

A Project Report

ON

Face Recognition system

*Submitted in fully fulfillment of the
requirement for the award of the degree of*

**Bachelor of Technology in Computer Science and
Engineering**



(Established under Galgotias University Uttar Pradesh Act No. 14 of 2011)

**Under the supervision of
Mr. Anupam Lakhanpal**

**Assistant Professor
Department of Computer Science and Engineering**

Submitted By

**Nipun Pratap Singh (19SCSE1010245)
Subham Singh (19SCSE1010222)**

**SCHOOL OF COMPUTING SCIENCE AND ENGINEERING
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
GALGOTIAS UNIVERSITY, GREATER NOIDA, INDIA**

DECEMBER 2021



**SCHOOL OF COMPUTING SCIENCE AND ENGINEERING
GALGOTIAS UNIVERSITY, GREATER NOIDA**

CANDIDATE'S DECLARATION

We hereby certify that the work which is being presented in the project, entitled "Face Recognition System " in Fully fulfillment of the requirements for the award of the B.Tech submitted in the School of Computing Science and Engineering of Galgotias University, Greater Noida, is an original work carried out during the period of August 2021 to December 2021, under the supervision of Mr. Anupam Lakhanpal, Department of Computer Science and Engineering of School of `Computing Science and Engineering , Galgotias University, Greater Noida .

CERTIFICATE

The Final Project Viva-Voce examination of Subham Singh, Nipun Pratap Singh has been held on 23-12-2021 and the work is recommended for the award of B.Tech.

Signature of Examiner(s)

Signature of Supervisor(s)

Signature of Project Coordinator

Signature of Dean

Date: December, 2021

Place: Greater Noida

ABSTRACT

Face recognition is most demanding field over the past two decades. We can analyze the reason stems from the need for automated visual and monitoring systems which have power to replace all human efforts done manually .These researches involve the knowledge such as neuroscience, computer vision, pattern recognition, machine leaning and many other techniques.

Face recognition is better than other diagnostic procedures namely; fingerprint, iris scanning, signature etc. The face recognition system uses two techniques face detection and face recognition and combine both to build a face recognition system. The first face detection process is performed on acquired images and it will find the object. The second face recognition process uses algorithm to find details about a person's face. There after this distinctive details, such as the distance between the eyes, the distance from the mouth to the nose and eyes or the shape of chin and all the specific details is then converted into statistical representation and compared with the data stored in the facial recognition database.

TABLE OF CONTENTS

CONTENTS	PAGE No:-
<i>Abstract</i>	3
<i>Introduction</i>	5-10
<i>Problem Formulation</i>	11
<i>Required Tools</i>	11
<i>Literature Survey</i>	12
<i>Flowcharts</i>	13-15
<i>Code and its implementation</i>	16-29
<i>Conclusion</i>	30

INTRODUCTION

It is a process/method of verifying and identifying people using their face.

Face recognition system is used to identifying the various faces in video and photos in real-time situation.

The facial recognition program is a part of applications for the processing of facial images and their importance in the field of research is increasing recently. This system uses human biometric information and is easier to use instead of fingerprints, iris scanning, signature etc., because these biometric forms are not very suitable for those people who are not interested in physical interaction. These systems are commonly used and preferred by people with security cameras in city life. Face recognition system can be used to prevent crime, video surveillance, personal security, this virtual system really change global era to very digitalized form where persons verification, and many other similar security activities provide assured results and it is great achievement and a next step for achievement of digital era.

This system has to resolve all the complexities which face during recognition. of individuals as there is lot of cases where some requirements are missing and not fulfilled mean image is not clear or any other traits are not visible , lighting condition, and lots of problem on the images with which software deals .This technology combine both techniques face detection and face matching in image analysis. The detection application is used to detect facial expression in given image. And the matching algorithm is used to distinguished well-constructed images. Face detection method detect face in the given image and Face recognition method is used to extract facial features from the given image like how's the eye , lips, nose looks like . Which makes this algorithm more transparent and complexion than single verification or detection and that task performed by recognition algorithms. The first step we take with the face recognition program is to get a picture from the camera. The second step we have to take is face detection in the image we found. And the third step is to recognize the face from output of detection part. The last step is to retain a person's identity as a result of the recognition component.

Face recognition is one of the most significant and best researches in the field of pattern recognition and computer vision due to its huge applications that apply to the biometric environment, information security control, law enforcement, surveillance system and smart card.

Biometric-based technology has emerged as the most promising option for personal identification in recent years since, instead of verifying people and allowing them access to physical and text-based domains, smart cards, and keys, these methods use a person's personality or behaviour to find and find who he is. Passwords and Pin's are hard to remember and can be stolen or guessed, cards, keys and the like can be misplaced, forgotten or created twice. However, living human characteristics cannot be misplaced, forgotten or stolen.

Face detection and recognition becomes the most biometrics authentication techniques from the past few years. Face recognition is the process of using the face properties in the biometric systems. Face recognition system has two main tasks: verification and identification. Face identification means a 1:N problem that compares a query face image against all image templates in a face database. We perform face recognition, an extremely complex visual task, almost instantaneously and our own recognition ability is far more robust than any computer's can hope to be. We can recognize a familiar individual under

very adverse lighting conditions, from varying angles or view points. Scaling differences (a face being near or far away), different backgrounds do not affect our ability to recognize faces and we can even recognize individuals with just a fraction of their face visible or even after several years have past. Furthermore, we are able to recognize the faces of several thousand individuals whom we have met during our lifetime. In the Face recognition process the input image is compared with the database. The input image is also called as probe and the database is called as gallery. Then it gives a match report and then the classification is done to identify the sub-population to which new observations belong.

