJHGFDSA")
        faces = faceCascade.detectMultiScale(
            gray,
            scaleFactor = 1.2,
            minNeighbors = 5,
            minSize = (64, 48),
           )

        for(x,y,w,h) in faces:

            cv2.rectangle(img, (x,y), (x+w,y+h), (0,255,0), 2)

            id, confidence = recognizer.predict(gray[y:y+h,x:x+w])

            # Check if confidence is less them 100 ==> "0" is perfect match
            if (confidence < 100):
                id = names[id]
                confidence = "  {0}%".format(round(100 - confidence))
            else:
                id = "unknown"
                confidence = "  {0}%".format(round(100 - confidence))
```

```
    cv2.putText(img, str(id), (x+5,y-5), font, 1, (255,255,255), 2)
    cv2.putText(img, str(confidence), (x+5,y+h-5), font, 1, (255,255,0), 1)

    print(confidence)
    print(id)
    f.write("<H2>" + str(confidence) + "  " + str(id)  + "<h2>")

#========================================

print("============================")
f.write("<hr><small>Powerd By: bones7456, check new version at ")
f.write("<a href=\"http://li2z.cn/?s=SimpleHTTPServerWithUpload\">")
f.write("here</a>.</small></body>\n</html>\n")
length = f.tell()
f.seek(0)
```

```
    self.send_response(200)
    self.send_header("Content-type", "text/html")
    self.send_header("Content-Length", str(length))
    self.end_headers()
    if f:
        self.copyfile(f, self.wfile)
        f.close()

def deal_post_data(self):
    boundary = self.headers.plisttext.split("=")[1]
    remainbytes = int(self.headers['content-length'])
    line = self.rfile.readline()
    remainbytes -= len(line)
    if not boundary in line:
        return (False, "Content NOT begin with boundary")
    line = self.rfile.readline()
    remainbytes -= len(line)
    fn = re.findall(r'Content-Disposition.*name="file"; filename="(.*)"', line)
    if not fn:
        return (False, "Can't find out file name...")
    path = self.translate_path(self.path)
    fn = os.path.join(path, fn[0])
    line = self.rfile.readline()
    remainbytes -= len(line)
    line = self.rfile.readline()
```

```
    remainbytes -= len(line)
    try:
        out = open(fn, 'wb')
    except IOError:
        return (False, "Can't create file to write, do you have permission to write?")

    preline = self.rfile.readline()
    remainbytes -= len(preline)
    while remainbytes > 0:
        line = self.rfile.readline()
        remainbytes -= len(line)
        if boundary in line:
            preline = preline[0:-1]
            if preline.endswith('\r'):
                preline = preline[0:-1]
            out.write(preline)
            out.close()
            return (True, "File '%s' upload success!" % fn)
        else:
            out.write(preline)
            preline = line
    return (False, "Unexpect Ends of data.")

def send_head(self):
    """Common code for GET and HEAD commands.
```

This sends the response code and MIME headers.

Return value is either a file object (which has to be copied
to the outputfile by the caller unless the command was HEAD,
and must be closed by the caller under all circumstances), or
None, in which case the caller has nothing further to do.

"""

```
print("send_head()")
path = self.translate_path(self.path)
f = None
if os.path.isdir(path):
    if not self.path.endswith('/'):
        # redirect browser - doing basically what apache does
        self.send_response(301)
        self.send_header("Location", self.path + "/")
        self.end_headers()
        return None
    for index in "index.html", "index.htm":
        index = os.path.join(path, index)
        if os.path.exists(index):
            path = index
            break
    else:
```

```
            return self.list_directory(path)
        ctype = self.guess_type(path)
        try:
            # Always read in binary mode. Opening files in text mode may cause
            # newline translations, making the actual size of the content
            # transmitted *less* than the content-length!
            f = open(path, 'rb')
        except IOError:
            self.send_error(404, "File not found")
            return None
        self.send_response(200)
        self.send_header("Content-type", ctype)
        fs = os.fstat(f.fileno())
        self.send_header("Content-Length", str(fs[6]))
        self.send_header("Last-Modified", self.date_time_string(fs.st_mtime))
        self.end_headers()
        return f



    def list_directory(self, path):
        """Helper to produce a directory listing (absent index.html).

