MINI-PROJECT REPORT

ON

‘AUTOMATIC DOOR LOCK USING RASPBERRY PI MODULE’

SUBMITTED BY

3162 Shreeya Mahajan (C22020111163)

3168 Aishwarya Anil Menon (C22020111169)

3175 Saloni Khursade (C22021112107)

UNDER THE GUIDANCE OF

Dr. SUPRIYA MANGALE

DEPARTMENT OF

ELECTRONICS AND TELECOMMUNICATION ENGINEERING

MKSSS’s

Cummins College of Engineering for Women, Pune

(An Autonomous Institute Affiliated to Savitribai Phule Pune University)

(2021-2022)

MAHARSHI KARVE STREE SHIKSHAN SANSTHA'S

CUMMINS COLLEGE OF ENGINEERING FOR WOMEN

KARVE NAGAR, PUNE-411 052. (INDIA)

(An Autonomous Institute Affiliated to Savitribai Phule Pune University)

MKSSS's Cummins COLLEGE OF ENGINEERING FOR WOMEN

CERTIFICATE

This is to certify that the Mini Project work entitled

'AUTOMATIC SMART DOOR'

is a bonafide record of the project work carried out in this institute

By

SHREEYA MILIND MAHAJAN (C. No.: C22020111163)

AISHWARYA ANIL MENON (C. No.: C22020111169)

SALONI GANESH KHURSADE (C. No. : C22021112107)

in partial completion of the term work for the Third Year B.Tech.

in

Electronics and Telecommunication Engineering

in the academic year **2021-2022.**

This Mini-Project Report is a record of their own work carried out under our supervision and guidance.

CERTIFIED

Dr. Supriya. Mangale	**Dr. Prachi Mukherji**	**Dr.Madhuri.Khambete**
Internal Guide	**HoD (E&Tc)**	**Principal,CCOEW,Pune-52.**

ACKNOWLEDGEMENT

We would like to express our sincere gratitude towards our mini-project guide **Dr. Supriya Mangle** for her constant encouragement and valuable guidance during the completion of this mini-project work.

We would also like to thank **Dr.Prachi Mukherji (H.O.D.,E&Tc)** for her continuous valuable guidance, support, valuable suggestions and her precious time in every possible way in spite of her busy schedule throughout our project activity.

We take this opportunity to express our sincere thanks to all the staff members of E&Tc. Department for their constant help whenever required. Finally, we express our sincere thanks to all those who helped us directly or indirectly in many ways in the completion of this mini-project work.

Student Names

SHREEYA MILIND MAHAJAN (C. No.: C22020111163 **)**

AISHWARYA ANIL MENON (C. No.: C22020111169 **)**

SALONI GANESH KHURSADE (C. No. : 22021112107 **)**

AUTOMATIC SMART DOOR

TABLE OF CONTENTS

Sr.No	Contents	Page No.
1	ABSTRACT	1
2	INTRODUCTION	2
3	LITERATURE SURVEY	3
4	METHODOLOGY	4
5	BLOCK DIAGRAM	8
6	CIRCUIT DIAGRAM	11
7	ALGORITHM AND CODE	14
8	RESULTS	23
9	OBSERVATION	25
10	CONCLUSION	26
11	COST OF THE PROJECT	27
12	REFERENCES	28

ABSTRACT

The main objective of this project is to make an Automatic door lock which would be helpful for security of household. Normal door locks require manual help to be opened or closed, which is not possible for children or old people to do or when someone is not present at home. Automatic door lock requires a keypad to set the password and a camera to scan the face of the person standing at the door. Only by entering the correct password would the camera get triggered and hence whoever is standing at the door will get his face captured. A solenoid lock plays a very important role here in the opening and closing of the door. A solenoid door lock is a type of electromagnetic lock that is used to secure doors. Solenoid door locks work by using an electromagnet to create a magnetic field that holds the door closed. When the power is turned off, the electromagnet is deactivated, and the door can be opened. This makes the opening and closing of door simple.

INTRODUCTION

Now a days, this word is becoming more digital, people wish systems which require less efforts and more output. Digital technologies made our lives simpler than ever before. A normal door lock with a metal lock and keys sounds very old and boring, carrying the keys everywhere you go is very hectic, one has to be very careful while carrying the keys, what if the keys are left somewhere and you are unable to get those? Isn't it scary imagining some stranger got those keys and tries to open the lock when you are away from your home. Password and camera-based door lock can solve this problem real quick. Only the owner would have the password to the lock and only he can access it, by entering the password camera gets triggered and hence by scanning the correct face the lock to the door opens. Simple to implement and simplest to use. Conventional doors are among the top choices for homeowners because they have aesthetic value. When installed at home or a business, they can blend in the exterior and indoor style of any property, adding and enhancing its curb appeal. Conventional doors are known to be energy efficient. If you have a pile of electric bills on your table, investing in these doors may help. But using your appliances responsibly is imperative. Also, proper insulation is another thing you cannot overlook.

Benefits:

1. Less Time-Consuming : While using a simple lock and key system, a lot of time is consumed in taking out the keys from your bag, inserting it inside the cavity of the lock, and then unlocking the door. However, using the face recognition feature offered by digital locks takes no more than a fraction of a second to open the door. The user is thus facilitated with a seamless experience.

2. Say Goodbye to Physical Keys : A key is an important object that is required to be handled with care and losing it becomes a matter of concern. Moreover, creating and distributing multiple copies of the key among your family members adds to the worry of misplacing them. Using a face recognition system, you and your entire family will be able to unlock and enter the door conveniently.