There are basically three approaches for face recognition :

PROJECT REPORT

Feature base approach: In feature based approach the local features like nose, eyes are segmented and it can be used as input data in face detection to easier the task of face recognition.

Holistic approach: In holistic approach the whole face taken as the input in the face detection system to perform face recognition.

Hybrid approach : Hybrid approach is combination of feature based and holistic approach. In this approach both local and whole face is used as the input to face detection system.

Face recognition using Artificial Intelligence:- The current technology amazes people with amazing innovations that not only make life simple but also bearable. Face recognition has over time proven to be the least intrusive and fastest form of biometric verification. Facial Recognition is a category of biometric software that maps an individual's facial features and stores the data as a face print. The software uses deep learning algorithms to compare a live captured image to the stored faceprint to verify one's identity. Image processing and machine learning are the backbones of this technology. Face recognition has received substantial attention from researchers due to human activities found in various applications of security like an airport, criminal detection, face tracking, forensic, etc. Compared to other biometric traits like palm print, iris, fingerprint, etc., face biometrics can be non-intrusive. They can be taken even without the user's knowledge and further can be used for security-based applications like criminal detection, face tracking, airport security, and forensic surveillance systems. Face recognition involves capturing face images from a video or a surveillance camera. They are compared with the stored database. Face recognition involves training known images, classify them with known classes, and then they are stored in the database. When a test image is given to the system it is classified and compared with the

- i) **Image Processing and Machine Learning** : Image processing by computers involves the process of Computer Vision. It deals with the high- level understanding of digital images or videos. The requirement is to automate tasks that the human visual systems can do. So, a computer should be able to recognize objects such as that of a face of a human being or a lamppost or even a statue.
- ii) **Image reading** : The computer reads any image as a range of values between 0 and 255. For any color image, there are 3 primary colors – Red, green, and blue. A matrix is formed for every primary color and later

these matrices combine to provide a Pixel value for the individual R, G, B colors. Each element of the matrices provide data about the intensity of the brightness of the pixel.

- iii) **OpenCV:** *OpenCV is a Python library that is designed to solve computer vision problems. OpenCV was originally developed in 1999 by Intel but later supported by Willow Garage. OpenCV supports a variety of programming*

languages such as C++, Python, Java, etc. Support for multiple platforms including Windows, Linux, and macOS. OpenCV Python is a wrapper class for the original C++ library to be used with Python. Using this, all of the OpenCV array structures get converted to/from NumPy arrays. This makes it easier to integrate it with other libraries that use NumPy. For example, libraries such as SciPy and Matplotlib.

- iv) **Machine learning** : Every Machine Learning algorithm takes a dataset as input and learns from the data it basically means to learn the algorithm from the provided input and output as data. It identifies the patterns in the data and provides the desired algorithm. For instance, to identify whose face is present in a given image, multiple things can be looked at as a pattern:

Height/width of the face.:

Height and width may not be reliable since the image could be rescaled to a smaller face or grid. However, even after rescaling, what remains unchanged

PROJECT REPORT

are the ratios - the ratio of the height of the face to the width of the face won't change.

Color of the face.:

Width of other parts of the face like lips, nose, etc.

There is a pattern involved - different faces have different dimensions like the ones above. Similar faces have similar dimensions. Machine Learning algorithms only understand numbers so it is quite challenging. This numerical representation of a "face" (or an element in the training set) is termed as a feature vector. A feature vector comprises of various numbers in a specific order.

PROBLEM FORMULATION

Face recognition programs identify people with their facial images. Face detection systems create an identity for an authorized person rather than just checking whether an identity document (ID) or key is used or whether the user knows the secret numbers (pins) or passwords. A standard facial vision statement can be made as follows photos provided of motion pictures or video of the group identifying or confirming one or more people in of the event we use a database of facial data Available details of such collateral as the age of race facial expressions and expressions can be used to reduce search improves the realization of problem solving including the division of face-to-face acquisitions from scenes that include feature from the facial region recognition or validation of input detection problems is anonymous face and system reports back to a specified id from the database of people who are known when the system verification problems need to be verified or declines the required ownership of the input face.

REQUIRED TOOLS

Digital images and with the help of some libraries:-

OpenCv, face recognition, numpy,

matplotlib and using python we can build a face recognition system.

LITERATURE SURVEY

This section is a basic overview of the major techniques used in the face recognition system that apply mostly to the front face of the human being.

The methods include neural networks, hidden Markov model, face matching done geometrically and template matching. Eigenface is one of the most widely used methods in face recognition and detection which are broadly called as the principle components in mathematical terms. The eigenvectors are ordered to represent different amounts of the variations in the faces. Neural networks are highly used in the face recognition and detection systems.

Template matching is one of the techniques through which the test image is represented

\
as a two- dimensional array of values which can be compared using Euclidean distance
with single template representing the whole face. This method can also use more than
one face template from different points of view to represent an individual face.

