INTERNSHIP REPORT

TITLE: Person of interest Identification & Notification from Live Video Stream Using Facial Recognition.

SUBMITTED BY:

B.SAI RAM VARMA

1215316310

M.ATAL BHUPATHI VARMA

1215316330

UNDER THE GUIDANCE OF:

Mr. Raghu Ram Reddy Tera

Chairman &Managing Director,

CASSINI SYSTEMS.

www.cassinisys.com

INDEX:

TITLE:

Person of interest Identification & Notification from Live Video Stream Using Facial Recognition.

ABSTRACT:

Develop a cloud-based Person of Interest Identification System from live video feed in public places using facial recognition. There will be four components in this project: a database to store the persons of interest, facial recognition service, web interface and notification service,The purpose of this system is to notify someone when a person of interest is identified from a video feed. The main application for such a system will be safety and security in public places like airports, malls, etc. where video feeds from various cameras are analyzed in real-time and matched with the persons of interest in the database and concerned personnel are notified.

Technologies C/C++, Python, Django, OpenCV, Facial Recognition, Image and Video Processing.

ACKNOWLEDGEMENT

On the submission of our internship report on “Human Face Detection and Recognition” we would like to extend our gratitude and sincere thanks to Mr. Raghu Ram Reddy Tera ,Chairman and Managing Director Cassini systems for his constant motivation and support during the course of our internship. We truly appreciate and value his esteemed guidance and encouragement from the beginning to the end of this project. We are indebted to him for having helped us shape the problem and providing insights towards the solution.

We want to thank our internship guide Mr. Praveen Siddhartha for providing a solid background for our project. They have been great sources of inspiration to us and we thank them from the bottom of our heart.

B.SAI RAM VARMA	M. ATAL BHUPATHI VARMA
ROLL NO:1215316310	ROLL NO:1215316330

CHAPTER 1: INTRODUCTION

INTRODUCTION:

The face is our primary focus of attention in social life playing an important role in conveying identity and emotions. We can recognize a number of faces learned throughout our lifespan and identify faces at a glance even after years of separation. This skill is quite robust despite of large variations in visual stimulus due to changing condition, aging and distractions such as beard, glasses or changes in hairstyle.

Computational models of face recognition are interesting because they can contribute not only to theoretical knowledge but also to practical applications. Computers that detect and recognize faces could be applied to a wide variety of tasks including criminal identification, security system, image and film processing, identity verification, tagging purposes and human-computer interaction. Unfortunately, developing a computational model of face detection and recognition is quite difficult because faces are complex, multidimensional and meaningful visual stimuli.

Face detection is used in many places now a days especially the websites hosting images like picassa, photobucket and facebook. The automatically tagging feature adds a new dimension to sharing pictures among the people who are in the picture and also gives the idea to other people about who the person is in the image. In our project, we have studied and implemented a pretty simple but very effective face detection algorithm which takes human skin color into account.

Our aim, which we believe we have reached, was to develop a method of face recognition that is fast, robust, reasonably simple and accurate with a relatively simple and easy to understand algorithms and techniques.

CHAPTER 2: FACE RECOGNITION vs BIOMETRICS

The information age is quickly revolutionizing the way transactions are completed. Everyday actions are increasingly being handled electronically, instead of with pencil and paper or face to face. This growth in electronic transactions has resulted in a greater demand for fast and accurate user identification and authentication. Access codes for buildings, banks accounts and computer systems often use PIN's for

identification and security clearance.

Using the proper PIN gains access, but the user of the PIN is not verified. When credit and ATM cards are lost or stolen, an unauthorized user can often come up with the correct personal codes. Despite warning, many people continue to choose easily guessed PIN‟s and passwords: birthdays, phone numbers and social security numbers. Recent cases of identity theft have highlighted the need for methods to prove that someone is truly who he/she claims to be.

Face recognition technology may solve this problem since a face is undeniably connected to its owner expect in the case of identical twins. Its nontransferable. The system can then compare scans to records stored in a central or local database or even on a smart card.

Each individual face can be represented exactly as the linear combination of “eigenfaces” or each face can also be approximated using those significant eigenfaces obtained using the most significant eigen values.

What are BIOMETRICS?

A biometric is a unique, measurable characteristic of a human being that can be used to automatically recognize an individual or verify an individual's identity. Biometrics can measure both physiological and behavioral characteristics. Physiological biometrics (based on measurements and data derived from direct measurement of a part of the human body) include:

1.FACE RECOGNITION

2.FINGER SCAN

3.IRIS SCAN

4.RETINA SCAN

5.HAND SCAN

Why we choose face recognition over other biometrics?

Facial structure is also a physiological modality that can be used for personal identification and authentication. Human facial structure is an individual characteristic. Facial recognition biometrics makes use of this fact to identify and authenticate individuals. Human brains have natural ability to remember and distinguish different faces. We identify and authenticate people just by recognizing their face on a daily basis. We recognize our family, friends, colleagues, neighbors and pets primarily by their facial structure

Facial recognition system can identify people by processing their digital images if their facial recognition identity has been pre-established. The system takes advantage of digital images or still frames from a video source, which are taken through the facial recognition algorithm. This algorithm extracts data out of facial characteristics like position and shape of eyes, nose, cheekbones and jaw. It can also measure distance between these characteristics and mapped data is stored in a database.

Fingerprint recognition vs. facial recognition

Image: Physiological modalities: fingerprint recognition vs. facial recognition

Biometric modalities are extensively used in personal identification and authentication applications. Physiological modalities are comparatively more stable than behavioral ones, and stay unaffected by factors like mood, psychology and fatigue. Fingerprint recognition is one of the popular modalities, which commonly used for application like physical and logical access control, employee identification, attendance and customer identification. Friction ridges on fingertips are commonly called fingerprints and they are one of the popular physiological characteristics that are used for personal identification.

Having its roots in forensic applications in the past, fingerprint recognition has gained considerable market penetration and popularity in recent years due to extensive use in consumer electronics like mobile phones and national ID programs. Unlike other biometric methods of identification, fingerprint recognition does not require user to stay steady or wear a specific posture.

Despite being the part of physiological biometrics, fingerprint recognition considerably differ from facial recognition. Both the recognition methods have their own advantages and disadvantages, but none of them can replace each other. Facial recognition is good in mass surveillance applications at crowded places, while personal identification with user consent is better achieved by fingerprint recognition. Both the recognition methods have been used in law enforcement extensively. Biometrics has been a part of forensics for more than 100 years, while modern mass surveillance is performed with facial recognition systems by various law enforcement and national security agencies.

3: FACE RECOGNITION

Face recognition is an easy task for humans. Experiments in tu06 have shown that one to three days old babies can distinguish between known faces. So how hard could it be for a computer? It turns out we know little about human recognition to date. Are inner features (eyes, nose, mouth) or outer features (head shape, hairline) used for a successful face recognition? How do we analyze an image and how does the brain encode it? It was shown by David Hubel and Torsten Wiesel, that our brain has specialized nerve cells responding to specific local features of a scene, such as lines, edges, angles or movement. Since we don't see the world as scattered pieces, our visual cortex must somehow combine the different sources of information into useful patterns. Automatic face recognition is all about extracting those meaningful features from an image, putting them into a useful representation and performing some kind of classification on them.