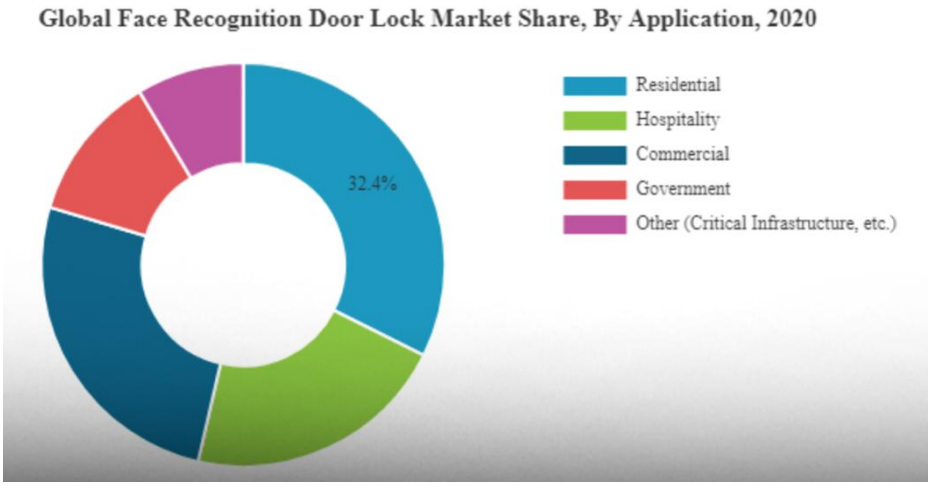
3. No Contact Needed : During the ongoing COVID pandemic, we need to protect ourselves by avoiding excessive contact with surfaces and objects, as this may lead to the possibility of contracting the virus. Thus, installing digital door locks with a face recognition feature in your home or office can ensure a more safe and sanitary experience.

4. Highly Secure : While a pin or password could be misused by someone who may have seen it, face recognition is a powerful feature that accurately identifies the person's face and their expression to unlock the door. The face recognition system using Haar classifier can scan over 170 facial points. This ensures that your home is completely secure. Moreover, it has the ability to distinguish between a real human face and an image or a video.

5. Old Age : Although fingerprints do not change with age, it can be more difficult to capture them in older people. This is because the skin loses elasticity with age, and the patterns become less prominent, especially due to the thickening of ridges and furrows. Facial recognition could be better in this case.

Applications : The application market of this security system is mostly hospitality, residential, and governmental organizations. Face recognition front door lock makes **high-**

security to reduce the risks of security problems in urban areas. The demand for a high-security system is more wanted through governmental organizations. They must offer a



steady *facial recognition door lock* to increase the surveillance of their entry or information.

Residential : For the past few years, the number of homeowners who choose and install automatic doors has been on the rise. No wonder because state-of-the-art doors are a lasting and worthwhile investment.

Safety : Burglary is a major public concern worldwide. In America, break-ins are a few of the property crime problems, according to a 2020 survey. On average, intrusion takes place every 26 seconds in the country, according to the FBI.
How to increase safety in your home? More than the security systems, using automatic doors may help. Whether you are at home or will go out of town for business purposes, these security units can shy away/deter burglars effectively, giving you peace of mind.

Convenience : Let's admit it! Rotating door knobs, pressing on door handles, closing doors, and pulling/pushing doors can be inconvenient and challenging, especially when you hold some of your personal items. Say bye to the hassles and other discomforts with automatic doors. There's no need to rotate door knobs or press handles, making your life simpler and less stressful than ever.

Commercial : Automatic doors are also common in commercial establishments, ranging from restaurants, hospitals to shopping malls. Why do startups and well-established companies prefer automatic doors? Some of the reasons are highlighted below:
Protection from an Unauthorized Person
Emergency Situations
Elderly Access

LITERATURE SURVEY

.

83 Smart Locking Systems for Homes

A door which serves as an entrance to the home or office isn't just about keeping the bad guys out, it's also about letting the right people in, like family, friends. With the Smart door access system, you can lock or unlock the door from your smartphone over the internet or Bluetooth and allow entry for your family and friends. An android application will be used to interface the door locking system, making it easier for the people to control the very door itself remotely. Facial Recognition is also used for making it easier to unlock the door, by just looking into camera one can unlock door, given, his image is present in the database. Smartphone's application-based control will require an internet connection.

Automatic Door Locking System

This paper consists of a project which gives service at low cost compared to the cost of the available security system. We want to make a system that will give 24 to 7 service by using registered password in this system we can unlock the door by which it increases the security level to prevent an unauthorized unlocking. If the user forgets the combination of passwords this system gives the flexibility to the user to change or reset the password. Security measures are very high as provided in two ways. First, we have to enter the password for blue-tooth connection and second is for unlocking the door in application. Both passwords can be changed as and when required. This automatic password-based lock system will give the user a more secure and low-cost way of locking-unlocking system.

Smart digital door lock for home automation

In this paper, we propose a smart digital door lock system for home automation. A digital door lock system is equipment that uses digital information such as secret codes, semi-conductors, smart cards, and fingerprints as the method for authentication instead of the legacy key system. In our proposed system, a ZigBee module is embedded in digital door lock and the door lock acts as a central main controller of the overall home automation system. Technically, our proposed system is a network of sensor nodes and actuators with digital door lock as base station. A door lock system proposed here consists of RFID reader for user authentication, touch LCD, motor module for opening and closing of the door, sensor modules for detecting the condition inside the house, communication module, and control module for controlling other modules. Sensor nodes for environment sensing are deployed at appropriate places at home. The status of the individual ZigBee module can be monitored and controlled by the centralized controller, digital door lock.

METHODOLOGY

Automatic door lock uses solenoid lock to follow open and close operation. It also provides home security and an emergency system to be activated. Automatic door lock not only refers to reducing human efforts but also is energy efficient and time saving. The main objective of Automatic door lock is to help handicapped, old aged people and children who are unable to operate door manually. Keypad and camera are externally interfaced with raspberry pi module, once correct password is entered the camera gets triggered and the person standing at door will get access to the door. The Python library that we have used in this project is the face_recognition module which uses HOG and SVM classifiers at the back end.

face_recognition library

The heart of the project is the face recognition library. This library is built using dlib's state-of-the-art face recognition built with deep learning. The model has an accuracy of 99.38% on the Labelled Faces in the Wild benchmark.This library can do following operations:

This library can do following operations:

1. Find all the faces that appear in a picture

2. Find and manipulate facial features in pictures. Get the locations and outlines of each person's eyes, nose, mouth and chin.

3. Identify faces in pictures

At the back end of this library HOG (Histogram of oriented gradients) algorithm is used along with the SVM (Support Vector Machine) classifier.

In the early 2000's Paul Viola and Michael Jones invented a way to detect faces that was fast enough to run on cheap cameras. The **Viola- Jones algorithm** has four stages:

1. Haar Feature Selection: The eye region is darker than the upper-cheeks and the nose bridge region is brighter than the eyes like that 4 features are extracted.