        Return value is either a file object, or None (indicating an
        error).  In either case, the headers are sent, making the
        interface the same as for send_head().
```

```
"""
print("list_directory()")
try:
    list = os.listdir(path)
except os.error:
    self.send_error(404, "No permission to list directory")
    return None
list.sort(key=lambda a: a.lower())
f = StringIO()
displaypath = cgi.escape(urllib.unquote(self.path))
f.write('<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">')
f.write("<html>\n<title>Directory listing for %s</title>\n" % displaypath)
f.write("<body>\n<h2>Directory listing for %s</h2>\n" % displaypath)
f.write("<hr>\n")
f.write("<form ENCTYPE=\"multipart/form-data\" method=\"post\">")
f.write("<input name=\"file\" type=\"file\"/>")
f.write("<input type=\"submit\" value=\"upload\"/></form>\n")
f.write("<hr>\n<ul>\n")
for name in list:
    fullname = os.path.join(path, name)
    displayname = linkname = name
    # Append / for directories or @ for symbolic links
    if os.path.isdir(fullname):
        displayname = name + "/"
```

```
            linkname = name + "/"
        if os.path.islink(fullname):
            displayname = name + "@"
            # Note: a link to a directory displays with @ and links with /
        f.write('<li><a href="%s">%s</a>\n'
                % (urllib.quote(linkname), cgi.escape(displayname)))
    f.write("</ul>\n<hr>\n</body>\n</html>\n")
    length = f.tell()
    f.seek(0)
    self.send_response(200)
    self.send_header("Content-type", "text/html")
    self.send_header("Content-Length", str(length))
    self.end_headers()
    return f

def translate_path(self, path):
    """Translate a /-separated PATH to the local filename syntax.

    Components that mean special things to the local file system
    (e.g. drive or directory names) are ignored.  (XXX They should
    probably be diagnosed.)

    """
    print("translate_path()")
    # abandon query parameters
```

```
    path = path.split('?',1)[0]
    path = path.split('#',1)[0]
    path = posixpath.normpath(urllib.unquote(path))
    words = path.split('/')
    words = filter(None, words)
    path = os.getcwd()
    for word in words:
        drive, word = os.path.splitdrive(word)
        head, word = os.path.split(word)
        if word in (os.curdir, os.pardir): continue
        path = os.path.join(path, word)
    return path


def copyfile(self, source, outputfile):
    """Copy all data between two file objects.

    The SOURCE argument is a file object open for reading
    (or anything with a read() method) and the DESTINATION
    argument is a file object open for writing (or
    anything with a write() method).

    The only reason for overriding this would be to change
    the block size or perhaps to replace newlines by CRLF
    -- note however that this the default server uses this
```

```
    to copy binary data as well.

    """
    print("copyfile()")
    shutil.copyfileobj(source, outputfile)

def guess_type(self, path):
    """Guess the type of a file.

    Argument is a PATH (a filename).

    Return value is a string of the form type/subtype,
    usable for a MIME Content-type header.

    The default implementation looks the file's extension
    up in the table self.extensions_map, using application/octet-stream
    as a default; however it would be permissible (if
    slow) to look inside the data to make a better guess.