Face recognition based on the geometric features of a face is probably the most intuitive approach to face recognition. One of the first automated face recognition systems was described in [Kanade73]: marker points (position of eyes, ears, nose, ...) were used to build a feature vector (distance between the points, angle between them, ...). The recognition was performed by calculating the euclidean distance between feature vectors of a probe and reference image. Such a method is robust against changes in illumination by its nature, but has a huge drawback: the accurate registration of the marker points is complicated, even with state-of-the-art algorithms. Some of the latest work on geometric face recognition was carried out in [Bru92]. A 22-dimensional feature vector was used and experiments on large datasets have shown, that geometrical features alone my not carry enough information for face recognition.

3.1The eigen face approach

In the language of information theory, the relevant information in a face needs to be extracted, encoded efficiently and one face encoding is compared with the similarly encoded database. The trick behind extracting such kind of information is to capture as many variations as possible from the set of training images. Mathematically, the principal components of the distribution of faces are found out using the eigenface approach. First the eigenvectors of the covariance matrix of the set of face images is found out and then they are sorted according to their corresponding eigenvalues. Then a threshold eigenvalue is taken into account and eigenvectors with eigenvalues less than that threshold values are discarded. So ultimately the eigenvectors having the most significant eigenvalues are selected. Then the set of face images are projected into the significant eigenvectors to obtain a set called eigenfaces. Every face has a contribution to the eigenfaces obtained. The best M eigenfaces from a M dimensional subspace is called “face space” Each individual face can be represented exactly as the linear combination of “eigenfaces” or each face can also be approximated using those significant eigenfaces obtained using the most significant eigen values.

The Eigenfaces method described in [TP91] took a holistic approach to face recognition: A facial image is a point from a high-dimensional image space and a lower-dimensional representation is found, where classification becomes easy. The lower-dimensional subspace is found with Principal Component Analysis, which identifies the axes with maximum variance. While this kind of transformation is optimal from a reconstruction standpoint, it doesn't take any class labels into account. Imagine a situation where the variance is generated from external sources, let it be light. The axes with maximum variance do not necessarily contain any discriminative information at all, hence a classification becomes impossible, so a class-specific projection with a Linear Discriminant Analysis was applied to face recognition in [BHK97]. The basic idea is to minimize the variance within a class, while maximizing the variance between the classes at the same time.

Recently various methods for a local feature extraction emerged. To avoid the high-dimensionality of the input data only local regions of an image are described, the extracted features are (hopefully) more robust against partial occlusion, illummination and small sample size. Algorithms used for a local feature extraction are Gabor Wavelets ([Wiskott97]), Discrete Cosinus Transform ([Messer06]) and Local Binary Patterns ([AHP04]). It's still an open research question what's the best way to preserve spatial information when applying a local feature extraction, because spatial information is potentially useful information.

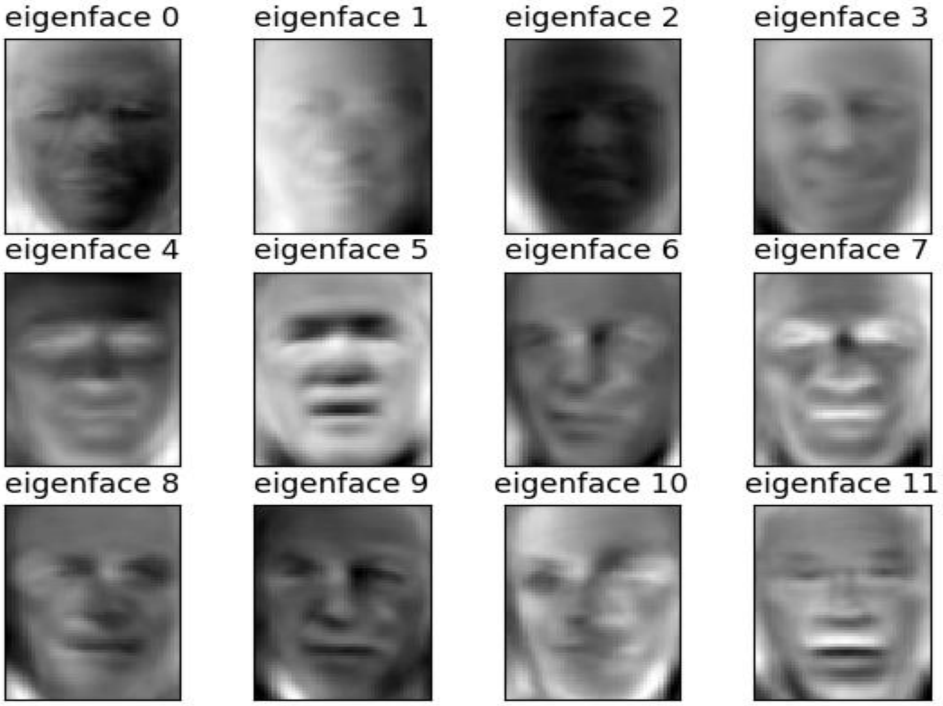
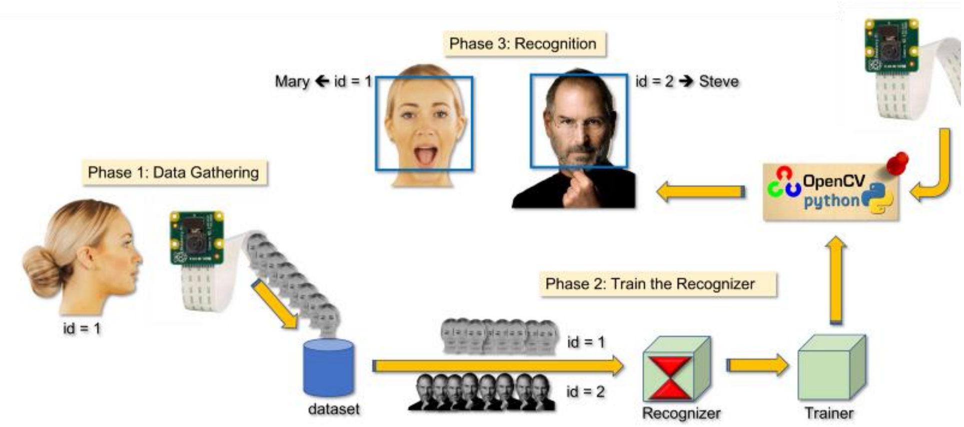


Figure 1: Eigen faces

FACE RECOGNITION BLOCK DIAGRAM:



PHASE 1: DATA GATHERING

PHASE 2: TRAIN THE RECOGNIZER

PHASE 3: RECOGNITION

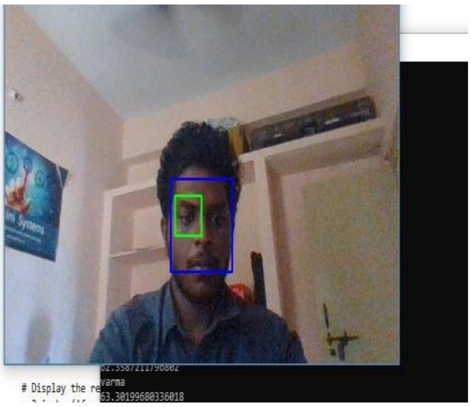
CHAPTER 4: FACE DETECTION

Face detection can be regarded as a specific case of object-class detection. In object-class detection, the task is to find the locations and sizes of all objects in an image that belong to a given class. Examples include upper torsos, pedestrians, and cars.