2. Creating an Integral Image

3. Adaboost Training

4. Cascading Classifiers

:In short, it generates a large set of features and uses the boosting algorithm AdaBoost to reduce the overcomplete set and the introduction of a degenerative tree of the boosted classifiers provides for robust and fast interferences. The detector is applied in a scanning fashion and used on greyscale images, the scanned window that is applied can also be scaled, as well as the features evaluated. But a much more reliable solution is HOG.

1. **HOG:**

Step by step:

1. Image is converted into black and white.

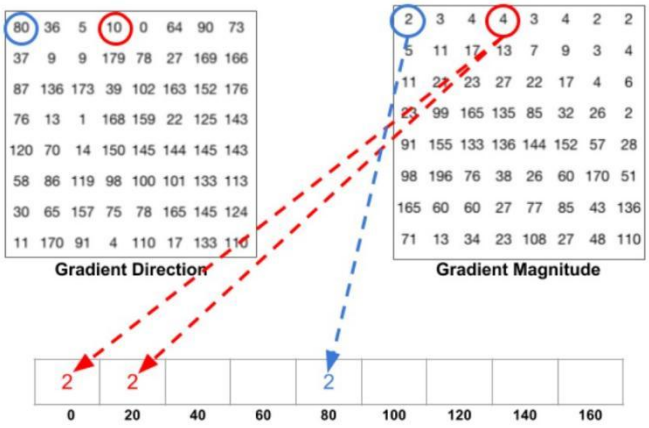
2. One pixel is taken at a time.

3. Considering the surrounding pixels, an arrow is drawn showing in which direction the image is getting darker. These arrows are called gradients. The gradient of the image is calculated. The gradient is obtained by combining magnitude and angle from the image. Considering a block of 3x3 pixels, first Gx and Gy is calculated for each pixel.

$$G_x(r,c) = I(r,c+1) - I(r,c-1) \quad G_y(r,c) = I(r-1,c) - I(r+1,c)$$

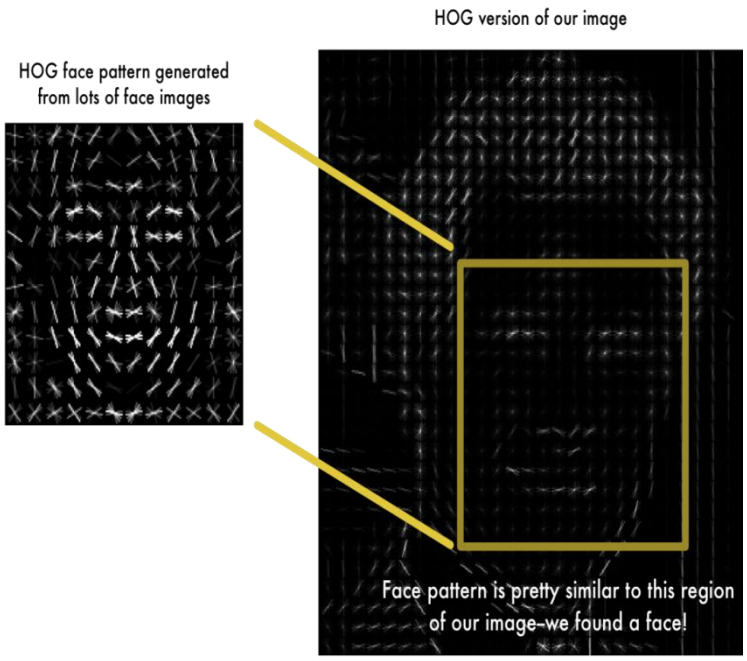
where r, c refer to rows and columns respectively. (Image by author)

4. After calculating Gx and magnitude and angle of each pixel is calculated. The image is divided into 8×8 cell blocks and a histogram of gradients is calculated for each 8×8 cell block. The histogram is essentially a vector of 9 buckets (numbers) corresponding to angles from 0 to 180 degree (20 degree increments). The values of these 64 cells (8X8) are binned and cumulatively added into these 9 buckets.This essentially reduces 64 values into 9 values.



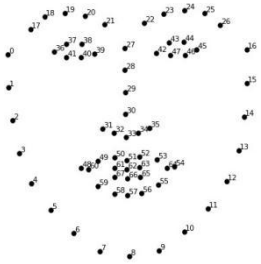
5. To calculate the final feature vector for the entire image patch, the 36×1 vectors are concatenated into one giant vector. So, say if there was an input picture of size 64×64 then the 16×16 block has 7 positions horizontally and 7 positions vertically.

6. In one 16×16 block we have 4 histograms which after normalization concatenate to form a 36×1 vector. This block moves positions horizontally and vertically totalling it to 7×7 = 49 positions. So when we concatenate them all into one giant vector we obtain a 36×49 = 1764 dimensional vector.This vector is now used to train classifiers such as SVM and then do object detection.



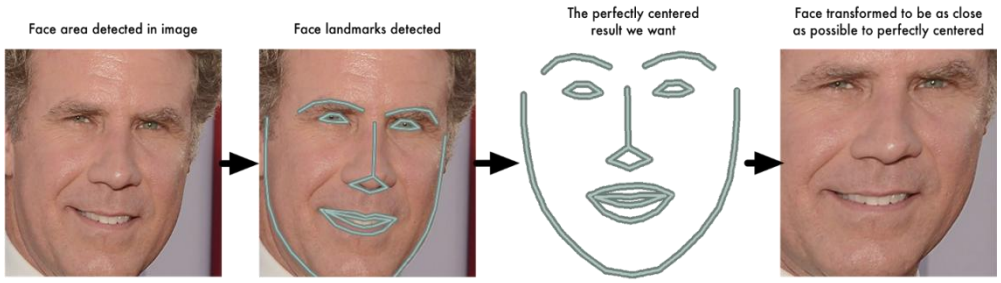
2. Further to find faces in this HOG image, all we have to do is find the part of our image that looks the most similar to a known HOG pattern that was extracted from a bunch of other training faces. From this algorithm we can find faces in any image using python and dlib library.

