**AI PROJECT:**

**FACIAL RECOGNITION**

**BY**

**AKSHAT SHARMA (B40)**

**SAGAR SHARMA (B41)**

**K18KK**



**Lovely Professional University**

**INTRODUCTION**

Facial recognition is a technology which is capable of identifying a person from a image or a video frame. Face Recognition is a recognition technique used to detect faces of individuals whose images saved in the data set. Despite other method of identification can be more accurate still it is used in many applications. In this, the system calculates the similarity between the input facial features and the facial features stored in the data set. Face recognition is extremely lightweight, which allows it to run smoothly on a wide range of devices, even without an Internet connection.

Facial recognition can be used in security systems and it can be compared to other biometrics like fingerprint or eye iris recognition system.

It could also be used in the university to mark the attendance of the students.

It can also be used in other avenues like ATM and accessing confidential files.

**WORK ANALYSIS**

In this project, we have firstly made a GUI interface which has certain options like register face, verify face and exit.

Then firstly we have to register the face and in this part of the program, the webcam will firstly open and the program will capture the images from the video source after every 1 sec and it will capture 100 photos.

And in other part of the program, the program will compare the input faces from the faces in the dataset.

If the face is recognized, then it will say unlocked

If the face is not recognized, then it will say locked

If the face not found in the rectangular box, it will say face not found.

And if the face is recognized it will also give confidence percentage which tells us how much features of face is matched.

**IMPLEMENTATION**

Our project consists of mainly 3 main sub parts –

1) GUI interface

2) Register Face

3) Verify Face

**1)** Firstly, GUI interface, in this with the help of tkinter we have made a gui interface which will give certain options which the user can choose. In the interface we have also added buttons from which user can choose from.

**CODE OF GUI**

|  |
| --- |
| from tkinter import \* |
|  | top = Tk() |
|  | top.title("FACE RECOGNITION") |
|  | top. Geometry("600x300") |
|  |  |
|  | def register\_face(): |
|  | def verify\_face(): |
|  | lb = Label(top,text="Welcome To Facial Recognition",font=("Arial Black",20)).pack(pady=5) |
|  |  |
|  | l1 = Button(top,text="Register Face",command=register\_face) |
|  | l1.pack(pady=25) |
|  |  |
|  | l2 = Button(top,text="Verify Face",command=verify\_face) |
|  | l2.pack(pady=10) |
|  |  |
|  | l3 = Button(top,text="Exit",command=exit) |
|  | l3.pack(pady=15) |
|  |  |
|  | top.mainloop() |

**2)** In the second part of the program, we have to register the face and in this part of the program, the webcam will firstly open and the program will capture 100 images from the video source after every 1 sec and it will capture 100 photos and it will capture the images in grayscale only because we have to compare images in the next part of program that’s why images has to be in grayscale.

**CODE FOR REGISTRING FACE**-

import cv2

import numpy as np

face\_classifier = cv2.CascadeClassifier('C:/Users/DELL/Desktop/AI/haarcascade\_frontalface\_default.xml')

def face\_extractor(img):

gray=cv2.cvtColor(img,cv2.COLOR\_BGR2GRAY)

faces = face\_classifier.detectMultiScale(gray,1.3,5)

if faces is():

return None

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

cropped\_face = img[y:y+h,x:x+w]

return cropped\_face

cap = cv2.VideoCapture(0)

count = 0

while True:

ret,frame = cap.read()

if face\_extractor(frame) is not None:

count+=1

face = cv2.resize(face\_extractor(frame),(200,200))

face = cv2.cvtColor(face,cv2.COLOR\_BGR2GRAY)

file\_path = 'C:/Users/DELL/Desktop/AI/samples/'+str(count)+'.jpg'

cv2.imwrite(file\_path,face)

cv2.putText(face,str(count),(50,50),cv2.FONT\_HERSHEY\_COMPLEX,1,(0,250,0),2)

cv2.imshow('Face Cropper',face)

else:

print('Face not found')

pass

if cv2.waitKey(1)==13 or count==100:

break

capture.release()

cv2.destroyAllWindows()

print("samples collected")

**3)** In the last part of the program, firstly the model is trained so that it can recognize the input face.

Then after that webcam will open , then it will match input face with the data stored in dataset and it will calculate confidence and if the confidence is more than 82 ,then it will say “unlocked” and will also print the confidence but if confidence is less than 82 then it will say “locked” but if the face is not found in rectangular box then it will say “face not found”.