Face-detection algorithms focus on the detection of frontal human faces. It is analogous to image detection in which the image of a person is matched bit by bit. Image matches with the image stores in database. Any facial feature changes in the database will invalidate the matching process.

Each possible face candidate is normalized to reduce both the lightning effect, which is caused by uneven illumination; and the shirring effect, which is due to head movement. The fitness value of each candidate is measured based on its projection on the eigen-faces. After a number of iterations, all the face candidates with a high fitness value are selected for further verification. At this stage, the face symmetry is measured and the existence of the different facial features is verified for each face candidate.

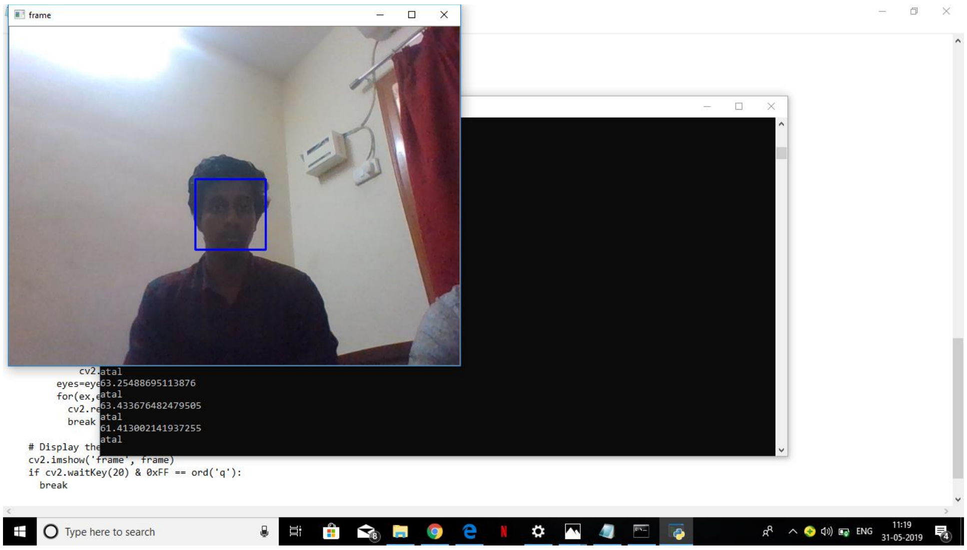
We use cascade classifiers to detect faces. The most commonly used cascade classifiers are haar cascade classifiers. These classifiers can detect faces, eyes, objects etc.



```
# Display the re
```

```
62.3587211796802
varma
63.30199680336018
```

Figure1: A Face is detected along with eyes using haar cascading techniques.



```
          cv2.
        eyes=eye
        for(ex,e
          cv2.re
          break

    # Display the
    cv2.imshow('frame', frame)
    if cv2.waitKey(20) & 0xFF == ord('q'):
      break
```

```
atal
63.25488695113876
atal
63.433676482479505
atal
61.413002141937255
atal
```

PART 5: IMPLEMENTATION OF FACE RECOGNITION TECHNOLOGY

The implementation of face recognition technology includes the following three stages:

- Data acquisition
- Input processing
- Face image classification and decision making

Data acquisition

- The input can be recorded video of the speaker or a still image. A sample of 1 sec duration consists of a 25 frame's video sequence. More than one camera can be used to produce a 3D representation of the face and to protect against the usage of photographs to gain unauthorized access.

Input processing

- A pre-processing module locates the eye position and takes care of the surrounding lighting condition and color variance. First the presence of faces or face in a scene must be detected. Once the face is detected, it must be localized.

- Some facial recognition approaches use the whole face while others concentrate on facial components and/ or regions (such as lips, eyes etc.). The appearance of the face can change change considerably during speech and due to facial expressions.

Face Image Classification and Decision Making

- A synergetic computer is a set of algorithms that simulate synergetic phenomena. In training phases the BIOID creates a prototype called face print for each person. A newly recorded pattern is pre-processed and compared with each face print stored in the database.

As comparisons are made, the system assigns a value to the comparison using a scale of one to ten. If a score is above a predetermined threshold, a match is declared

PROPOSED IMPLEMENTATION

OPERATION SYSTEM:

WINDOWS 10

SUBJECTS:

1. BASIC PYTHON

2.COMPUTER VISION

3.FACE RECOGNITION TECHNIQUES

4.IMAGE PROCESSING

PROGRAMMING LANGUAGE:

1.PYTHON (V.3.7.3)

HARDWARE:

1.RASPBERRY PI BOARD

2.CAMERA

PACKAGES &LIBRARIES

1.CV2

2.NUMPY

3.PICKLE

4.PILLOW

5.SYS

6.REQUESTS

7.SMTPLIB

8.PYAUTOGUI

9.EMAIL

10.DATETIME

11.STRUCT &SOCKET

12.JSON

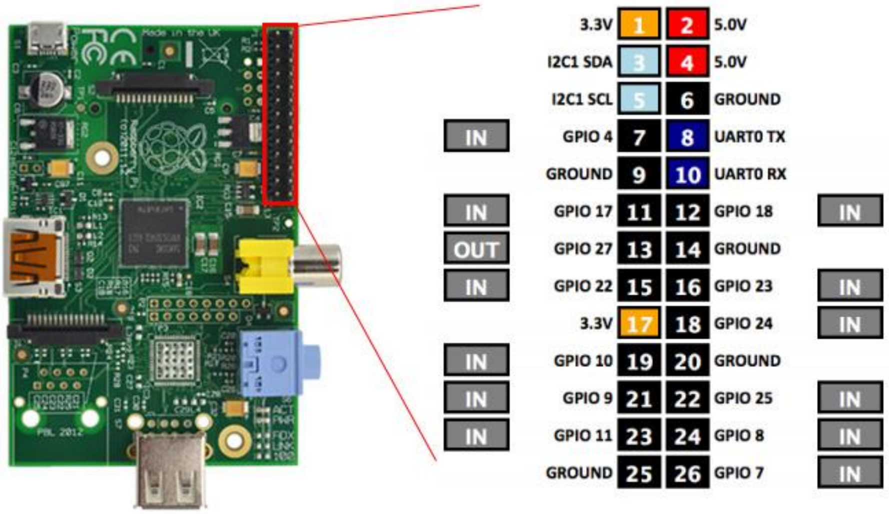
OTHER REQUIREMENTS

Databases consisting of images that has to be identified to train the data.

