*Research Inventy: International Journal of Engineering And Science*
*Vol.14, Issue 5 (May 2024), PP 49-56*
*Issn (e): 2278-4721, Issn (p):2319-6483, www.researchinventy.com*

# Face Recognition Door Lock System using Raspberry Pi

[1] Vishwajeet Dhananjay Patil, [2] Ritesh Sugriv Dolare, [3] Atharva Yuvraj Ghatge
[123]*Students, Department of Electronics & Telecommunication, KIT's College of Engineering, Maharashtra, India Corresponding Author: Prof.Mrs. M.V.Gangapure*

***Abstract:*** *One of the crucial difficulties we aim to find in computer vision is to recognize items automatically without human interaction in a picture. Face detection may be seen as an issue when the face of human beings is detected in a picture. The initial step towards many face-related technologies, including face recognition or verification, is generally facial detection. Face detection however may be quite beneficial. A biometric identification system besides fingerprint and iris would likely be the most effective use of face recognition. The door lock system in this project consists of Raspberry Pi, camera module, relay module, power input and output, connected to a solenoid lock. It employs the two different facial recognition algorithms to detect the faces and train the model for recognition purpose.*

***Keywords:*** *Face recognition, raspberry pi, opencv.*

Date of Submission: 29-04-2024 Date of acceptance: 08-05-2024

## I. INTRODUCTION

Robbery and illegal entrance are both regular occurrences in today's society. As a result, security is important in everyday life. People are usually preoccupied with their daily activities, but they also want to ensure the safety of their valuable items. They are prone to misplacing important things such as keys, wallets, and credit cards. If they don't have these, they won't be able to get to their residence or any other place. A lock opener, an authentication password, an RFID tag, or an ID swipe card are mostly needed for getting into a normal security system. On the other side, many security measures have flaws, such as the danger of being forgotten or stolen by unauthorized people. As a result, enhanced authentication solutions that provide a greater level of security as a template are needed [1]. Biometric Authentication is one of them. The word "Biometrics" refers to the measuring of human traits. In computer science, biometric authentication is a way of identity and access control. It's also used to figure out who's who in a group of people that are being watched. Individuals are identified using unique biometric identifiers with measurable traits. Physiological biometric indicators are widely used to identify and categorize people. In contrast to behavioural attributes, the shape of the body has an effect on physiological characteristics. Fingerprints, palm veins, facial recognition, DNA, palm print, hand geometry, eye identification, retina, and odour/scent are only a few examples [2]. A. Face Recognition Face Recognition looks to be one of the most widespread, collected, and accessible technologies among the above listed ways of biometric identifications. Because it relies on the same identifier that people use to differentiate one person from another, biometric face recognition, also known as Automatic Face Recognition (AFR) [2], is a particularly appealing biometric method. Understanding the complicated human visual system and how people represent faces in order to differentiate between different identities is one of its key focuses. Faces can be separated according to their algorithms. There are several face recognition techniques. These includes: Classical face recognition approach, holistic approach, statistical approach, model- based approach, feature based, artificial intelligence approach, hybrid approach, Gabor wavelets approach, Face descriptor- based methods, 3D- based face recognition and Video- based face recognition.

## BACKGROUND THEORY

Majority of face authentication systems look for distinguishing features such as the length of the nose, the wideness of the eyes, the width and angle of the jaw, the prominence of the cheekbones, and the spacing between the eyes, and gives an ID to each. The system then compares that picture to another image using these numerical values and determines how collinear images are. Face recognition image origin includes pre- existing photographs from numerous angles as well as video camera feeds. Face detection, Pre-processing, Feature extraction, and Feature matching/Face recognition are all elements of a facial recognition system.