**CODE FOR FACE RECOGNIZITION**

import cv2

import numpy as np

from os import listdir

from os.path import isfile, join

data\_path = 'C:/Users/AKSHAT SHARMA/Desktop/AI'

onlyfiles = [f for f in listdir(data\_path) if isfile(join(data\_path,f))]

Training\_Data, Labels = [], []

for i, files in enumerate(onlyfiles):

image\_path = data\_path + onlyfiles[i]

images = cv2.imread(image\_path, cv2.IMREAD\_GRAYSCALE)

Training\_Data.append(np.asarray(images, dtype=np.uint8))

Labels.append(i)

Labels = np.asarray(Labels, dtype=np.int32)

model = cv2.face.LBPHFaceRecognizer\_create()

model.train(np.asarray(Training\_Data), np.asarray(Labels))

print("Model Training Complete!!!!!")

face\_classifier = cv2.CascadeClassifier('C:/Users/AKSHAT SHARMA/Desktop/AI/haarcascade\_frontalface\_default.xml')

def face\_detector(img, size = 0.5):

gray = cv2.cvtColor(img, cv2.COLOR\_BGR2GRAY)

faces = face\_classifier.detectMultiScale(gray,1.3,5)

if faces is():

return img,[]

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

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

roi = img[y:y+h, x:x+w]

roi = cv2.resize(roi, (200,200))

return img,roi

capture = cv2.VideoCapture(0)

while True:

ret, frame = capture.read()

image, face = face\_detector(frame)

try:

face = cv2.cvtColor(face, cv2.COLOR\_BGR2GRAY)

result = model.predict(face)

if result[1] < 500:

confidence = int(100\*(1-(result[1])/300))

display\_string = str(confidence)+'% Confidence it is user'

if confidence > 82:

cv2.putText(image,display\_string,(100,120), cv2.FONT\_HERSHEY\_COMPLEX,1,(250,120,255),2)

cv2.putText(image, "Unlocked", (250, 450), cv2.FONT\_HERSHEY\_COMPLEX, 1, (0, 255, 0), 2)

cv2.imshow('Face Cropper', image)

else:

cv2.putText(image, "Locked", (250, 450), cv2.FONT\_HERSHEY\_COMPLEX, 1, (0, 0, 255), 2)

cv2.imshow('Face Cropper', image)

except:

cv2.putText(image, "Face Not Found", (250, 450), cv2.FONT\_HERSHEY\_COMPLEX, 1, (255, 0, 0), 2)

cv2.imshow('Face Cropper', image)

pass

if cv2.waitKey(1)==13:

break

capture.release()

cv2.destroyAllWindows()

**FULL PROGRAM CODE COMBINED**

from tkinter import \*

top = Tk()

top.title("FACE RECOGNITION")

top.geometry("600x300")

def register\_face():

import cv2

import numpy as np

face\_classifier = cv2.CascadeClassifier('C:/Users/AKSHAT SHARMA/Desktop/AI/haarcascade\_frontalface\_default.xml')

def face\_extractor(img):

gray=cv2.cvtColor(img,cv2.COLOR\_BGR2GRAY)

faces = face\_classifier.detectMultiScale(gray,1.3,5)

if faces is():

return None

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

cropped\_face = img[y:y+h,x:x+w]

return cropped\_face

cap = cv2.VideoCapture(0)

count = 0

while True:

ret,frame = cap.read()

if face\_extractor(frame) is not None:

count+=1

face = cv2.resize(face\_extractor(frame),(200,200))

face = cv2.cvtColor(face,cv2.COLOR\_BGR2GRAY)

file\_path = 'C:/Users/AKSHAT SHARMA/Desktop/AI/SAMPLES/user'+str(count)+'.jpg'

cv2.imwrite(file\_path,face)

cv2.putText(face,str(count),(50,50),cv2.FONT\_HERSHEY\_COMPLEX,1,(0,250,0),2)

cv2.imshow('Face Cropper',face)

else:

print('Face not found')

pass

if cv2.waitKey(1)==13 or count==100:

break

cap.release()

cv2.destroyAllWindows()

print("samples collected")

def verify\_face():

import cv2

import numpy as np

from os import listdir

from os.path import isfile, join

data\_path = 'C:/Users/AKSHAT SHARMA/Desktop/AI/SAMPLES/'

onlyfiles = [f for f in listdir(data\_path) if isfile(join(data\_path,f))]