3. Face landmark estimation algorithm and Josephine Sullivan is used. In this algorithm landmarks which exist on every face so with the the code to find these landmarks on every face.



invented by Vahid Kazemi they have decided 68 help of ML they will train

4. After detecting the landmarks of the face, the image can be rotated, sheared, scaled to bring the eyes, mouth at the centre of the image as possible.



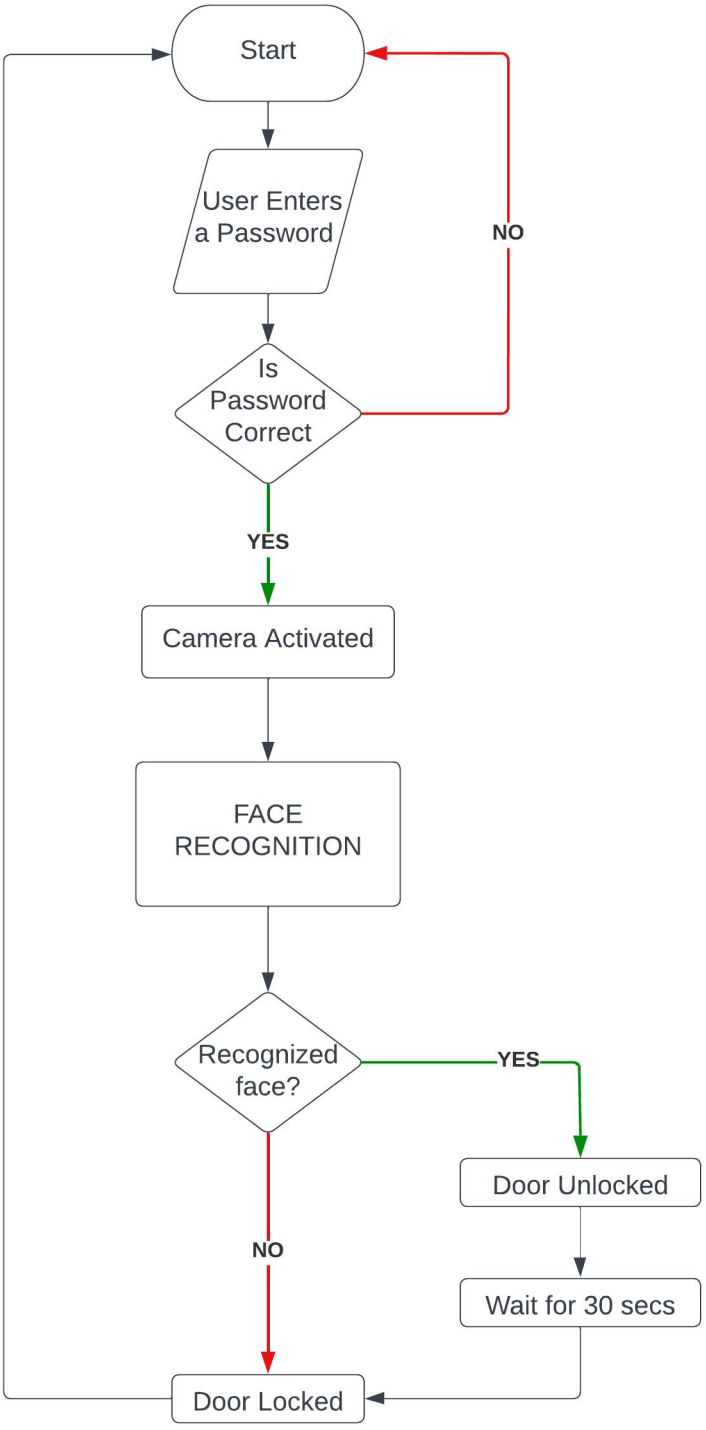
5. To train a Deep Convolutional Neural Network: Instead of training the network to recognize pictures objects, we are going to train it to generate 128 measurements for each face. The training process works by looking at 3 face images at a time:

1. Load a training face image of a known person.
2. Load another picture of the same known person.
3. Load a picture of a totally different person.

Then the algorithm looks at the measurements it is currently generating for each of those three images. It then tweaks the neural network slightly so that it makes sure the measurements it generates for #1 and #2 are slightly closer while making sure the measurements for #2 and #3 are slightly further apart. As we are using a face-recognition library, it already contains the trained neural network so we don't have to train the neural networks from scratch.

6. Recognising the face: Find the person in our database of known people who has the closest measurements to our test image. Train a classifier that can take in the measurements from a new test image and tell which known person is the closest match.

BLOCK DIAGRAM



Enter Password:

The Camera is mounted at a distance from the entrance to capture the frontal images of the students. The captured image is preferred to be of the size 640x480 to avoid resizing of the image in the back-end as we observed resizing may sometimes result in poor performance.

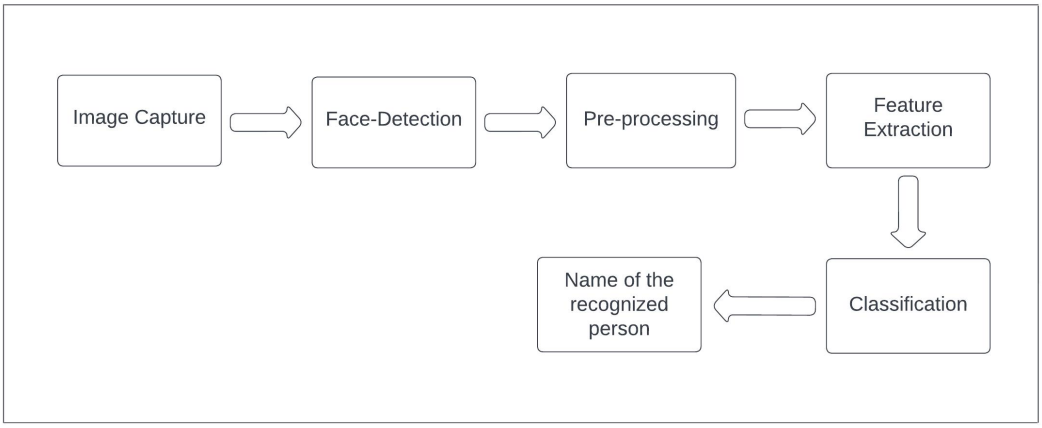
Check the Password:

Once the password is entered , it checks for the accuracy . The default password which we have assigned in the code is 1234 so if the same has been entered then and then only the password will get accepted, unless and until the password is entered correctly the camera would not get triggered .