PROJECT DESIGN

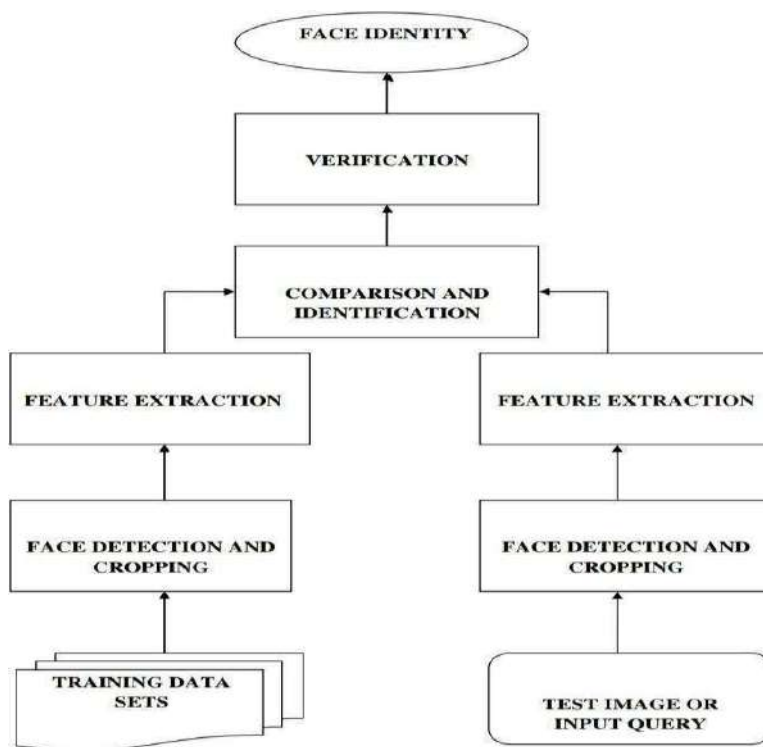


Fig . Architecture Diagram for Proposed Method

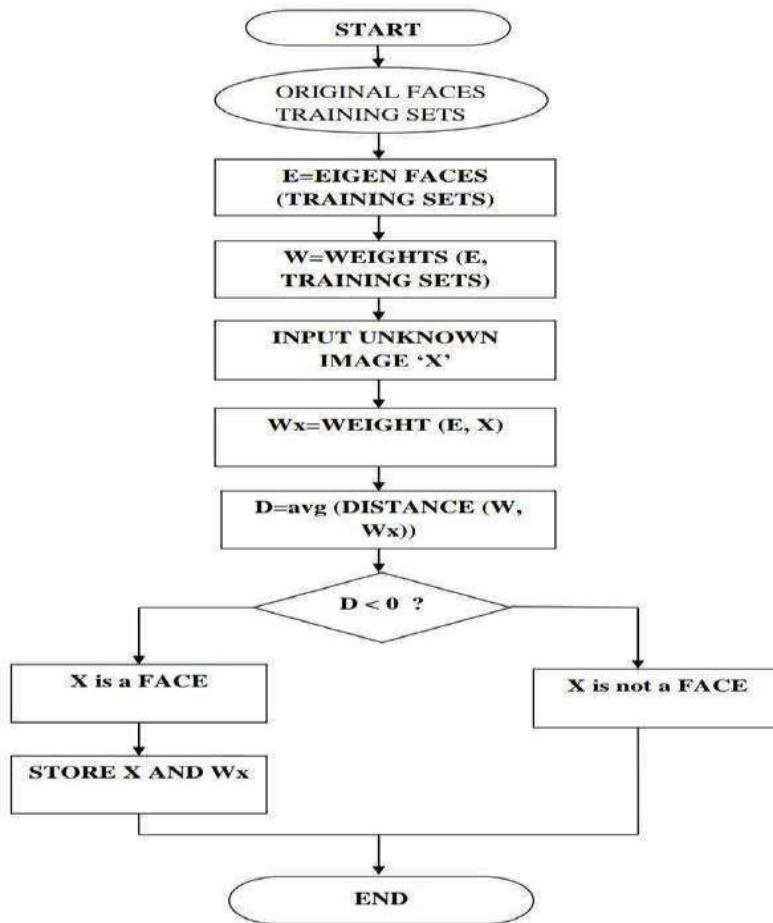


Fig. Eigen Face Based Algorithm Flow Chart

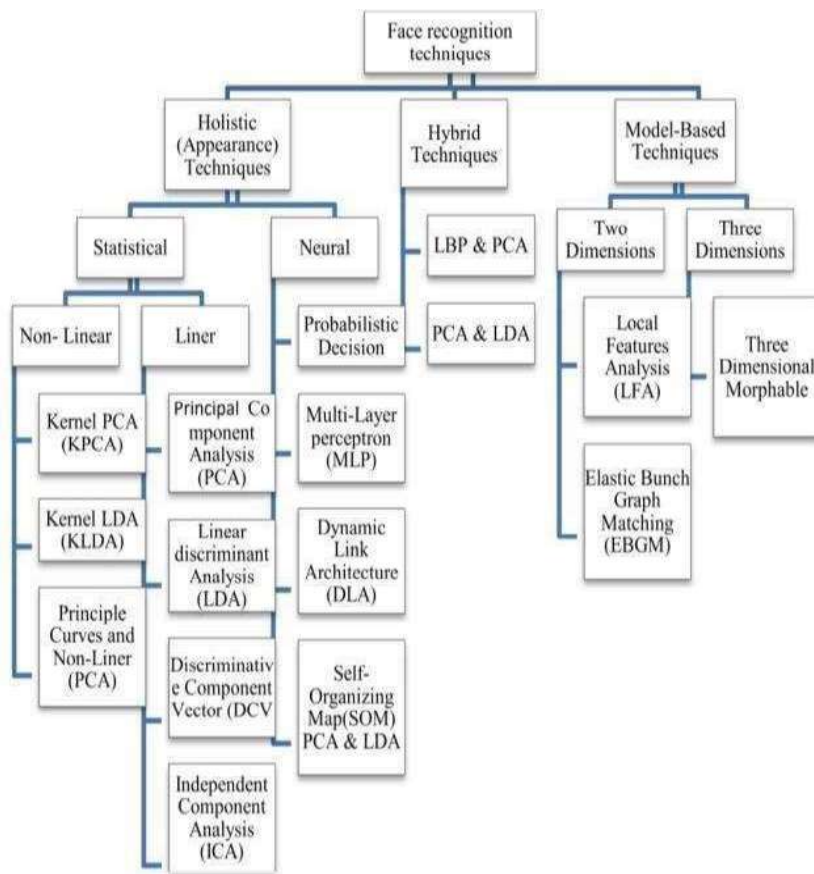


Fig. Face Recognition approaches

CODE AND ITS IMPLEMENTATION:-

```
# IMPORTING

import tkinter as tk

from tkinter import ttk

from tkinter import messagebox as mess

import tkinter.simpledialog as tsd

import cv2,os

import csv

import numpy as np

from PIL import Image

import pandas as pd

import datetime

import time
```

FUNCTIONS

```
def assure_path_exists(path):
```

```
\
dir = os.path.dirname(path)
```

```
if not os.path.exists(dir):
```

```
    os.makedirs(dir)
```

```
#####
```

```
def tick():
```

```
    time_string = time.strftime("%H:%M:%S")
```

```
    clock.config(text=time_string)
```

```
    clock.after(200,tick)
```

```
def contact():
```

```
    mess._show(title='Contact us', message="Please contact us on :
```

```
'shubhamkumar8180323@gmail.com' ")
```

```
def check_haarcascade():
```

```
    exists = os.path.isfile("haarcascade_frontalface_default.xml")
```

```
    if exists:
```

```
GROUP_ID: BT3141
```



```
\
    pass

else:

    mess._show(title='Some file missing', message='Please contact us for help')

    window.destroy()
```

```
def save_pass():

    assure_path_exists("TrainingImageLabel/")

    exists1 = os.path.isfile("TrainingImageLabel\psd.txt")

    if exists1:

        tf = open("TrainingImageLabel\psd.txt", "r")

        key = tf.read()

    else:

        master.destroy()

        new_pas = tsd.askstring('Old Password not found', 'Please enter a new password below',
show=*)

        if new_pas == None:

            mess._show(title='No Password Entered', message='Password not set!! Please try again')