Now the important point in face recognition is, both the detection of faces and recognition have same four common processes namely,

1) Acquisition/ Face Detection
2) Pre-Processing

3) Feature Extraction
4) Training

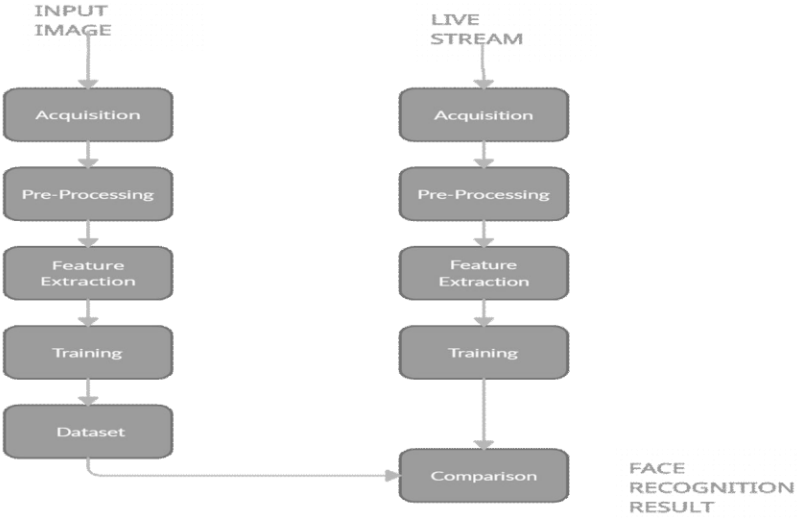


Fig. 1 Process Flow after recognizing a face while creating data set and while recognition process.

In this project, both of these processes experience four common procedures namely, face acquisition, face detection using Haar cascade classifier and pre-processing, feature extraction using Linear Binary Pattern algorithm to compute LBP values. These values are kept in the database only in case of processing an input image. Finally, comparison of the values in the database with the values computed via live streaming takes place which recognizes the human face as known or unknown based on the matches.

A. Face Detection Face detection algorithms come in various types, each with its own set of flaws and benefits. Some employ skin tones, while others employ outlines, and still, others utilize templates, neural networks, or filters. The problem with these methods is that they are computationally costly. A picture is nothing more than a set of pigment and/or light intensity numbers. Because of the significant differences in form and pigmentation inside a human face, analyzing these pixels for face recognition takes time and effort. Re-analysis of pixels is frequently needed for scaling and accuracy. Viola and Jones created Haar Classifiers, a method that uses AdaBoost classifier cascades based on Haar-like properties rather than pixels to quickly recognize any object, including human faces. This is the algorithm that we're using in this project for identifying the faces

1) Haarcascade Algorithm: Haar-like properties are the basis for Haar categorizer object detection. Rather of using a pixel's illumination values, these traits employ the difference in brightness values between consecutive quadrilateral groupings of pixels. To identify relative bright and dull regions, the contrast variances between pixel sets are employed. A Haar-like feature is made by two or three neighbouring sets with a relative brightness variance. To detect a picture, Haar like characteristics are applied, as illustrated in Fig 2. By adjusting the index of pixel set under investigation, Haar characteristics may be easily scaled.

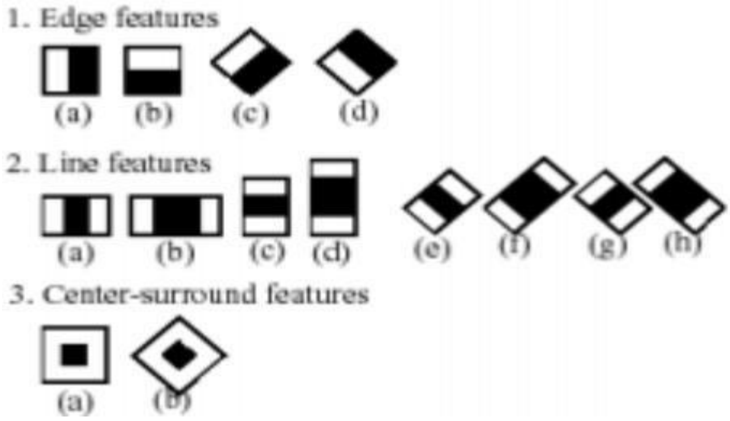


Fig. 2 Common Haar Features

B. Pre-Processing and Feature Extraction

For feature extraction, we’re using LBPH (Local Binary Pattern Histogram) method is utilized. The Linear Binary Pattern Histogram is the most popular choice for "Feature extraction". It works with a great deal of judgment. Using this technique, the image's features will be taken in real-time. The Linear Binary Pattern Histogram method is split into two phases: training and assessment. During the training phase, picture samples are taught to be recognized, and during the estimation phase, the picture to be tested is compared to the samples learned in the dataset.