*the images of the subjects has to be stored in separate folders with their names as file names.

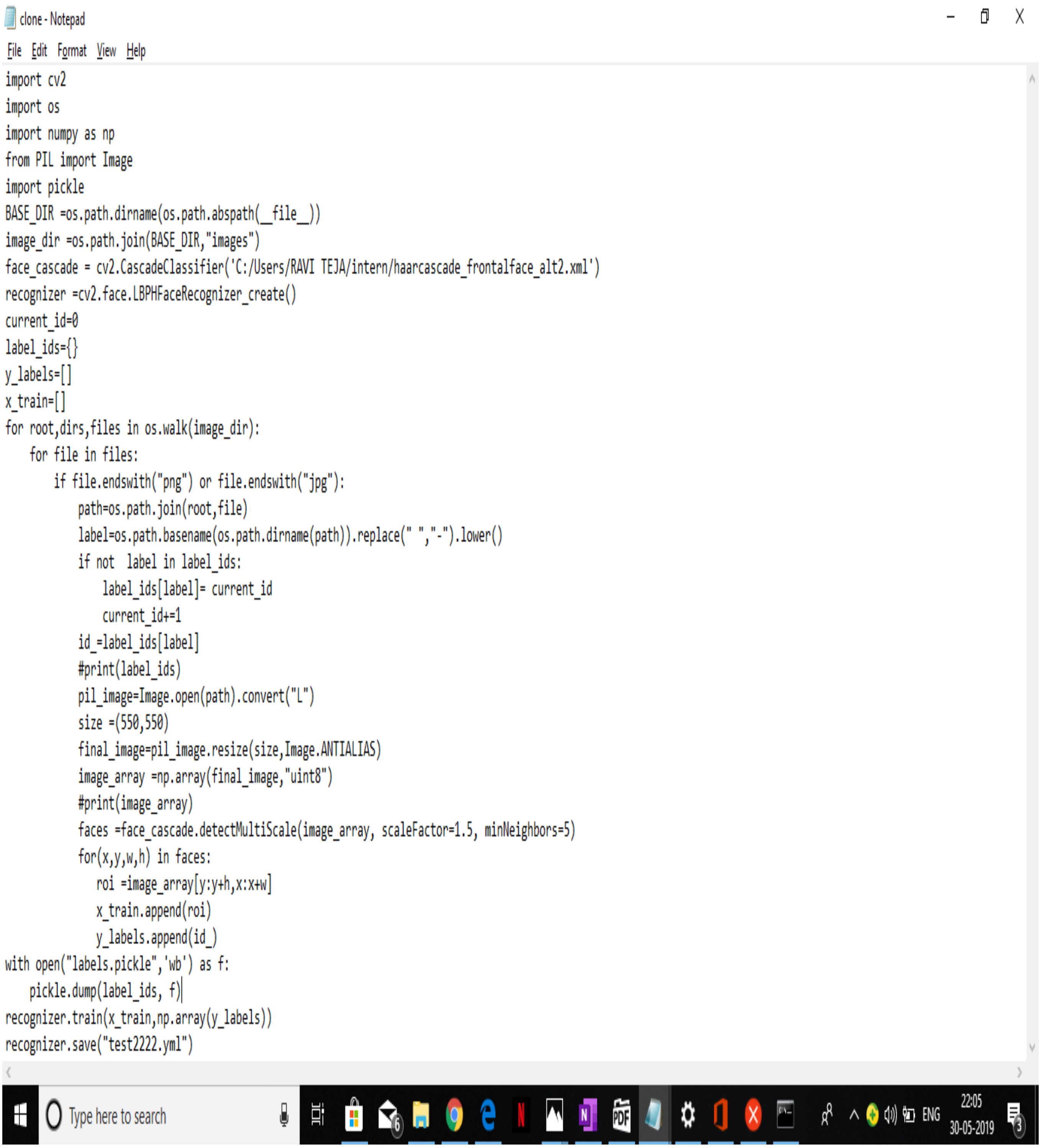
RASPBERRY PI BOARD:

RASPBERRY PI BOARD :



	3.3V	1	2	5.0V	
	I2C1 SDA	3	4	5.0V	
	I2C1 SCL	5	6	GROUND	
IN	GPIO 4	7	8	UART0 TX	
	GROUND	9	10	UART0 RX	
IN	GPIO 17	11	12	GPIO 18	IN
OUT	GPIO 27	13	14	GROUND	
IN	GPIO 22	15	16	GPIO 23	IN
	3.3V	17	18	GPIO 24	IN
IN	GPIO 10	19	20	GROUND	
IN	GPIO 9	21	22	GPIO 25	IN
IN	GPIO 11	23	24	GPIO 8	IN
	GROUND	25	26	GPIO 7	IN

IMAGE PROCESSING

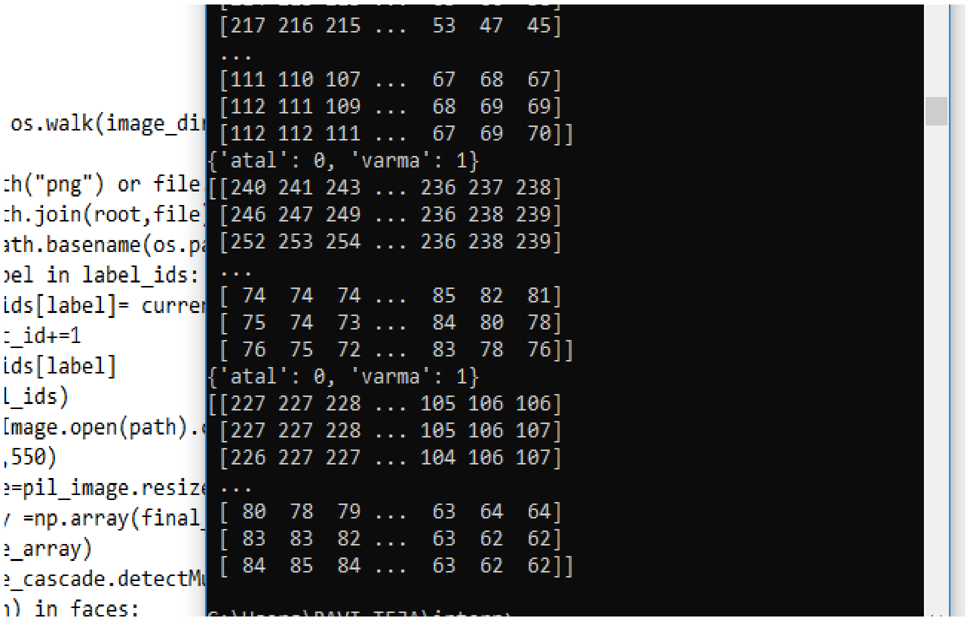


```
import cv2
import os
import numpy as np
from PIL import Image
import pickle
BASE_DIR =os.path.dirname(os.path.abspath(__file__))
image_dir =os.path.join(BASE_DIR,"images")
face_cascade = cv2.CascadeClassifier('C:/Users/RAVI TEJA/intern/haarcascade_frontalface_alt2.xml')
recognizer =cv2.face.LBPHFaceRecognizer_create()
current_id=0
label_ids={}
y_labels=[]
x_train=[]
for root,dirs,files in os.walk(image_dir):
    for file in files:
        if file.endswith("png") or file.endswith("jpg"):
            path=os.path.join(root,file)
            label=os.path.basename(os.path.dirname(path)).replace(" ","-").lower()
            if not  label in label_ids:
                label_ids[label]= current_id
                current_id+=1
            id_=label_ids[label]
            #print(label_ids)
            pil_image=Image.open(path).convert("L")
            size =(550,550)
            final_image=pil_image.resize(size,Image.ANTIALIAS)
            image_array =np.array(final_image,"uint8")
            #print(image_array)
            faces =face_cascade.detectMultiScale(image_array, scaleFactor=1.5, minNeighbors=5)
            for(x,y,w,h) in faces:
               roi =image_array[y:y+h,x:x+w]
               x_train.append(roi)
               y_labels.append(id_)
with open("labels.pickle",'wb') as f:
    pickle.dump(label_ids, f)
recognizer.train(x_train,np.array(y_labels))
recognizer.save("test2222.yml")
```