        else:

            tf = open("TrainingImageLabel\psd.txt", "w")

            tf.write(new_pas)
```

```

\
    mess._show(title='Password Registered', message='New password was registered
successfully!!')

    return

op = (old.get())

newp= (new.get())

nnewp = (nnew.get())

if (op == key):

    if(newp == nnewp):

        txf = open("TrainingImageLabel\psd.txt", "w")

        txf.write(newp)

    else:

        mess._show(title='Error', message='Confirm new password!!!')

        return

else:

    mess._show(title='Wrong Password', message='Please enter correct old password.')

    return

mess._show(title='Password Changed', message='Password changed successfully!!')

master.destroy()

```

```

\
global master

master = tk.Tk()

master.geometry("400x160")

master.resizable(False,False)

master.title("Change Password")

master.configure(background="white")

lbl4 = tk.Label(master,text='  Enter Old Password',bg='white',font=('times', 12, ' bold '))

lbl4.place(x=10,y=10)

global old

old=tk.Entry(master,width=25 ,fg="black",relief='solid',font=('times', 12, ' bold '),show=('*'))

old.place(x=180,y=10)

lbl5 = tk.Label(master, text='  Enter New Password', bg='white', font=('times', 12, ' bold '))

lbl5.place(x=10, y=45)

global new

new = tk.Entry(master, width=25, fg="black",relief='solid', font=('times', 12, ' bold '),show=('*'))

new.place(x=180, y=45)

lbl6 = tk.Label(master, text='Confirm New Password', bg='white', font=('times', 12, ' bold '))

lbl6.place(x=10, y=80)

global nnew

nnew = tk.Entry(master, width=25, fg="black", relief='solid',font=('times', 12, ' bold '),show=('*'))

nnew.place(x=180, y=80)

cancel=tk.Button(master,text="Cancel", command=master.destroy ,fg="black" ,bg="red"