Camera Activated:

The correct password will trigger the camera to open and to detect the face , if detected face matches the image in the dataset then the door will get unlock , if not then all the steps are to be repeated from the beginning.

Face Recognition:



FACE RECOGNITION BLOCK

Image Capture:

The Camera is mounted at a distance from the entrance to capture the frontal images of the students. The captured image is preferred to be of the size 640x480 to avoid resizing of the image in the back-end as we observed resizing may sometimes result in poor performance.

Face Detection:

A proper and efficient face detection algorithm always enhances the performance of face recognition systems.

We have used a **face_recognition library** for this. It has many methods (functions) to deal with faces in images and one of them is known as face_locations that will find the face's

locations inside a particular image and we will pass the image file that we've used in the previous line of code.

Pre-processing:

The clicked image is first converted from BGR to RGB using “BGR2RGB”.

Database Development:

As we chose biometric based system enrolment of every individual is required. This database development phase consists of image capture of every individual and extracting the biometric feature, in our case it is face, and later it is enhanced using pre-processing techniques and stored in the database. Here we have added images of our batchmates.

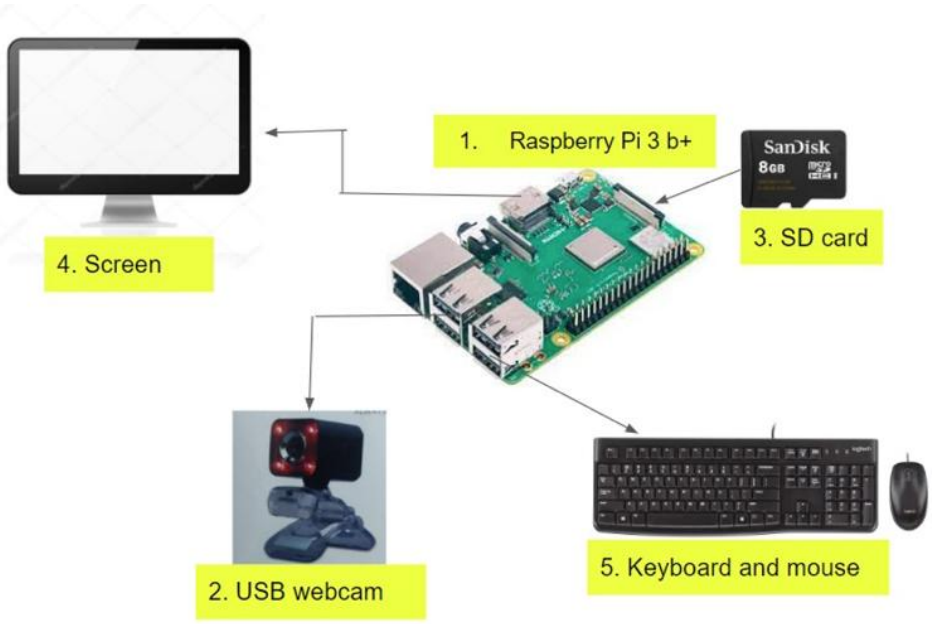
Feature Extraction and Classification:

The classifier used here is SVM(Support Vector Machine).It is a supervised machine learning algorithm used for both classification and regression. The advantage of support vector machines is that it is effective in high dimensional spaces like cases where the number of dimensions is greater than the number of samples.

Post-processing:

In the proposed system, after recognizing the faces of the students, the names are updated into the csv file.

CIRCUIT DIAGRAM



1. Raspberry Pi 3 b+ module: Heart of the system. The Raspberry Pi 3 Model B is the third generation Raspberry Pi. This powerful credit-card sized single board computer can be used for many applications and supersedes the original Raspberry Pi Model B+ and Raspberry Pi 2 Model B. This model maintains the popular board format and brings you a more powerful processor, 10x faster than the first-generation Raspberry Pi. Additionally, it adds wireless LAN & Bluetooth connectivity making it the ideal solution for powerful connected designs.

Specifications:

Processor: Broadcom BCM2387 chipset. 1.2GHz Quad-Core ARM Cortex-A53 802.11 b/g/n Wireless LAN and Bluetooth 4.1

GPU: Dual Core VideoCore IV® Multimedia Co-Processor

Memory: 1GB LPDDR2

Access: Extended 40-pin GPIO header

Dimensions: 85 x 56 x 17mm

Power: Micro USB socket 5V1, 2.5A

2. USB Webcam:

Camera is connected to the USB port of Raspberry Pi

Product: Zeb-Crystal Pro USB Powered Webcam

Manufacturer: Zebronics

Feature: 3P high quality lens

Video resolution: 640 x 480 (30 FPS)

3. SD Card: Inserted into the SD card slot of the Raspberry Pi. The SD card will contain the OS of Raspberry Pi, codes and all the data. 32 GB SD card is required for Raspberry Pi.

4. Screen:Connected to Raspberry Pi through HDMI cable. It displays the operations going on in Raspberry Pi and acts as a user interface.

5. Keyboard and mouse: Connected to Raspberry Pi at the USB ports. Helps in operating Raspberry Pi.

6. Relay Motor:A power relay module is an electrical switch that is operated by an electromagnet. The electromagnet is activated by a separate low-power signal from a micro controller. When activated, the electromagnet pulls to either open or close an electrical circuit. Similarly, we have used relay module to open or close the camera if face is detected.

Specifications of Relay - Breakdown Voltage 300 V peak

Carry Current 1.0 Amps

Switch Speed 1 mS

Life Expectancy (mechanical) > 100 million

7. Solenoid Lock:12V Solenoid lock has a slug with a slanted cut and a good mounting bracket. It's basically an electronic lock, designed for a basic cabinet, safe or door. When 9-12VDC is applied, the slug pulls in so it doesn't stick out and the door can be opened. It does not use any power in this state. It is very easy to install for automatic door lock systems like electric door lock with the mounting board. This solenoid in particular is nice and strong.