figure 3:python code to train data.

With the execution of the above code in the cmd prompt the images in the data base.the images in the database are converted into numpy arrays with their respective pixel values. An LPHB recognizer is used to identify the faces.

The below picture shows the numpy arrays into which the given images are converted into.



```
os.walk(image_di

h("png") or file
h.join(root,file
ath.basename(os.pa
el in label_ids:
ids[label]= curren
_id+=1
ids[label]
_ids)
Image.open(path).
,550)
=pil_image.resize
 =np.array(final_
_array)
_cascade.detectMu
) in faces:
```

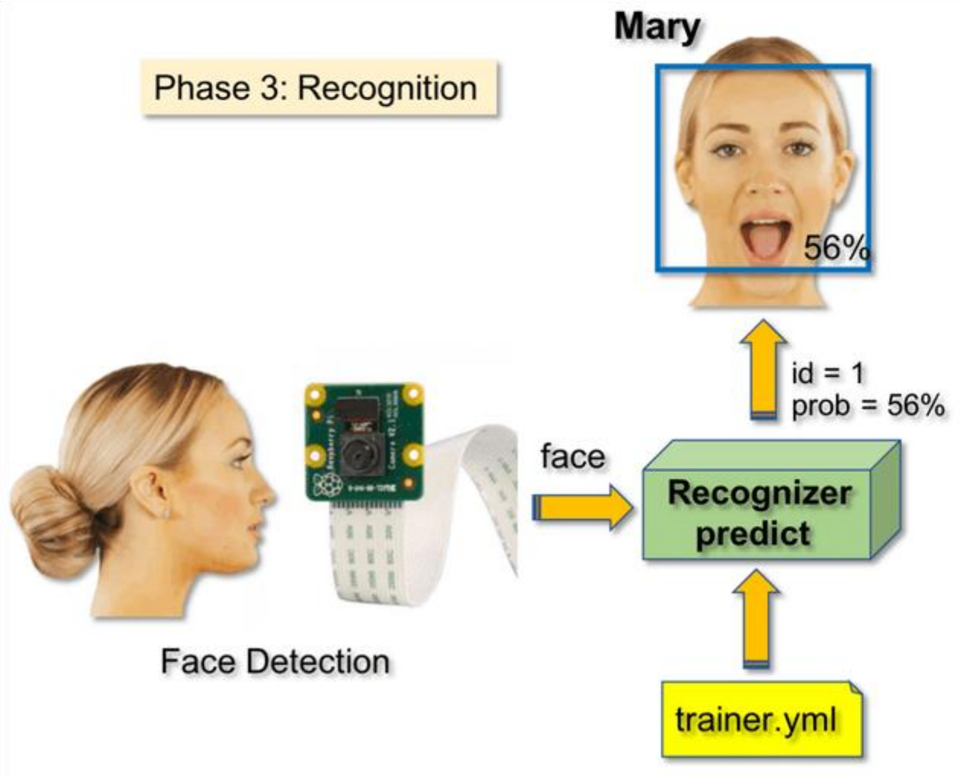
```
 [217 216 215 ...  53  47  45]
 ...
 [111 110 107 ...  67  68  67]
 [112 111 109 ...  68  69  69]
 [112 112 111 ...  67  69  70]]
{'atal': 0, 'varma': 1}
[[240 241 243 ... 236 237 238]
 [246 247 249 ... 236 238 239]
 [252 253 254 ... 236 238 239]
 ...
 [ 74  74  74 ...  85  82  81]
 [ 75  74  73 ...  84  80  78]
 [ 76  75  72 ...  83  78  76]]
{'atal': 0, 'varma': 1}
[[227 227 228 ... 105 106 106]
 [227 227 228 ... 105 106 107]
 [226 227 227 ... 104 106 107]
 ...
 [ 80  78  79 ...  63  64  64]
 [ 83  83  82 ...  63  62  62]
 [ 84  85  84 ...  63  62  62]]
```

Figure a. The numpy arrays with their labels as atal:0 and varma:1

The numpy arrays are stored in a .yml file which contains all the processed images data. This .yml file is used in the face recognizer to detect and identify the faces.

A recognizer is created in the main program and extracts this trained data to compare with the data obtained from the live feed from the camera.