```

```

\
,height=1,width=25 , activebackground = "white" ,font=('times', 10, ' bold '))

cancel.place(x=200, y=120)

save1 = tk.Button(master, text="Save", command=save_pass, fg="black", bg="#3e4848", height
= 1,width=25, activebackground="white", font=('times', 10, ' bold '))

save1.place(x=10, y=120)

master.mainloop()

```

```
def psw():
```

```
    assure_path_exists("TrainingImageLabel/")
```

```
    exists1 = os.path.isfile("TrainingImageLabel\psd.txt")
```

```
    if exists1:
```

```
        tf = open("TrainingImageLabel\psd.txt", "r")
```

```
        key = tf.read()
```

```
    else:
```

```
        new_pas = tsd.askstring('Old Password not found', 'Please enter a new password below',
```

```
show='*')
```

```
        if new_pas == None:
```

```
            mess._show(title='No Password Entered', message='Password not set!! Please try again')
```

```
        else:
```

```
            tf = open("TrainingImageLabel\psd.txt", "w")
```

```

\
    tf.write(new_pas)

    mess._show(title='Password Registered', message='New password was registered
successfully!!')

    return

password = tsd.askstring('Password', 'Enter Password', show='*')

if (password == key):

    TrainImages()

elif (password == None):

    pass

else:

    mess._show(title='Wrong Password', message='You have entered wrong password')

```

```

def clear():

    txt.delete(0, 'end')

    res = "1)Take Images >>> 2)Save Profile"

    message1.configure(text=res)

```

```

def clear2():

    txt2.delete(0, 'end')

```

```
\
res = "1)Take Images >>> 2)Save Profile"
```

```
message1.configure(text=res)
```

```
def TakelImages():
```

```
    check_haarcascadefile()
```

```
    columns = ['SERIAL NO.', ", 'ID', ", 'NAME']
```

```
    assure_path_exists("StudentDetails/")
```

```
    assure_path_exists("TrainingImage/")
```

```
    serial = 0
```

```
    exists = os.path.isfile("StudentDetails\StudentDetails.csv")
```

```
    if exists:
```

```
        with open("StudentDetails\StudentDetails.csv", 'r') as csvFile1:
```

```
            reader1 = csv.reader(csvFile1)
```

```
            for l in reader1:
```

```
                serial = serial + 1
```

```
            serial = (serial // 2)
```

```
            csvFile1.close()
```

```
    else:
```

```
        with open("StudentDetails\StudentDetails.csv", 'a+') as csvFile1:
```

```
            writer = csv.writer(csvFile1)
```

```

\
    writer.writerow(columns)

    serial = 1

    csvFile1.close()

    Id = (txt.get())

    name = (txt2.get())

    if ((name.isalpha()) or (' ' in name)):

        cam = cv2.VideoCapture(0)

        harcascadePath = "haarcascade_frontalface_default.xml"

        detector = cv2.CascadeClassifier(harcascadePath)

        sampleNum = 0

        while (True):

            ret, img = cam.read()

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

            faces = detector.detectMultiScale(gray, 1.3, 5)

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

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

                # incrementing sample number

                sampleNum = sampleNum + 1

                # saving the captured face in the dataset folder TrainingImage

                cv2.imwrite("TrainingImage\ " + name + "." + str(serial) + "." + Id + '.' + str(sampleNum)

+ ".jpg",

                                gray[y:y + h, x:x + w])

```

```

\
    # display the frame

    cv2.imshow('Taking Images', img)

    # wait for 100 milliseconds

    if cv2.waitKey(100) & 0xFF == ord('q'):

        break

    # break if the sample number is morethan 100

    elif sampleNum > 100:

        break

cam.release()

cv2.destroyAllWindows()

res = "Images Taken for ID : " + Id

row = [serial, ", Id, ", name]

with open('StudentDetails\StudentDetails.csv', 'a+') as csvFile:

    writer = csv.writer(csvFile)

    writer.writerow(row)

csvFile.close()

message1.configure(text=res)

else:

    if (name.isalpha() == False):

        res = "Enter Correct name"

        message.configure(text=res)

```


\

```
def TrainImages():  
  
    check_harcascadefile()  
  
    assure_path_exists("TrainingImageLabel/")  
  
    recognizer = cv2.face_LBPHFaceRecognizer.create()  
  
    harcascadePath = "harcascade_frontalface_default.xml"  
  
    detector = cv2.CascadeClassifier(harcascadePath)  
  
    faces, ID = getImagesAndLabels("TrainingImage")  
  
    try:  
  
        recognizer.train(faces, np.array(ID))  
  
    except:  
  
        mess._show(title='No Registrations', message='Please Register someone first!!!')  
  
        return  
  
    recognizer.save("TrainingImageLabel\Trainer.yml")  
  
    res = "Profile Saved Successfully"  
  
    message1.configure(text=res)  
  
    message.configure(text='Total Registrations till now : ' + str(ID[0]))
```

```
def getImagesAndLabels(path):  
    GROUP_ID: BT3141
```

```

\
# get the path of all the files in the folder

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

# create empty face list

faces = []

# create empty ID list

Ids = []

# now looping through all the image paths and loading the Ids and the images

for imagePath in imagePaths:

    # loading the image and converting it to gray scale

    pillowImage = Image.open(imagePath).convert('L')

    # Now we are converting the PIL image into numpy array

    imageNp = np.array(pillowImage, 'uint8')

    # getting the Id from the image

    ID = int(os.path.split(imagePath)[-1].split(".")[1])

    # extract the face from the training image sample

    faces.append(imageNp)

    Ids.append(ID)

return faces, Ids