1) LBPH Algorithm: Following the identification of the face, the following step is to extract features using the linear binary pattern technique. The test image is converted to greyscale as the first stage in this method. This picture, which is L x M pixels in size, will be split into areas. The areas have the same pixel size, resulting in n × n regions. Linear binary pattern operator will be applied to each area. In this process, it will compare the middle pixel with its neighbour pixels. If the pixel size is bigger to centre pixel it is ‘1’ or it is ‘0. Accordingly, the LBP code for the centre pixel is produced by joining the neighbour pixel values (ones or zeroes) into a binary code, which is converted to 256-dimensional decimal for ease as a texture description of the centre pixel.

**SYSTEM BLOCK DIAGRAM**

I. A. Block Diagram The architecture of the Raspberry pi facial recognition system is smaller, easier than the PC-based facial recognition system, and has lower power usage. It is freer to build applications on Linux thanks to open-source code. For the face detection and identification method, the principal component analysis (Eigen faces) algorithm is used. The system is inexpensive, fast, highly durable and offers sufficient versatility to meet various system requirements. Using technology:

II. 1) Image Processing: This process is used for image capture and recognition compared with database images.

III. 2) Embedded System Design: This method is used for the module, which combines hardware, software and many other featured components.

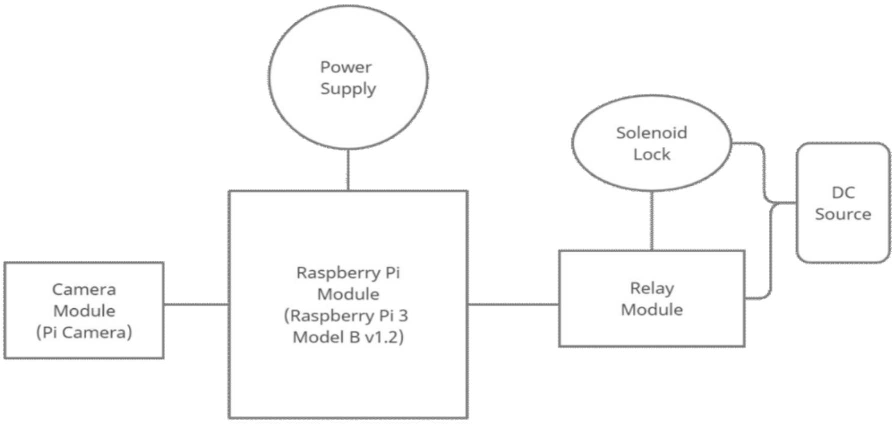
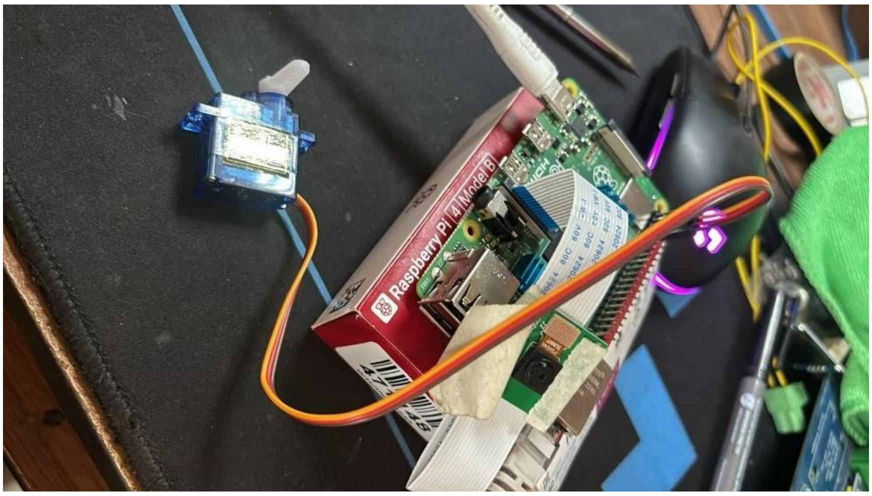