Specifications of 12V solenoid lock:

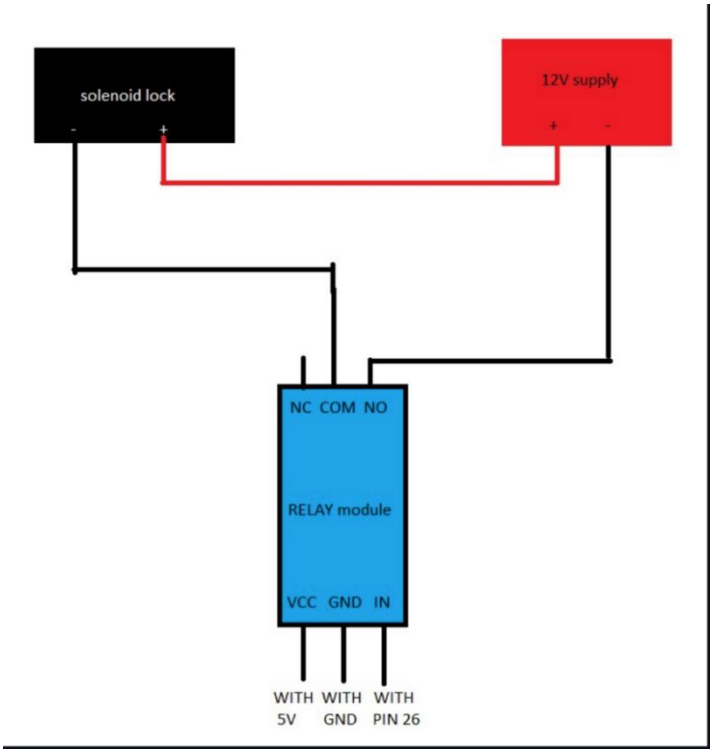
Operating voltage: 12VDC

Draws 650mA at 12V, 500 mA at 9V when activated

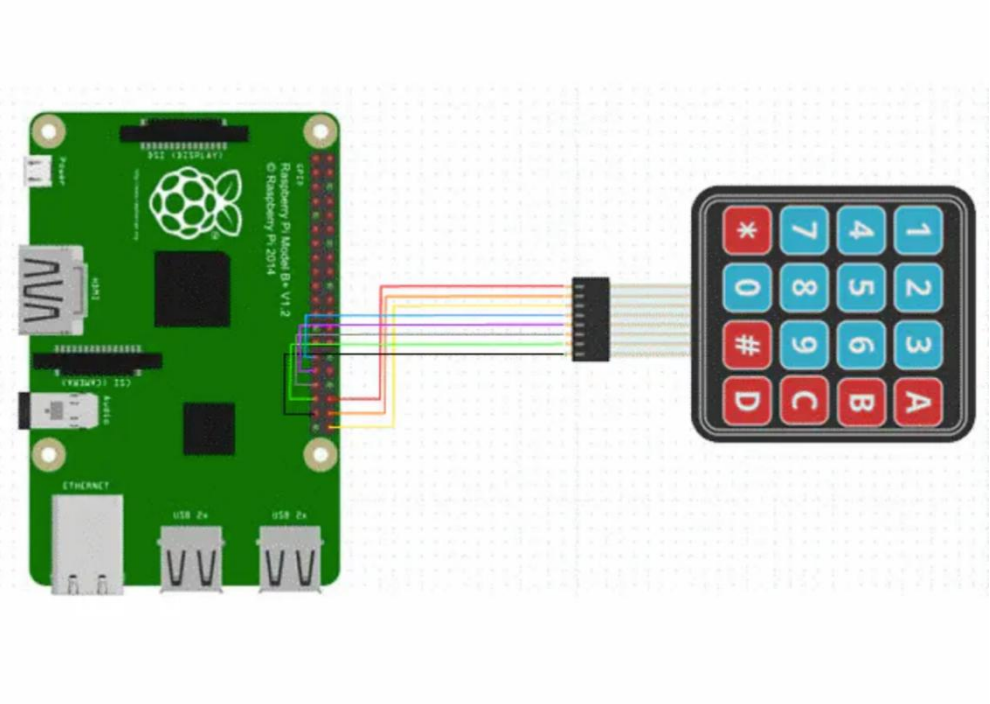
Designed for 1-10 seconds long activation time

Wire length: 222.25mm

8. 12V Supply: These power supplies work by regulating the output voltage with a high-frequency switching process utilizing feedback with pulse width modulation. Moreover, switching regulated power supplies run substantial EMI filtering to reduce the load and line's regular and differential noise.



9. 4x4 Keypad: Heart of the system. The Raspberry Pi 3 Model B is the third generation Raspberry Pi. This powerful credit-card sized single board computer can be used for many applications and supersedes the original Raspberry Pi.



ALGORITHM

1. START
2. Import all the required libraries and files.
3. Declare all the BCM pins for each connection.
4. Make a function which takes input passkey from the user and checks whether it matches with the secret passkey. Its return type should be boolean.
5. If the password is not matching, then start again from the top.
6. If the password is correct, then the program automatically switches on the webcam and a window, with the camera on, pops up.
7. If the user standing in front of the camera is a known person then a box should appear around their face with the name of the user.
8. Otherwise, the user must be named as unknown.
9. If the person is declared as unknown then the solenoid lock should not open.
10. If the person is a known person, it checks if the person has the access to open the door. If not then the lock should still stay closed.
11. When the known person has been given the access in the program, then the lock must open.
12. After waiting for 30 secs, the door should be locked again.
13. END