Training\_Data, Labels = [], []

for i, files in enumerate(onlyfiles):

image\_path = data\_path + onlyfiles[i]

images = cv2.imread(image\_path, cv2.IMREAD\_GRAYSCALE)

Training\_Data.append(np.asarray(images, dtype=np.uint8))

Labels.append(i)

Labels = np.asarray(Labels, dtype=np.int32)

model = cv2.face.LBPHFaceRecognizer\_create()

model.train(np.asarray(Training\_Data), np.asarray(Labels))

print("Model Training Complete!!!!!")

face\_classifier = cv2.CascadeClassifier('C:/Users/AKSHAT SHARMA/Desktop/AI/haarcascade\_frontalface\_default.xml')

def face\_detector(img, size = 0.5):

gray = cv2.cvtColor(img, cv2.COLOR\_BGR2GRAY)

faces = face\_classifier.detectMultiScale(gray,1.3,5)

if faces is():

return img,[]

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

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

roi = img[y:y+h, x:x+w]

roi = cv2.resize(roi, (200,200))

return img,roi

cap = cv2.VideoCapture(0)

while True:

ret, frame = cap.read()

image, face = face\_detector(frame)

try:

face = cv2.cvtColor(face, cv2.COLOR\_BGR2GRAY)

result = model.predict(face)

if result[1] < 500:

confidence = int(100\*(1-(result[1])/300))

display\_string = str(confidence)+'% Confidence it is user'

if confidence > 82:

cv2.putText(image,display\_string,(100,120), cv2.FONT\_HERSHEY\_COMPLEX,1,(250,120,255),2)

cv2.putText(image, "Unlocked", (250, 450), cv2.FONT\_HERSHEY\_COMPLEX, 1, (0, 255, 0), 2)

cv2.imshow('Face Cropper', image)

else:

cv2.putText(image, "Locked", (250, 450), cv2.FONT\_HERSHEY\_COMPLEX, 1, (0, 0, 255), 2)

cv2.imshow('Face Cropper', image)

except:

cv2.putText(image, "Face Not Found", (250, 450), cv2.FONT\_HERSHEY\_COMPLEX, 1, (255, 0, 0), 2)

cv2.imshow('Face Cropper', image)

pass

if cv2.waitKey(1)==13:

break

cap.release()

cv2.destroyAllWindows()

lb = Label(top,text="Welcome To Facial Recognition",font=("Arial Black",20)).pack(pady=5)

l1 = Button(top,text="Register Face",command=register\_face)

l1.pack(pady=25)

l2 = Button(top,text="Verify Face",command=verify\_face)