Power Supply
Solenoid Lock
DC Source
Camera Module (Pi Camera)
Raspberry Pi Module (Raspberry Pi 3 Model B v1.2)
Relay Module

Fig.4 Block layout of the system Raspberry Pi

IV. GPIO pins give an output of 3.3V only. The solenoid lock needs 9 -12V to operate. So we used an external DC source and relay linked with pi to work the lock. The VCC and GND of the relay module attached to Raspberry Pi pins. Then the signal pin of the relay module is given to the GPIO 26 of Raspberry Pi. On the other side of the relay module, connect



V. the negative from the external DC source to the negative of the solenoid door lock. Connect the positive from the DC power source to the common of the relay module and then connect normally open from the relay module to positive of the solenoid door lock.

## II. EXPERIMENTAL PROCEDURE

The project is designed and executed with the help of Raspberry Pi for the door unlocking, which ensures that our homes are safely protected. Raspberry Pi operates the video camera to record images and monitors them. Open CV/ Python Library is developed by using a saved facet database as a given picture of a scene to recognize or check one or more people in the scene. The pictures are then derived and will match photos from the collection. The door opens immediately if the pictures are paired. Otherwise, the door lock stays closed.

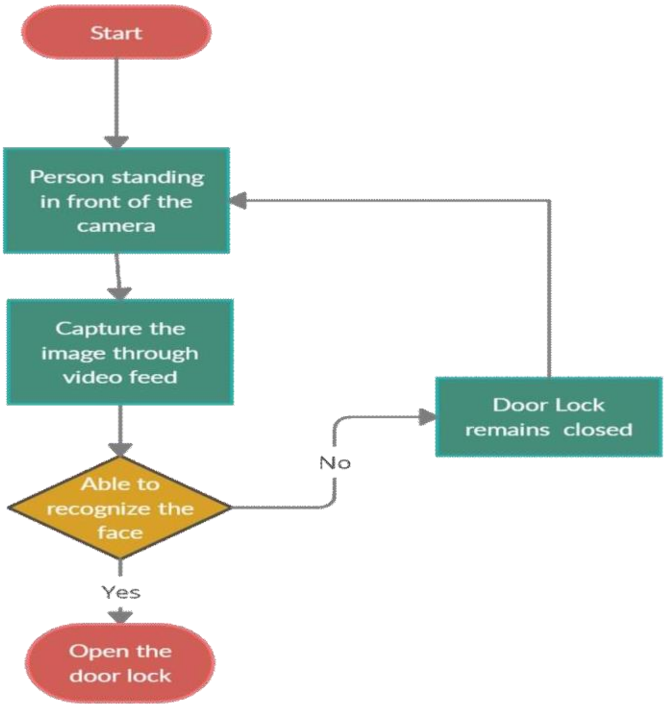

Start
Person standing in front of the camera
Capture the image through video feed
Able to recognize the face
No
Door Lock remains closed
Yes
Open the door lock

Fig.6 Flow Chart of the process

The architecture of the Raspberry pi facial recognition system is smaller, easier than the PC-based facial recognition system, and has lower power usage. It is freer to build applications on Linux thanks to open-source code. For the face detection and identification method, the principal component analysis (Eigen faces) algorithm is used. The system is inexpensive, fast, highly durable and offers sufficient versatility to meet various system requirements. Using technology:

1) Image Processing: This process is used for image capture and recognition compared with database

images.

2) Embedded System Design: This method is used for the module, which combines hardware, software and many other featured components. In this project, we worked on 3 very different phases for Face Recognition:

A. Collect the Data

We worked with face recognition using this Haarcascade classifier. We need to extract features from positive and negative images and train the model which then identifies the faces. The OpenCV consists of a teacher as well as a detector. OpenCV already features many pre-trained classifiers for face, eyes, smile, etc. Those XML files can be download from Haarcascades directory. This python code consists of 2 stages face detection and then gathering the detected data into a dataset. For getting the data into the dataset, create a folder to hold the data inside the project directory.