CODE

main.py

```
from imutils.video import VideoStream
from imutils.video import FPS
import imutils
import pickle
import time
import os
import password
import RPi.GPIO as GPIO

import face_recognition
import cv2
import numpy as np

pin=26

GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)
GPIO.setup(pin,GPIO.OUT)
GPIO.output(pin, GPIO.HIGH)



def unlock():
    '''
    pin: pin which you attach on your pi with solenoid lock, default is 26
    '''
    GPIO.setmode(GPIO.BCM)
    GPIO.setup(pin,GPIO.OUT)
    GPIO.output(pin,GPIO.LOW)

def lock():
    '''
    pin: pin which you attach on your pi with solenoid lock, default is 26
    '''
    GPIO.setmode(GPIO.BCM)
    GPIO.setup(pin,GPIO.OUT)
    GPIO.output(pin,GPIO.HIGH)

# This is a demo of running face recognition on live video from your webcam. It's a little more complicated
than the
# other example, but it includes some basic performance tweaks to make things run a lot faster:
#   1. Process each video frame at 1/4 resolution (though still display it at full resolution)
```

```
#   2. Only detect faces in every other frame of video.

# PLEASE NOTE: This example requires OpenCV (the `cv2` library) to be installed only to read from your webcam.
# OpenCV is not required to use the face_recognition library. It's only required if you want to run this
# specific demo. If you have trouble installing it, try any of the other demos that don't require it instead.

# Get a reference to webcam #0 (the default one)

# unlock()
# time.sleep(2)
# print("U")
# lock()
# time.sleep(2)
# print("L")

video_capture = cv2.VideoCapture(0)

# Load a sample picture and learn how to recognize it.
shreeya_image = face_recognition.load_image_file("3.jpeg")
shreeya_face_encoding = face_recognition.face_encodings(shreeya_image)[0]

# Load a second sample picture and learn how to recognize it.
aishwarya_image = face_recognition.load_image_file("2.jpeg")
aishwarya_face_encoding = face_recognition.face_encodings(aishwarya_image)[0]

# Load a second sample picture and learn how to recognize it.
saloni_image = face_recognition.load_image_file("1.jpeg")
saloni_face_encoding = face_recognition.face_encodings(saloni_image)[0]

# Create arrays of known face encodings and their names
known_face_encodings = [
    shreeya_face_encoding,
    aishwarya_face_encoding,
    saloni_face_encoding
]
known_face_names = [
    "Shreeya",
    "Aishwarya",
    "Saloni"
]

# Initialize some variables
face_locations = []
face_encodings = []
```

```
face_names = []
process_this_frame = True

while True:
    try:
        password.init()
        matched = False
    except KeyboardInterrupt:
        break
    if(not password.check_passcd()):
        print ("Password is incorrect")
    else:
        tic = time.time()
        while True:
            # Grab a single frame of video
            ret, frame = video_capture.read()

            # Only process every other frame of video to save time
            if process_this_frame:
                # Resize frame of video to 1/4 size for faster face recognition processing
                small_frame = cv2.resize(frame, (0, 0), fx=0.25, fy=0.25)

                # Convert the image from BGR color (which OpenCV uses) to RGB color (which
face_recognition uses)
                rgb_small_frame = small_frame[:, :, ::-1]

                # Find all the faces and face encodings in the current frame of video
                face_locations = face_recognition.face_locations(rgb_small_frame)
                face_encodings = face_recognition.face_encodings(rgb_small_frame, face_locations)

                face_names = []
                for face_encoding in face_encodings:
                    # See if the face is a match for the known face(s)
                    matches = face_recognition.compare_faces(known_face_encodings, face_encoding)
                    name = "Unknown"

                    # # If a match was found in known_face_encodings, just use the first one.
                    if True in matches:

                        first_match_index = matches.index(True)
                        name = known_face_names[first_match_index]

                        if name == 'Shreeya':
                            print("Door is Unlocked")
                            unlock()
```

```
            print("Waiting for 15 seconds")
            time.sleep(15) #Lock will remains open for 10 seconds.
            print("Door is Locked")
            lock()
            GPIO.cleanup()


        # Or instead, use the known face with the smallest distance to the new face
        face_distances = face_recognition.face_distance(known_face_encodings, face_encoding)
        best_match_index = np.argmin(face_distances)
        if matches[best_match_index]:
            name = known_face_names[best_match_index]


        face_names.append(name)

process_this_frame = not process_this_frame


# Display the results
for (top, right, bottom, left), name in zip(face_locations, face_names):
    # Scale back up face locations since the frame we detected in was scaled to 1/4 size
    top *= 4
    right *= 4
    bottom *= 4
    left *= 4

    # Draw a box around the face
    cv2.rectangle(frame, (left, top), (right, bottom), (0, 0, 255), 2)

    # Draw a label with a name below the face
    cv2.rectangle(frame, (left, bottom - 35), (right, bottom), (0, 0, 255), cv2.FILLED)
    font = cv2.FONT_HERSHEY_DUPLEX
    cv2.putText(frame, name, (left + 6, bottom - 6), font, 1.0, (255, 255, 255), 1)

tok = time.time()
# Display the resulting image
cv2.imshow('Video', frame)

# Hit 'q' on the keyboard to quit!
if cv2.waitKey(1) & 0xFF == ord('q'):
    video_capture.release()
    cv2.destroyAllWindows()
    break
# Release handle to the webcam
```

```
        if tok-tic > 30:
            video_capture.release()
            cv2.destroyAllWindows()
            Break
```

password.py

```
''' 4-digit passcode'''
import RPi.GPIO as GPIO
import time
import os

import config1

#star_list=[]
pos=[(100,160),(140,160),(180,160),(220,160)]

Matrix = [['1','2','3'],['4','5','6'],['7','8','9'],['*','0','#']]

Row = [18,23,24,21]
Col = [5,6,13]

def check_passcd():
    GPIO.setmode(GPIO.BCM)
    GPIO.setwarnings(False)
    for j in range(3):
        GPIO.setup(Col[j],GPIO.OUT)
        GPIO.output(Col[j],1)
    for i in range(4):
        GPIO.setup(Row[i],GPIO.IN,pull_up_down = GPIO.PUD_UP)
    enter = []
    star_list = []
    while True:
        time.sleep(0.2)
        for j in range(3):
            GPIO.output(Col[j],0)
            for i in range(4):
                if (not GPIO.input(Row[i])):
                    if (Matrix[i][j] != '#'):
                        enter.append(Matrix[i][j])
                        if len(star_list) == 0:
                            star_list.append(Matrix[i][j])
                        else:
                            star_list[len(star_list)-1] = '*'
                            star_list.append(Matrix[i][j])
                        print(star_list)
```

```
                else:
                    print ("enter")
                    print(enter)
                    passcode = config1.reset()
                    print ("passcode")
                    print(passcode)
                    print ("Passcode matched?")
                    print(enter == passcode)
                    return(enter == passcode)

                while(not GPIO.input(Row[i])):
                    pass

            GPIO.output(Col[j],1)

    return False
#return(enter == config.PASSCODE)

def init():
    init_passcd = [1,2,3,4]
    file = open(config1.PASSCODE_FILE,'w')
    file.writelines(["%s\n" % digit for digit in init_passcd])
    file.close()

def reset():
    GPIO.setmode(GPIO.BCM)
    GPIO.setwarnings(False)
    for j in range(3):
        GPIO.setup(Col[j],GPIO.OUT)
        GPIO.output(Col[j],1)
    for i in range(4):
        GPIO.setup(Row[i],GPIO.IN,pull_up_down = GPIO.PUD_UP)
    new_passcd = []
    star_list = []

    while True:
        time.sleep(0.2)
        for j in range(3):
            GPIO.output(Col[j],0)
            for i in range(4):
                if (not GPIO.input(Row[i])):
                    if (Matrix[i][j] != '#'):
                        new_passcd.append(Matrix[i][j])
                        if len(star_list) == 0:
                            star_list.append(Matrix[i][j])
```

```
                else:
                    star_list[len(star_list)-1] = '*'
                    star_list.append(Matrix[i][j])
            else:
                #print(enter)
                print("New Passcode")
                print(new_passcd)
                file = open(config1.PASSCODE_FILE,'w')
                file.writelines(["%s\n" % digit for digit in new_passcd])
                file.close()
                #print(config.PASSCODE)
                return

            while(not GPIO.input(Row[i])):
                pass

        GPIO.output(Col[j],1)
```