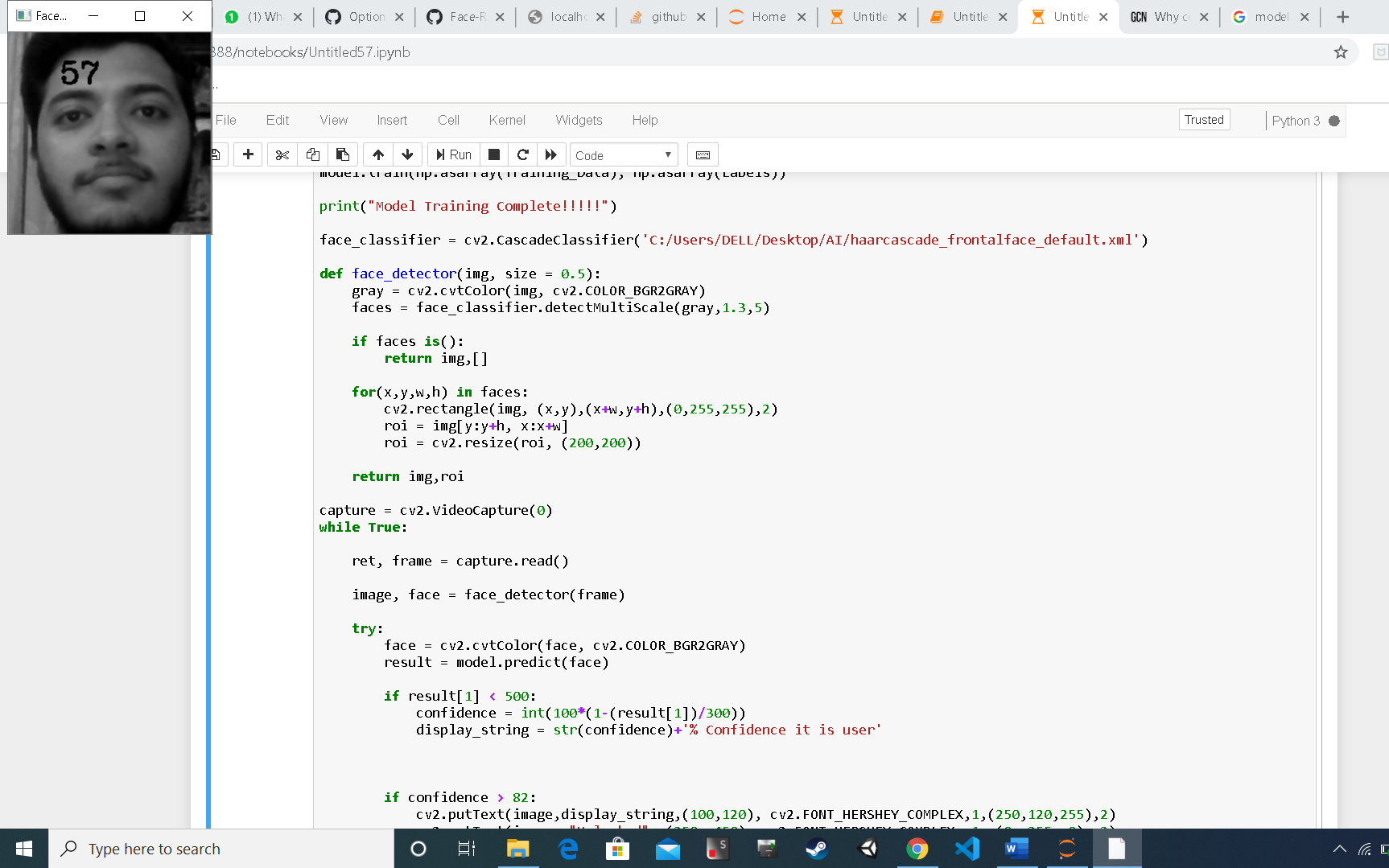
l2.pack(pady=10)

l3 = Button(top,text="Exit",command=exit)