B. Train the Model

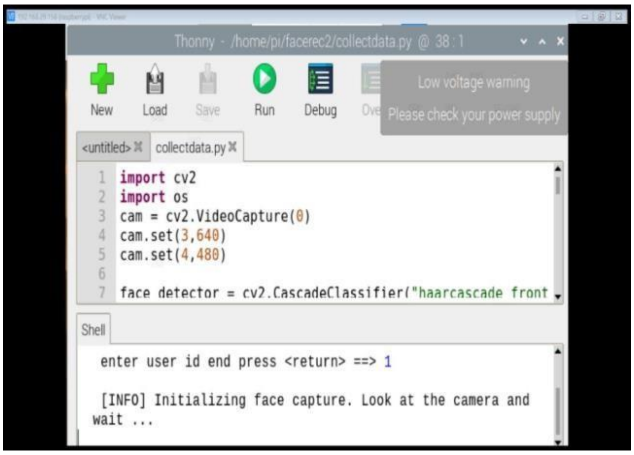
On this second step, we took all user data from the collection and train the OpenCV Recognizer. This is done straight by a specific OpenCV method. The result will be a .yml file that will be saved on a “trainer/” path. A new python file training.py is made to write the trainer program.

C. Recognize the Faces

In the final step of the project, we will capture a fresh face on our camera through a live video stream and if this human’s face is snapshotted and features extracted before, the recognizer model makes a prediction returning the person’s name and an index, shown how confident the recognizer is with this match. And then the solenoid lock is opened to which the electricity is send by raspberry Pi through the relay module.

## III. RESULTS AND DISCUSSIONS

A. Collecting the Data When this program is run, first it asks us to enter the face id to store the images taken later under that face id. Then it tells us to look at the camera and wait till the face images are captured. Since it’s only a single camera, it won’t take much time.



Low voltage warning
Please check your power supply

```
import cv2
import os
cam = cv2.VideoCapture(0)
cam.set(3,640)
cam.set(4,480)

face detector = cv2.CascadeClassifier("haarcascade front
```

```
 enter user id end press <return> ==> 1

 [INFO] Initializing face capture. Look at the camera and
wait ...
```

Fig. 7 Face ID

The camera is turned on and it starts taking pictures of what is present in front of the camera. It does identify the face and takes frames of the video capture happening at live. We took 90 images per face, i.e., 90 frames throughout the video clip. Given below are the information frames of face 1and face 2. They are put into the dataset directory which is inside the project directory.

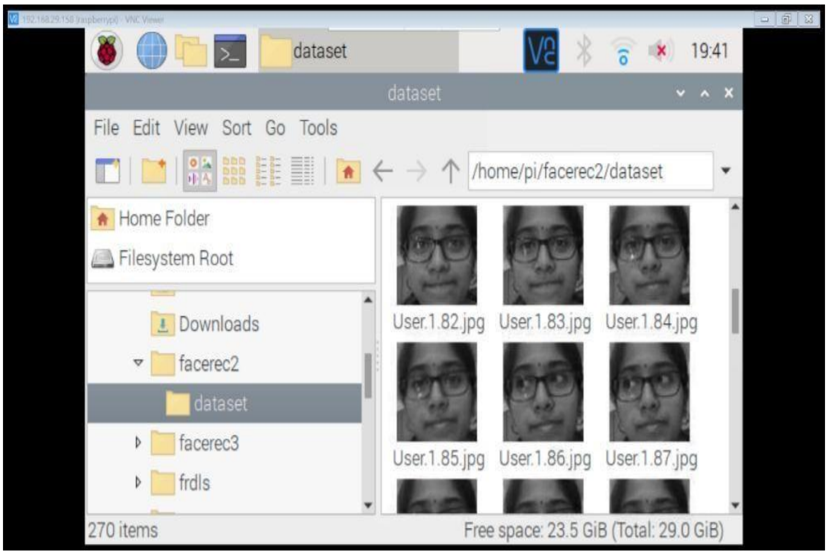
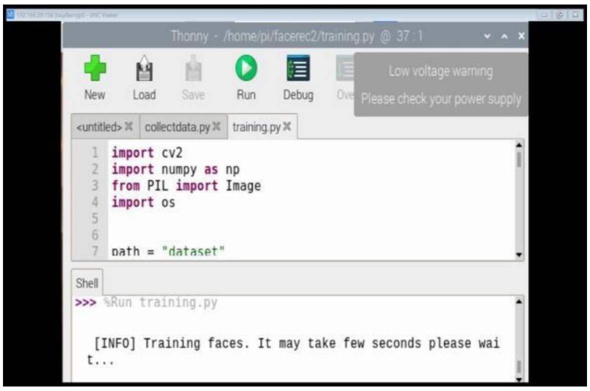


Fig. 8 Face id =1 face frames

After the detection of 90 faces is completed, the program stops the live capture and tells us at the output window that face detection is completed.