```

```

\
check_harcascadefile()

assure_path_exists("Attendance/")

assure_path_exists("StudentDetails/")

for k in tv.get_children():

    tv.delete(k)

msg = "

i = 0

j = 0

recognizer = cv2.face.LBPHFaceRecognizer_create() # cv2.createLBPHFaceRecognizer()

exists3 = os.path.isfile("TrainingImageLabel\Trainer.yml")

if exists3:

    recognizer.read("TrainingImageLabel\Trainer.yml")

else:

    mess._show(title='Data Missing', message='Please click on Save Profile to reset data!!')

    return

harcascadePath = "haarcascade_frontalface_default.xml"

faceCascade = cv2.CascadeClassifier(harcascadePath);

cam = cv2.VideoCapture(0)

font = cv2.FONT_HERSHEY_SIMPLEX

col_names = ['Id', ", 'Name', ", 'Date', ", 'Time']

exists1 = os.path.isfile("StudentDetails\StudentDetails.csv")

```

\
if exists1:

```
df = pd.read_csv("StudentDetails\StudentDetails.csv")
```

else:

```
mess._show(title='Details Missing', message='Students details are missing, please check!')
```

```
cam.release()
```

```
cv2.destroyAllWindows()
```

```
window.destroy()
```

while True:

```
ret, im = cam.read()
```

```
gray = cv2.cvtColor(im, cv2.COLOR_BGR2GRAY)
```

```
faces = faceCascade.detectMultiScale(gray, 1.2, 5)
```

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

```
cv2.rectangle(im, (x, y), (x + w, y + h), (225, 0, 0), 2)
```

```
serial, conf = recognizer.predict(gray[y:y + h, x:x + w])
```

```
if (conf < 50):
```

```
ts = time.time()
```

```
date = datetime.datetime.fromtimestamp(ts).strftime('%d-%m-%Y')
```

```
timeStamp = datetime.datetime.fromtimestamp(ts).strftime('%H:%M:%S')
```

```
aa = df.loc[df['SERIAL NO.']== serial]['NAME'].values
```

```
ID = df.loc[df['SERIAL NO.']== serial]['ID'].values
```

```
ID = str(ID)
```

```
ID = ID[1:-1]
```

```

\
    bb = str(aa)

    bb = bb[2:-2]

    attendance = [str(ID), ", ", bb, ", ", str(date), ", ", str(timeStamp)]

else:

    Id = 'Unknown'

    bb = str(Id)

    cv2.putText(im, str(bb), (x, y + h), font, 1, (255, 255, 255), 2)

cv2.imshow('Taking Attendance', im)

if (cv2.waitKey(1) == ord('q')):

    break

ts = time.time()

date = datetime.datetime.fromtimestamp(ts).strftime('%d-%m-%Y')

exists = os.path.isfile("Attendance\Attendance_" + date + ".csv")

if exists:

    with open("Attendance\Attendance_" + date + ".csv", 'a+') as csvFile1:

        writer = csv.writer(csvFile1)

        writer.writerow(attendance)

    csvFile1.close()

else:

    with open("Attendance\Attendance_" + date + ".csv", 'a+') as csvFile1:

        writer = csv.writer(csvFile1)

```

```

\
    writer.writerow(col_names)

    writer.writerow(attendance)

csvFile1.close()

with open("Attendance\Attendance_" + date + ".csv", 'r') as csvFile1:

    reader1 = csv.reader(csvFile1)

    for lines in reader1:

        i = i + 1

        if (i > 1):

            if (i % 2 != 0):

                iidd = str(lines[0]) + ' '

                tv.insert("", 0, text=iidd, values=(str(lines[2]), str(lines[4]), str(lines[6])))

csvFile1.close()

cam.release()

cv2.destroyAllWindows()

#USED STUFFS

global key

key = "

ts = time.time()

date = datetime.datetime.fromtimestamp(ts).strftime('%d-%m-%Y')
GROUP_ID: BT3141

```

```
\  
day,month,year=date.split("-")
```

```
mont={01:'January',  
      '02':'February',  
      '03':'March',  
      '04':'April',  
      '05':'May',  
      '06':'June',  
      '07':'July',  
      '08':'August',  
      '09':'September',  
      '10':'October',  
      '11':'November',  
      '12':'December'  
}
```

```
#GUI FRONT-END
```

```
window = tk.Tk()
```

```
window.geometry("1280x720")
```

```
window.resizable(True,False)
```

```
window.title("Attendance System")
```

```
GROUP_ID: BT3141
```

```

\
window.configure(background='#262523')

frame1 = tk.Frame(window, bg="#00aeff")
frame1.place(relx=0.11, rely=0.17, relwidth=0.39, relheight=0.80)

frame2 = tk.Frame(window, bg="#00aeff")
frame2.place(relx=0.51, rely=0.17, relwidth=0.38, relheight=0.80)

message3 = tk.Label(window, text="Face Recognition Based Attendance System"
,fg="white",bg="#262523" ,width=55 ,height=1,font=('times', 29, ' bold '))
message3.place(x=10, y=10)

frame3 = tk.Frame(window, bg="#c4c6ce")
frame3.place(relx=0.52, rely=0.09, relwidth=0.09, relheight=0.07)

frame4 = tk.Frame(window, bg="#c4c6ce")
frame4.place(relx=0.36, rely=0.09, relwidth=0.16, relheight=0.07)

datef = tk.Label(frame4, text = day+"-"+mont[month]+"-"+year+" | ", fg="orange",bg="#262523"
,width=55 ,height=1,font=('times', 22, ' bold '))
datef.pack(fill='both',expand=1)