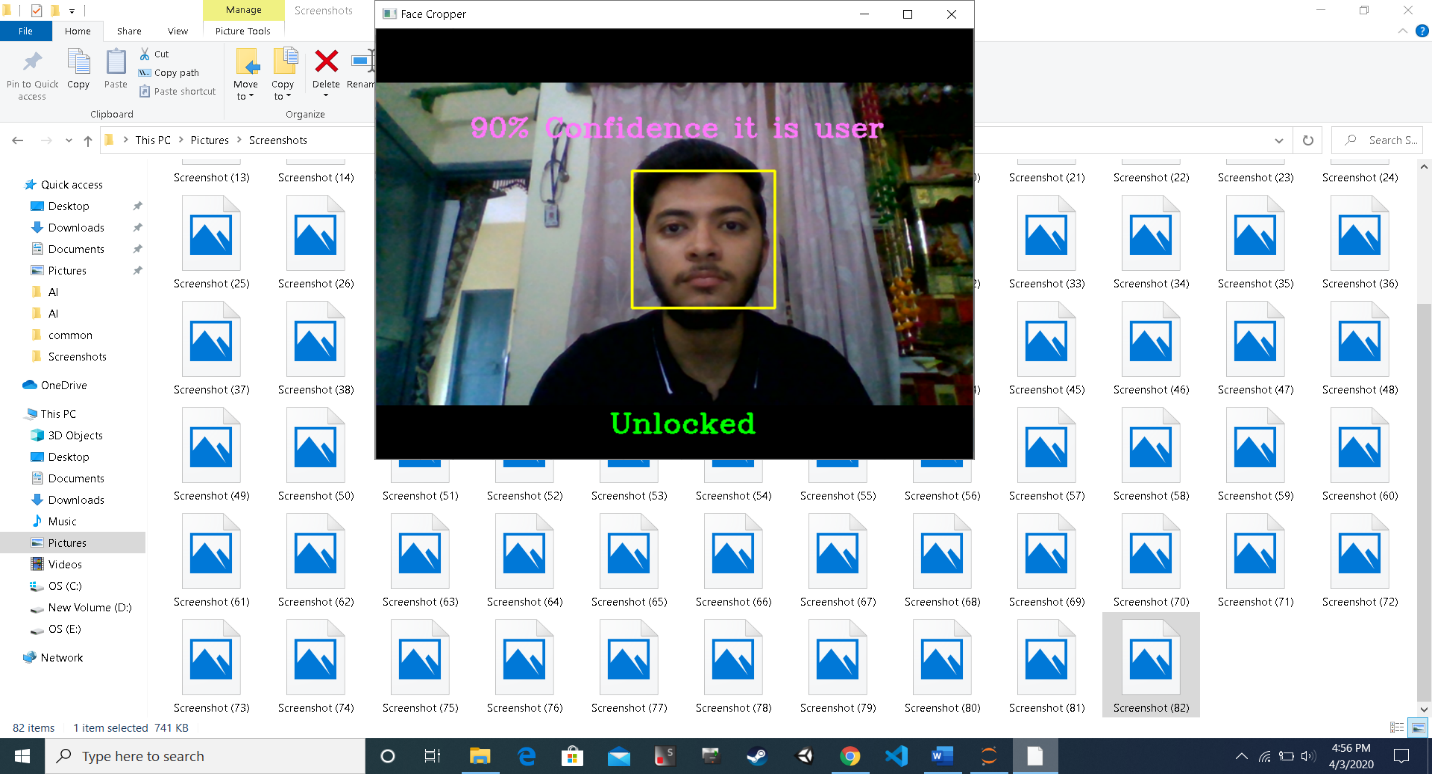
l3.pack(pady=15)

top.mainloop()

**RESULTS**

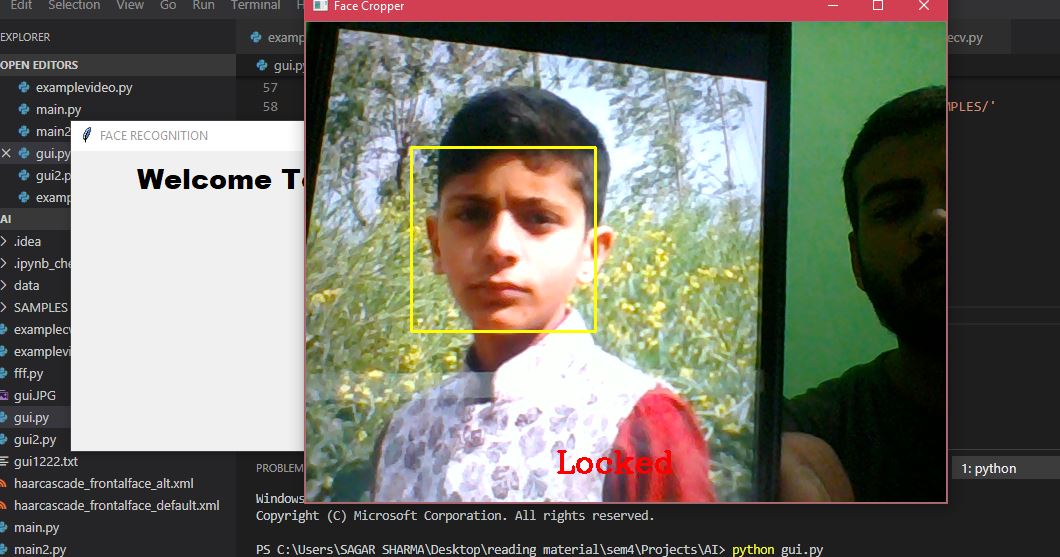
****

as you can see in this photo, the second part of the program i.e. register\_face is running and it will register my face and it will capture 100 photos and store in the data set(samples).