B. Training the Data

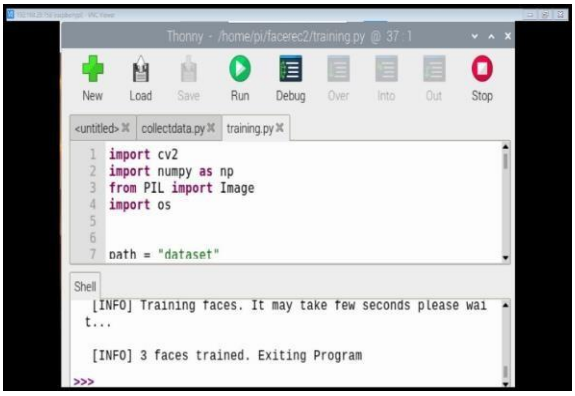
After getting the data, the data is to be trained. So training.py program is run each and every time a new face is identified. It trains all the faces found in the dataset directory everytime. More the faces, more is the time taken by the computer to extract all the required features from the dataset that are to perform face recognition. All the features extracted and the face ids are put in a file “trainer.yml”. It is a .yml file that can be read by recognizer only. In this project, we’re using LBPH Face Recognizer for face identification.



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


path = "dataset"
```

```
>>> %Run training.py

 [INFO] Training faces. It may take few seconds please wai
t...
```

Fig. 10 Training Faces

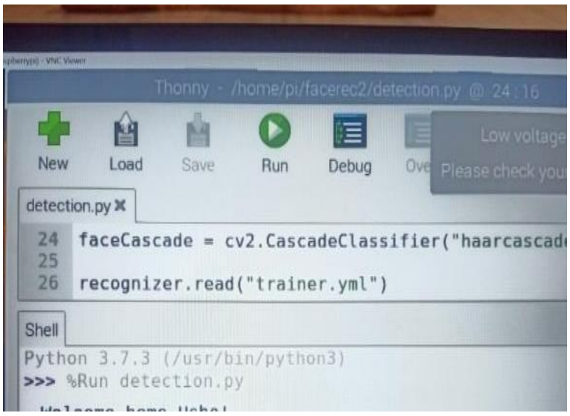


```
import cv2
import numpy as np
from PIL import Image
import os


path = "dataset"
```

```
 [INFO] Training faces. It may take few seconds please wai
t...

 [INFO] 3 faces trained. Exiting Program
>>>
```

Fig. 11 All faces in the collection are trained.

C. Face Recognition and Door Lock Operation This the main code of the project where the recognition of face and the operation of door lock is determined. When a person is standing in front of the door, the program is run, first it allows the camera to start capturing the visual in front it. The solenoid lock is kept closed until the face is recognized. If the face is recognized, the raspberry pi sends signal to solenoid lock through relay module which is opened then and then it closes off again after time lapse of 10 seconds. The time lapse value is manually set and can be changed whenever.