IMPLEMENTATION OF DETECTION AND IDENTIFICATION

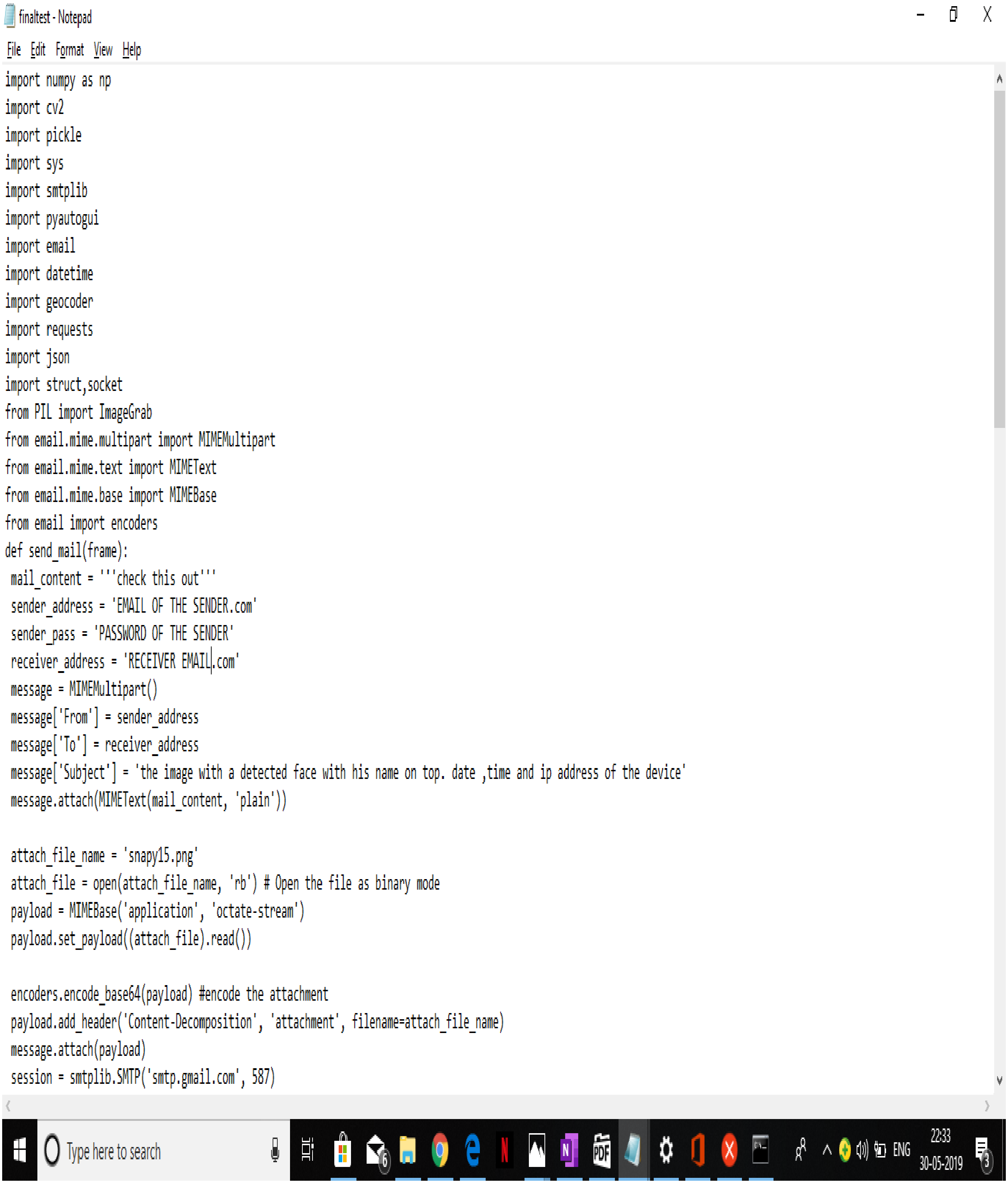


In this phase the image is detected and if it meets the confidence limit a rectangular box is drawn around the detected image along with the name of the person, time and date at which the face is detected.

An IP address is also added to it so the location of the device can be identified with the host network IP address. An email with the snapshot of the identified face and the data is sent via mail.

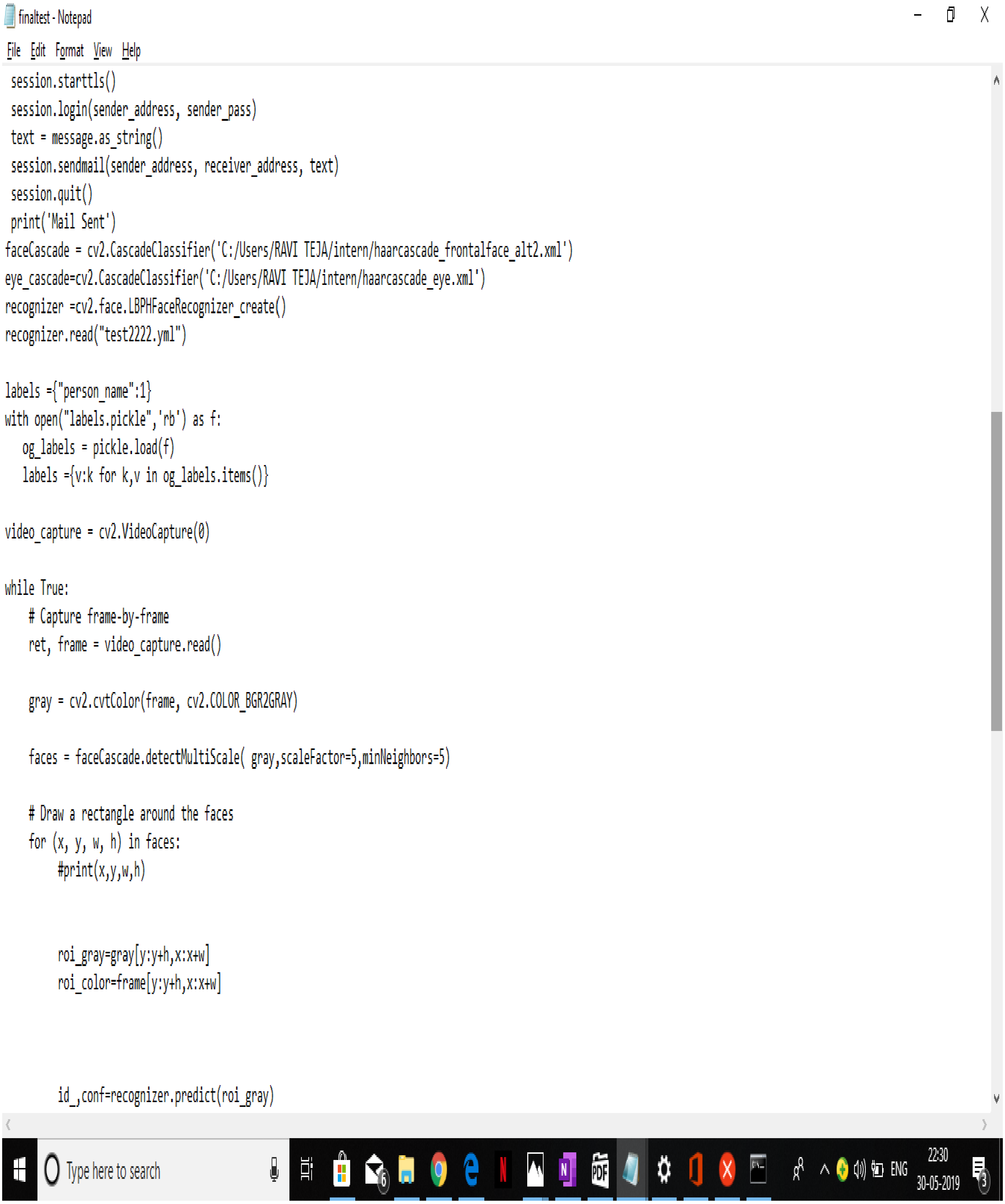
Fig: face detected by the recognizer used.