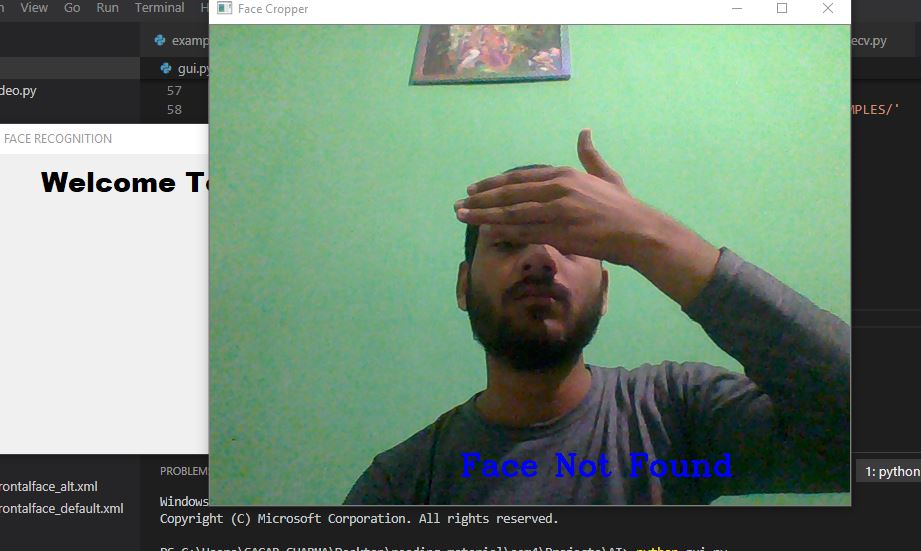
In this you can see as the confidence is more than 82 because the photo is stored in the data set so it is printing unlocked and also printing the confidence

**LOCKED**

****

As you can see this photo is not stored in the data set that’s why it is printing locked

**FACE NOT FOUND**

****

As you can see in this photo the model is not able to detect the face that’s why it is printing face not found

**IMPORTANT LIBRARIES USED**

* **Open cv:** OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. 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
* **OS:** The OS module in python provides functions for interacting with the operating system. OS, comes under Python’s standard utility modules. This module provides a portable way of using operating system dependent functionality.

**listdir -** method in python is used to get the list of all files and directories in the specified directory. If we don’t specify any directory, then list of files and directories in the current working directory will be returned.

**Is file** - This method in Python is used to check whether the specified path is an existing regular file or not.

* **NUMPY-** NumPy is the core library for scientific computing in Python. It provides a high-performance multidimensional array object, and tools for working with these arrays. A NumPy array is a grid of values, all of the same type, and is indexed by a tuple of nonnegative integers. The number of dimensions is the rank of the array

**TEAM RESPONSIBILITIES**

**The code for Gui interface and the register\_face is written by sagar Sharma whereas the code for face recognition and model training (i.e. verify\_face) is written by Akshat Sharma.**

**REFERENCES**

* **Geeksforgeeks.com**
* **Pyimagesearch.com**
* **Realpython.com**
* **Towardsdatascience.com**
* **You tube**