```
faceCascade = cv2.CascadeClassifier("haarcascad

recognizer.read("trainer.yml")
```

```
Python 3.7.3 (/usr/bin/python3)
>>> %Run detection.py
 Welcome home Usha!
```

Fig. 12 Output Window

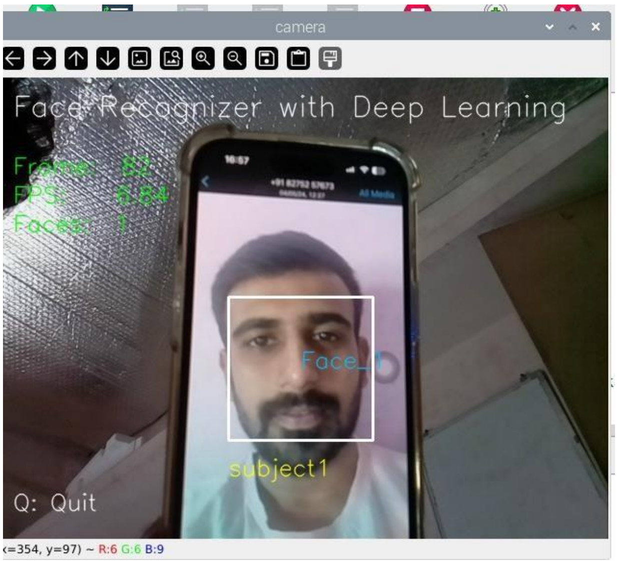

camera
Face Recognizer with Deep Learning
Frame: 82
FPS: 6.84
Faces: 1
Face_1
subject1
Q: Quit
(=354, y=97) ~ R:6 G:6 B:9

Fig. 13 Face Recognized

## IV. CONCLUSION

In this project, door locking system using face recognition process has been created using a Raspberry Pi, relay module, solenoid lock and a raspberry pi camera. We used the Haar Cascade classifier method to identify the face and Local Binary Pattern Histogram (LBPH) in recognizing the face. The face recognition process depends on various attributes like illumination, face position and so on, these problems have been handled to get better results, thereby giving an output with a confidence between 60 to 70 percentage. We decide that various operations are successfully tested and results are documented.

## ACKNOWLEDGEMENT

We would like to express our deep gratitude to **Prof.Mrs. M.V.Gangapure** & H.O.D **Prof. Y.M.Patil** for their constant encouragement and belief in us. Their guidance and attention throughout the projectwork has been of immensehelp to us.

We express our sincere thanks to all the teaching and non-teaching staff andall thosewhohave directly or indirectly helped in making projecta success.

Yours Sincerely,

Mr. Vishwajeet Patil Mr. Ritesh Dolare Mr. Atharva Ghatage

## REFERENCES

[1]. Multi-faces recognition process using Haar cascades and eigenface method T Mantoro, MA Ayu 2018

[2]. A Survey of Face Recognition Techniques Rabia Jafri* and Hamid R. Arabnia* Journal of Information Processing Systems, Vol.5, No.2

[3]. "Implementation of Human Face Detection System for Door Security using Raspberry Pi" 1. Shrutika V. Deshmukh, 2. Prof Dr. U. A. Kshirsagar, IJIREEICE, ISO 3297:2007 Certified, Vol. 5, Issue 4, April 2017.

[4]. "Home locking system through face recognition using Raspberry Pi and GSM module" 1. Ms. Poojitha S, 2. Ms. Yashodha S.A, 3. Ms. Priyadarshini S, Bangalore institute of technology, Bengaluru, India, Project reference no.: 40s_be_0592.

[5]. "Survey on real-time facial expression recognition techniques" 1. Shubhada Deshmukh, 2. Manasi Patwardhan, 3. Anjali Mahajan 1. PIET, VIT, Pune, India 2. Computer Engineering, VIT, Pune, India 3. PIET, Nagpur, India, ISSN 2047-4938.

[6]. "Face Recognition Using Haar - Cascade Classifier for Criminal Identification" 1. Senthamizh Selvi.R, 2. D.Sivakumar, 3. Sandhya.J.S , 4. Siva Sowmiya.S, Ramya.S , Kanaga Suba Raja.S International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878, Volume-7, Issue-6S5, April 2019.

[7]. Cuimei Li, Qi Zhiliang. "Human face detection algorithm via Haar cascade classifier with three additional classifiers", 13th IEEE International Conference on Electronic Measurement & Instruments, pp. 01-03, 2017.

[8]. S L Suma, Sarika Raga. "Real Time Face Recognition of Human Faces by using LBPH and Viola Jones Algorithm.", International Journal of Scientific Research in Computer Science and Engineering, Vol.6, Issue.5, pp.01- 03, Oct. 2018.

[9]. https://arsfutura.com/magazine/building-a-face-recognition_powered-door-lock/

[10]. https://research.vit.ac.in/publication/facial-recognition enabled-smart-door-unlock