```



```
clock = tk.Label(frame3,fg="orange",bg="#262523" ,width=55 ,height=1,font=('times', 22, ' bold '))
```

```
clock.pack(fill='both',expand=1)
```

```
tick()
```

```
head2 = tk.Label(frame2, text="                For New Registrations                ",
```

```
fg="black",bg="#3ece48" ,font=('times', 17, ' bold '))
```

```
head2.grid(row=0,column=0)
```

```
head1 = tk.Label(frame1, text="                For Already Registered                ",
```

```
fg="black",bg="#3ece48" ,font=('times', 17, ' bold '))
```

```
head1.place(x=0,y=0)
```

```
lbl = tk.Label(frame2, text="Enter ID",width=20 ,height=1 ,fg="black" ,bg="#00aeff"
```

```
,font=('times', 17, ' bold '))
```

```
lbl.place(x=80, y=55)
```

```
txt = tk.Entry(frame2,width=32 ,fg="black",font=('times', 15, ' bold '))
```

```
txt.place(x=30, y=88)
```

```
lbl2 = tk.Label(frame2, text="Enter Name",width=20 ,fg="black" ,bg="#00aeff" ,font=('times', 17,
```

```
' bold '))
```

```
lbl2.place(x=80, y=140)
```

```
\  
txt2 = tk.Entry(frame2,width=32 ,fg="black",font=('times', 15, ' bold ') )
```

```
txt2.place(x=30, y=173)
```

```
message1 = tk.Label(frame2, text="(1)Take Images >>> 2)Save Profile" ,bg="#00aeff" ,fg="black"  
,width=39 ,height=1, activebackground = "yellow" ,font=('times', 15, ' bold '))
```

```
message1.place(x=7, y=230)
```

```
message = tk.Label(frame2, text="" ,bg="#00aeff" ,fg="black" ,width=39,height=1,  
activebackground = "yellow" ,font=('times', 16, ' bold '))
```

```
message.place(x=7, y=450)
```

```
lbl3 = tk.Label(frame1, text="Attendance",width=20 ,fg="black" ,bg="#00aeff" ,height=1  
,font=('times', 17, ' bold '))
```

```
lbl3.place(x=100, y=115)
```

```
res=0
```

```
exists = os.path.isfile("StudentDetails\StudentDetails.csv")
```

```
if exists:
```

```
    with open("StudentDetails\StudentDetails.csv", 'r') as csvFile1:
```

```
        reader1 = csv.reader(csvFile1)
```

```
        for l in reader1:
```

```
GROUP_ID: BT3141
```

```

\
    res = res + 1

res = (res // 2) - 1

csvFile1.close()

else:

    res = 0

message.configure(text='Total Registrations till now : '+str(res))

#MENUBAR

menubar = tk.Menu(window,relief='ridge')

filemenu = tk.Menu(menubar,tearoff=0)

filemenu.add_command(label='Change Password', command = change_pass)

filemenu.add_command(label='Contact Us', command = contact)

filemenu.add_command(label='Exit',command = window.destroy)

menubar.add_cascade(label='Help',font=('times', 29, ' bold '),menu=filemenu)

#TREEVIEW ATTENDANCE TABLE

tv= ttk.Treeview(frame1,height =13,columns = ('name','date','time'))

tv.column('#0',width=82)

tv.column('name',width=130)

tv.column('date',width=133)

```

```

\
tv.column('time',width=133)

tv.grid(row=2,column=0,padx=(0,0),pady=(150,0),columnspan=4)

tv.heading('#0',text ='ID')

tv.heading('name',text ='NAME')

tv.heading('date',text ='DATE')

tv.heading('time',text ='TIME')

```

#SCROLLBAR

```

scroll=ttk.Scrollbar(frame1,orient='vertical',command=tv.yview)

scroll.grid(row=2,column=4,padx=(0,100),pady=(150,0),sticky='ns')

tv.configure(yscrollcommand=scroll.set)

```

#BUTTONS

```

clearButton = tk.Button(frame2, text="Clear", command=clear ,fg="black" ,bg="#ea2a2a"
,width=11 ,activebackground = "white" ,font=('times', 11, ' bold '))

clearButton.place(x=335, y=86)

clearButton2 = tk.Button(frame2, text="Clear", command=clear2 ,fg="black" ,bg="#ea2a2a"
,width=11 , activebackground = "white" ,font=('times', 11, ' bold '))

clearButton2.place(x=335, y=172)

```

```

takeImg = tk.Button(frame2, text="Take Images", command=TakeImages ,fg="white" ,bg="blue"