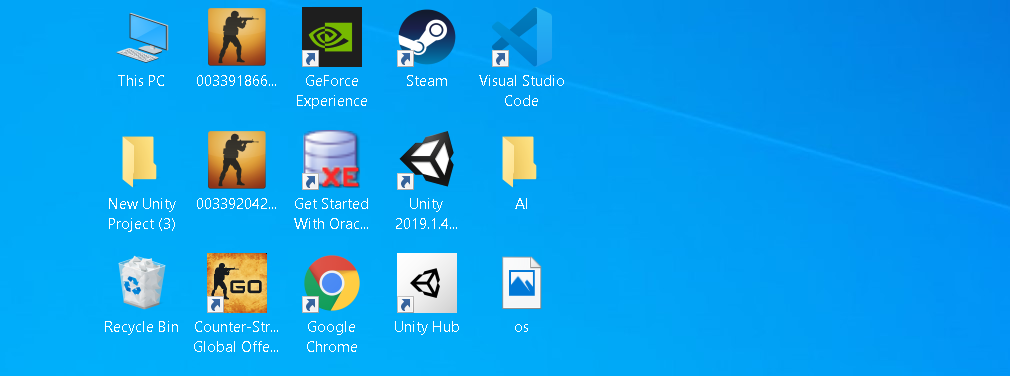
**READ ME**

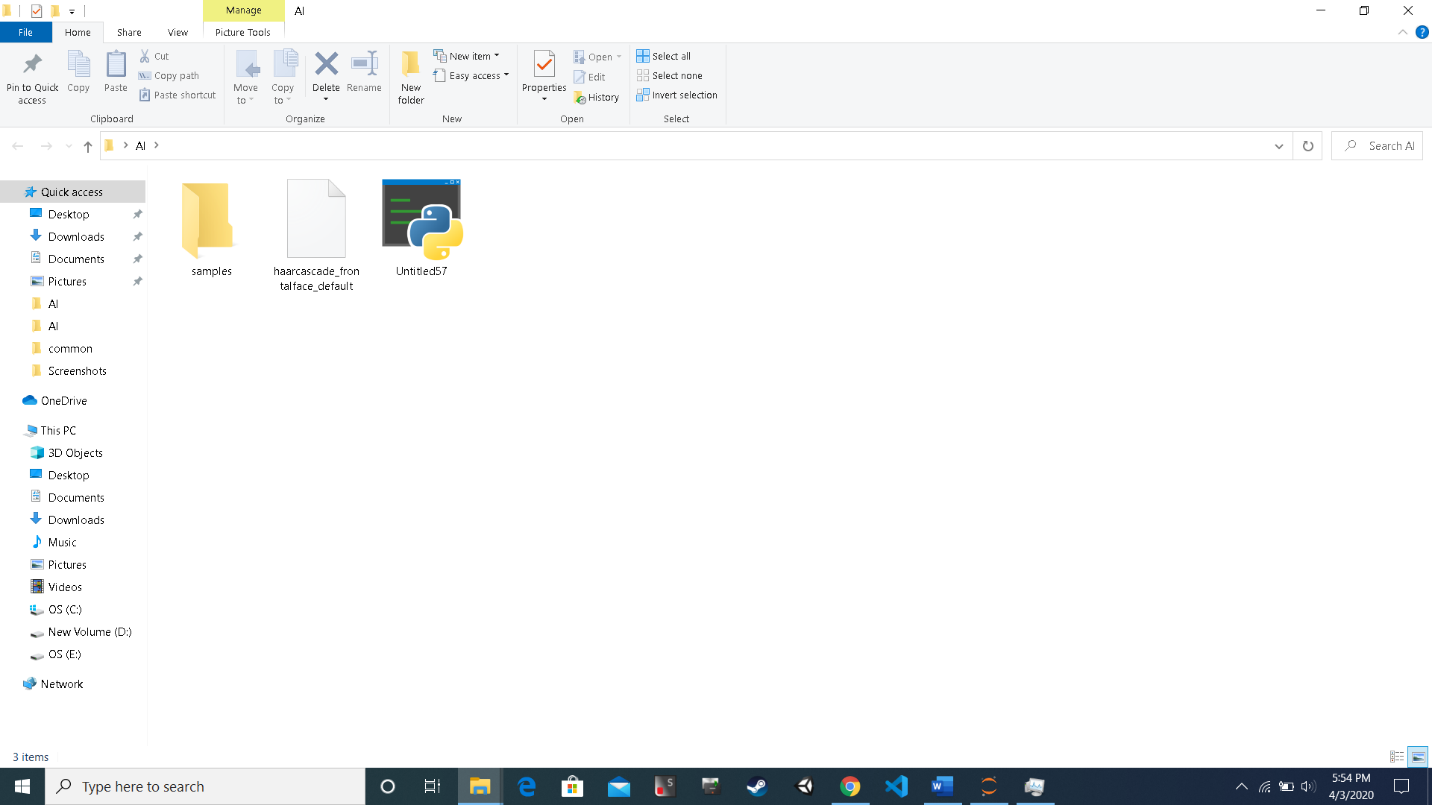
**In order to run the program, firstly the data path had to be changed and you have to create a folder and then you have to save Cascade classifier in it and in that folder you can create a folder named “samples” where the registered face will be stored.**

**Firstly in the register\_face method, you have to give the path of the cascade classifier and then in the ”file\_path” you have give the path of folder where the registered face is stored and in the verify\_face method again you have to give the path of cascade classifier.**