config1.py

```
PASSCODE_FILE = 'passcode.txt'

def reset():
PASSCODE = open(PASSCODE_FILE,'r').read().splitlines()
return PASSCODE
```

passcode.txt

```
1
2
3
4
```

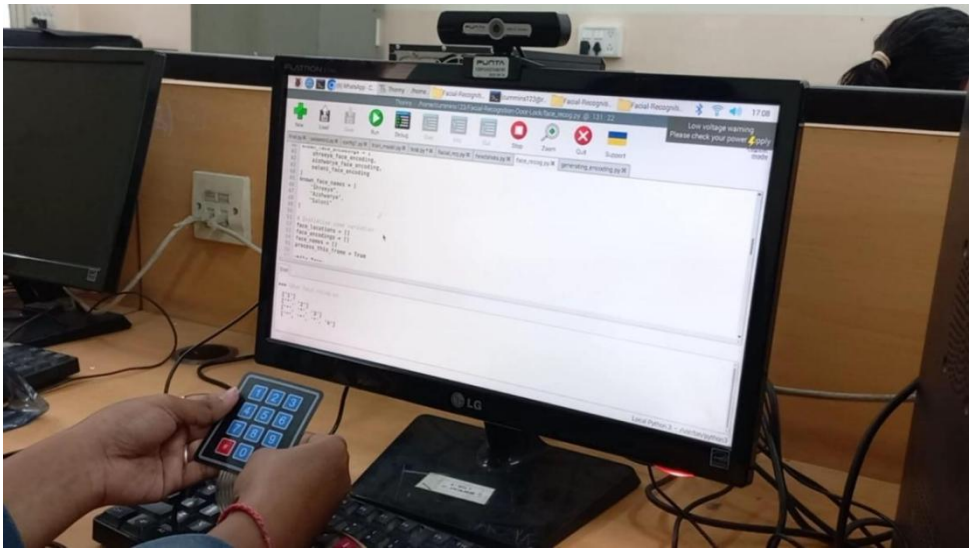
RESULTS

CONNECTIONS:

USER ENTERING THE PASSWORD:

Secret Passkey: 1234

Passkey user entered: 1234



SOLENOID LOCK RESPONSE:

Passkey matched: TRUE

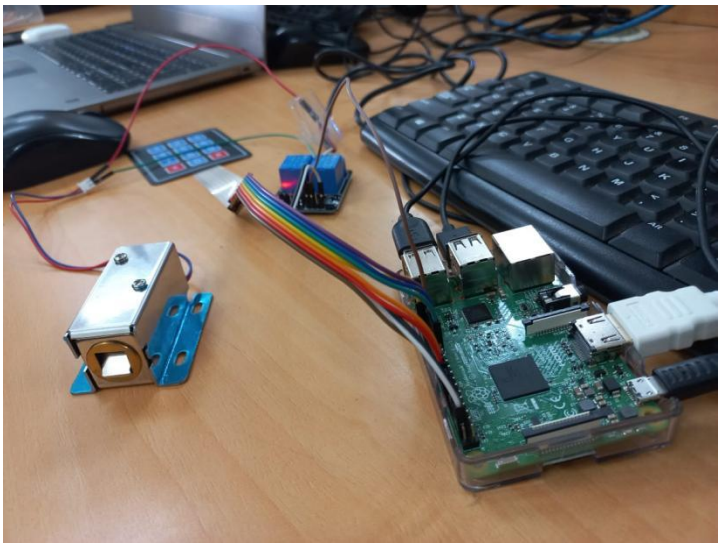
Solenoid lock: OPEN

Delay: 30 secs

Solenoid lock: CLOSED

Passkey matched: TRUE

Solenoid lock: CLOSED



OBSERVATIONS

After entering the correct password, the camera got triggered and the face got scanned which in turn opened the lock of the door. The camera will trigger only if correct password is entered.

CONCLUSION

To deal with real life problems mentioned above, automatic door lock works really great by reducing human efforts, it is a good source of security. Automatic door lock system requires less time as compared to manual lock, also overcomes the fear of losing the keys or some stranger taking it away and unlocking the door. Even some companies, banks have installed Automatic door-lock system.

COST OF PROJECT

Sr. No.	Component	Quantity	Price
1.	Raspberry Pi 3 B+	1	Rs. 4500
2.	USB Webcam	1	Rs. 699
3.	SD card (32 GB)	1	Rs. 500
4.	12V Adapter	1	Rs. 200
5.	4x4 Keypad	1	Rs. 90
6.	Relay Board	1	Rs. 69
7.	Solenoid Lock	1	Rs. 309
8.	Connecting Wires	30	Rs. 60
8.	Screen (optional)	1	Rs. 4000
9.	Keyboard and mouse(optional)	1	Rs. 400
Total Cost: Rs. 6,427 (optional ones not included)			

REFERENCES

1. Wikipedia
2. “Automatic door locking system” Andrew Amair
3. “Smart locking systems for home” Utkarsh Sundaram
4. Rapid Object Detection using a Boosted Cascade of Simple Features