Program to recognize face:



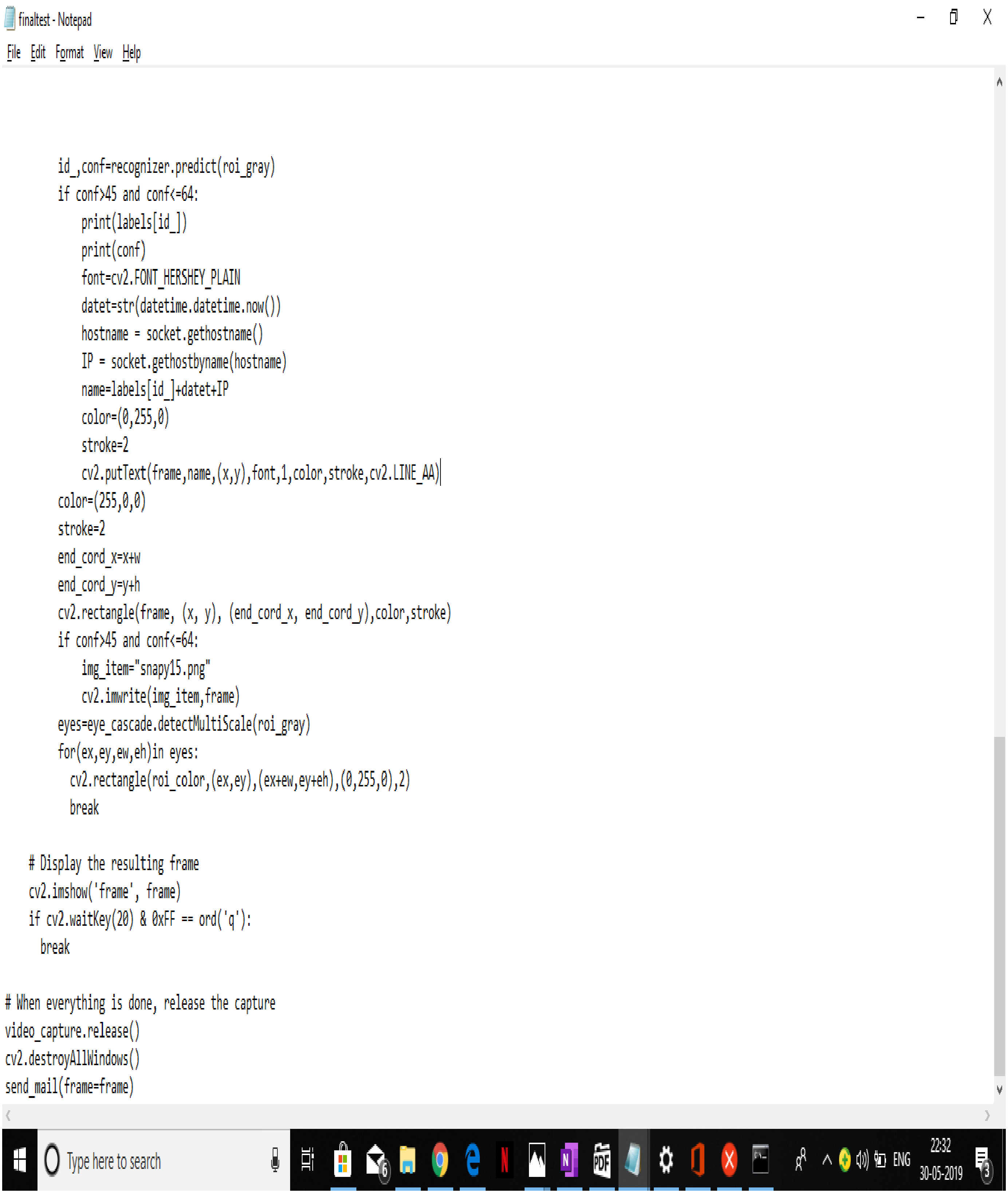
```
import numpy as np
import cv2
import pickle
import sys
import smtplib
import pyautogui
import email
import datetime
import geocoder
import requests
import json
import struct,socket
from PIL import ImageGrab
from email.mime.multipart import MIMEMultipart
from email.mime.text import MIMEText
from email.mime.base import MIMEBase
from email import encoders
def send_mail(frame):
 mail_content = '''check this out'''
 sender_address = 'EMAIL OF THE SENDER.com'
 sender_pass = 'PASSWORD OF THE SENDER'
 receiver_address = 'RECEIVER EMAIL.com'
 message = MIMEMultipart()
 message['From'] = sender_address
 message['To'] = receiver_address
 message['Subject'] = 'the image with a detected face with his name on top. date ,time and ip address of the device'
 message.attach(MIMEText(mail_content, 'plain'))

 attach_file_name = 'snapy15.png'
 attach_file = open(attach_file_name, 'rb') # Open the file as binary mode
 payload = MIMEBase('application', 'octate-stream')
 payload.set_payload((attach_file).read())

 encoders.encode_base64(payload) #encode the attachment
 payload.add_header('Content-Decomposition', 'attachment', filename=attach_file_name)
 message.attach(payload)
 session = smtplib.SMTP('smtp.gmail.com', 587)
```

fig (1)



```
 session.starttls()
 session.login(sender_address, sender_pass)
 text = message.as_string()
 session.sendmail(sender_address, receiver_address, text)
 session.quit()
 print('Mail Sent')
faceCascade = cv2.CascadeClassifier('C:/Users/RAVI TEJA/intern/haarcascade_frontalface_alt2.xml')
eye_cascade=cv2.CascadeClassifier('C:/Users/RAVI TEJA/intern/haarcascade_eye.xml')
recognizer =cv2.face.LBPHFaceRecognizer_create()
recognizer.read("test2222.yml")

labels ={"person_name":1}
with open("labels.pickle",'rb') as f:
   og_labels = pickle.load(f)
   labels ={v:k for k,v in og_labels.items()}

video_capture = cv2.VideoCapture(0)

while True:
    # Capture frame-by-frame
    ret, frame = video_capture.read()

    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

    faces = faceCascade.detectMultiScale( gray,scaleFactor=5,minNeighbors=5)

    # Draw a rectangle around the faces
    for (x, y, w, h) in faces:
         #print(x,y,w,h)


         roi_gray=gray[y:y+h,x:x+w]
         roi_color=frame[y:y+h,x:x+w]



         id_,conf=recognizer.predict(roi_gray)
```