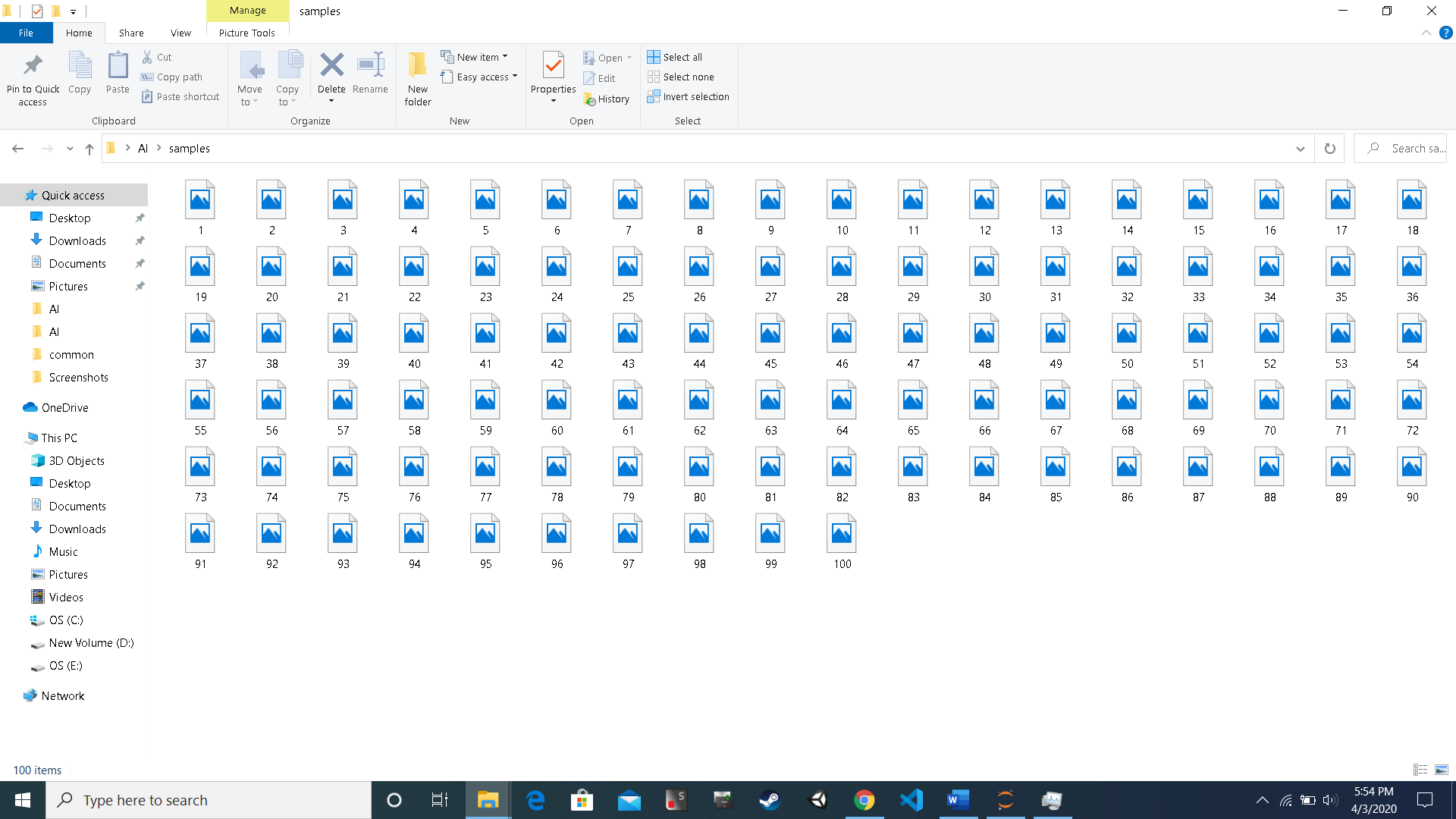
**Then you can run the program, if you click on register\_face then It will register your face and if you click on verify\_face then it will recognize the face if it is present in the data set.**

**As you can see in my desktop, I have created a folder named AI**

****

****

**And in the AI folder I have saved the cascade classifier and there is another folder named “sample” for storing the registered face.**

****

**And you can see in sample folder, there are registered face are stored**