    """

    print("guess_type()")
    base, ext = posixpath.splitext(path)
    if ext in self.extensions_map:
        return self.extensions_map[ext]
```

```
        ext = ext.lower()
        if ext in self.extensions_map:
            return self.extensions_map[ext]
        else:
            return self.extensions_map['']

    if not mimetypes.inited:
        mimetypes.init() # try to read system mime.types
    extensions_map = mimetypes.types_map.copy()
    extensions_map.update({
        '': 'application/octet-stream', # Default
        '.py': 'text/plain',
        '.c': 'text/plain',
        '.h': 'text/plain',
        })


def test(HandlerClass = SimpleHTTPRequestHandler,
         ServerClass = BaseHTTPServer.HTTPServer):
    BaseHTTPServer.test(HandlerClass, ServerClass)

if __name__ == '__main__':
    test()
```

**SOURCE CODE FOR FACIAL RECOGNITION DOOR LOCK:**

```
""
Real Time Face Recogition
    ==> Each face stored on dataset/ dir, should have a unique numeric integer ID as 1, 2, 3, etc
    ==> LBPH computed model (trained faces) should be on trainer/ dir
Based on original code by Anirban Kar: https://github.com/thecodacus/Face-Recognition

Developed by Marcelo Rovai - MJRoBot.org @ 21Feb18

'''

import cv2
import numpy as np
import os
import paho.mqtt.client as paho
import time

broker="broker.hivemq.com"
port=1883

def on_publish(client,userdata,result):             #create function for callback
   print("data published \n")
   pass
client1 = paho.Client("contrtjufbtdfiuhyytol1")                    #create client object
client1.on_publish = on_publish                          #assign function to callback
```

```
client1.connect(broker,port)




recognizer = cv2.face.LBPHFaceRecognizer_create()

recognizer.read('trainer/trainer.yml')

cascadePath = "haarcascade_frontalface_default.xml"

faceCascade = cv2.CascadeClassifier(cascadePath);


font = cv2.FONT_HERSHEY_SIMPLEX




#iniciate id counter

id = 0


# names related to ids: example ==> Marcelo: id=1,  etc

names = ['None', 'Yashraj', 'Yashasvi', 'Saiel', 'Yuvraj', 'Saquib', 'Raksha']


# Initialize and start realtime video capture

cam = cv2.VideoCapture(0)

cam.set(3, 640) # set video widht
```

```
cam.set(4, 480) # set video height

# Define min window size to be recognized as a face
minW = 0.1*cam.get(3)
minH = 0.1*cam.get(4)


print(minW)
print(minH)
i = 0;


i = i + 1
while True:
    ret, img =cam.read()
#     img = cv2.flip(img, -1) # Flip vertically

    #client1.publish("leotech/door", i)
    i=i+1
    gray = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)

    faces = faceCascade.detectMultiScale(
        gray,
        scaleFactor = 1.5,
        minNeighbors = 10,
```

```
    minSize = (int(minW), int(minH)),
   )
for(x,y,w,h) in faces:

    cv2.rectangle(img, (x,y), (x+w,y+h), (0,255,0), 2)
    id, confidence = recognizer.predict(gray[y:y+h,x:x+w])

    # Check if confidence is less them 100 ==> "0" is perfect match
    if (confidence < 100):
        con = 100.0 - confidence
        id = names[id]
        confidence = "  {0}%".format(round(100 - confidence))

        print(con)
        if(con > 40.0):
            print("Match")
            client1.publish("/acro/face/name", id + " is at the door")
            client1.publish("/acro/face", '1')
            time.sleep(5)
        else:
            print("Not Match")

    else:
        id = "unknown"
        confidence = "  {0}%".format(round(100 - confidence))
```

```
        cv2.putText(img, str(id), (x+5,y-5), font, 1, (255,255,255), 2)
        cv2.putText(img, str(confidence), (x+5,y+h-5), font, 1, (255,255,0), 1)

    cv2.imshow('camera',img)

    k = cv2.waitKey(10) & 0xff # Press 'ESC' for exiting video
    if k == 27:
        break

# Do a bit of cleanup
print("\n [INFO] Exiting Program and cleanup stuff")
cam.release()
cv2.destroyAllWindows()
```