Fig(2



```
        id_,conf=recognizer.predict(roi_gray)
        if conf>45 and conf<=64:
            print(labels[id_])
            print(conf)
            font=cv2.FONT_HERSHEY_PLAIN
            datet=str(datetime.datetime.now())
            hostname = socket.gethostname()
            IP = socket.gethostbyname(hostname)
            name=labels[id_]+datet+IP
            color=(0,255,0)
            stroke=2
            cv2.putText(frame,name,(x,y),font,1,color,stroke,cv2.LINE_AA)
        color=(255,0,0)
        stroke=2
        end_cord_x=x+w
        end_cord_y=y+h
        cv2.rectangle(frame, (x, y), (end_cord_x, end_cord_y),color,stroke)
        if conf>45 and conf<=64:
            img_item="snapy15.png"
            cv2.imwrite(img_item,frame)
        eyes=eye_cascade.detectMultiScale(roi_gray)
        for(ex,ey,ew,eh)in eyes:
          cv2.rectangle(roi_color,(ex,ey),(ex+ew,ey+eh),(0,255,0),2)
          break

    # Display the resulting frame
    cv2.imshow('frame', frame)
    if cv2.waitKey(20) & 0xFF == ord('q'):
      break

# When everything is done, release the capture
video_capture.release()
cv2.destroyAllWindows()
send_mail(frame=frame)
```

fig (3)

Outputs:

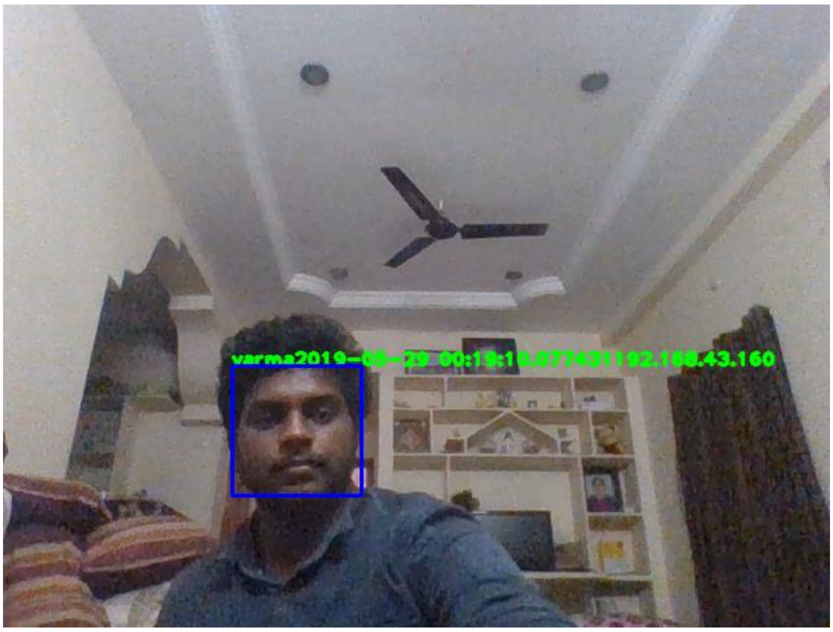
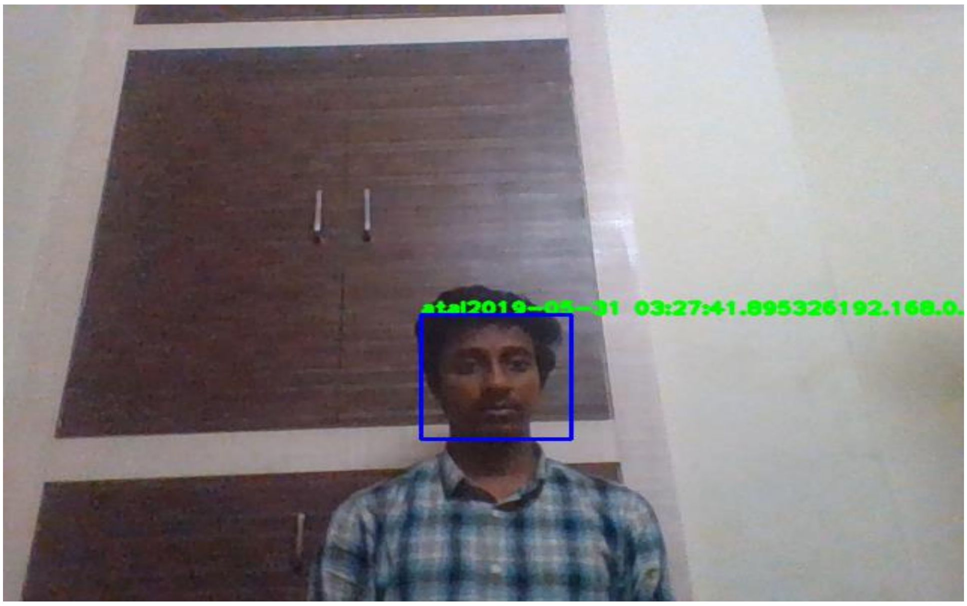


Fig :output of detected face.

Fig 2:



CHAPTER 6:

ADVANTAGES &DISADVANTAGES:

Advantages:

a. There are many benefits to face recognition systems such as its convenience and Social acceptability. All you need is your picture taken for it to work.

b. Face recognition is easy to use and in many cases it can be performed without a Person even knowing.

c. Face recognition is also one of the most inexpensive biometric in the market and Its price should continue to go down.

Disadvantages:

a. Face recognition systems can't tell the difference between identical twins.

APPLICATIONS:

Government Use:

a. Law Enforcement: Minimizing victim trauma verifying Identify for court records, and comparing school surveillance camera images to know child molesters.

b. Security/Counterterrorism: Access control, comparing surveillance images to Know terrorist.

c. Immigration: Rapid progression through Customs.

Commercial Use:

a. Residential Security: Alert homeowners of approaching personnel.

b. Banking using ATM: The software is able to quickly verify a customer's face.

PART 7:

SCOPE FOR FUTURE WORK

The use of spherical canonical images allows us to perform matching in the spherical harmonic transform domain, which does not require preliminary alignment of the images. The errors introduced by embedding into an expressional space with some predefined geometry are avoided. In this facial expression recognition setup, end-to-end processing comprises the face surface acquisition and reconstruction, smoothening, sub sampling to approximately 2500 points.

Facial surface cropping measurement of large positions of distances between all the points using a parallelized parametric version is utilized. The general experimental evaluation of the face expressional system guarantees better face recognition rates. Having examined techniques to cope with expression variation, in future it may be investigated in more depth about the face classification problem and optimal fusion of color and depth information.

Further study can be laid down in the direction of allele of gene matching to the geometric factors of the facial expressions. The genetic property evolution framework for facial expressional system can be studied to suit the requirement of different security models such as criminal detection, governmental confidential security breaches etc.

CONCLUSION:

Face recognition technologies have been associated generally with very costly top secure applications. Today technologies have evolved and the cost of equipment's is going down dramatically due to the integration and the increasing processing power. Certain applications of face recognition technology are now cost effective, reliable and highly accurate. As a result there are no technological or financial barriers for stepping from the pilot project to widespread.

The physiological characteristics of the human face with relevance to various expressions such as happiness, sadness, fear, anger, surprise and disgust are associated with geometrical structures which restored as base matching template for the recognition system.

THANK YOU.......