```

```
,width=34 ,height=1, activebackground = "white" ,font=('times', 15, ' bold '))

takeImg.place(x=30, y=300)

trainImg = tk.Button(frame2, text="Save Profile", command=psw ,fg="white" ,bg="blue"
,width=34 ,height=1, activebackground = "white" ,font=('times', 15, ' bold '))

trainImg.place(x=30, y=380)

trackImg = tk.Button(frame1, text="Take Attendance", command=TrackImages ,fg="black"
,bg="yellow" ,width=35 ,height=1, activebackground = "white" ,font=('times', 15, ' bold '))

trackImg.place(x=30,y=50)

quitWindow = tk.Button(frame1, text="Quit", command=window.destroy ,fg="black" ,bg="red"
,width=35 ,height=1, activebackground = "white" ,font=('times', 15, ' bold '))

quitWindow.place(x=30, y=450)

#END

window.configure(menu=menubar)

window.mainloop()
```

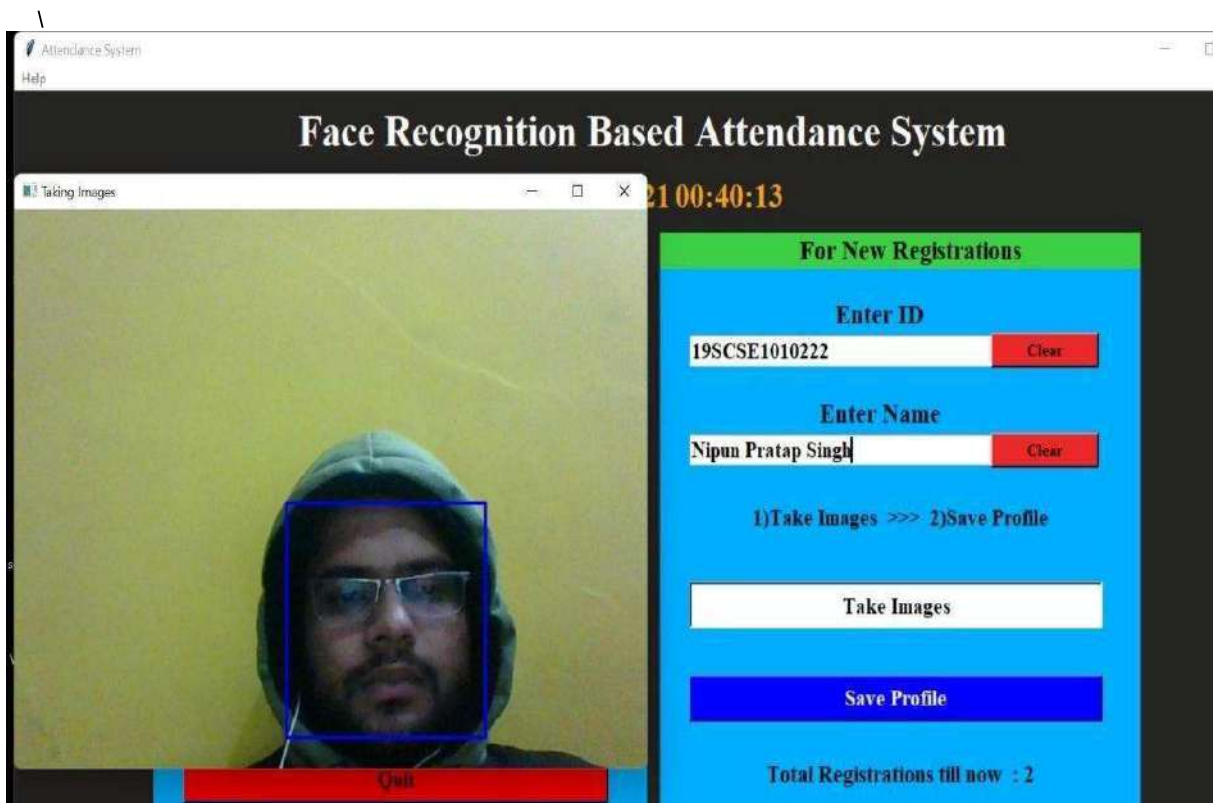
\

OUTPUT:-

INSERTION OF THE IMAGE:-

PERSON:-1

IMAGE IDENTIFICATION:-





RECORD OUR STATUS:-



RECORD IN CSV FILE:-

GROUP_ID: BT3141

Attendance_21-12-2021 - Excel

File Home Insert Page Layout Formulas Data Review View Help Tell me what you want to do

Clipboard Font Alignment Number Styles Cells Editing

Calibri 11 A A

B I U - - - - -

General

Conditional Formatting Format as Table Cell Styles

Insert Delete Format

Autosum Fill Sort & Filter Select

Share

A1

1	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
1	Id	Name	Date			Time																	
2																							
3	19SCSE1010245	Shubham Singh	#####			21:17:28																	
4																							
5	19SCSE1010222	Nipun Prata Singh	#####			21:17:42																	
6																							
7																							
8																							
9																							
10																							
11																							
12																							
13																							
14																							
15																							
16																							
17																							
18																							
19																							
20																							
21																							
22																							
23																							
24																							
25																							
26																							
27																							

Attendance_21-12-2021

CONCLUSION

1.) *The first conclusion we conclude is related to cost-Generally we used face recognition less system in very expensive top secure systems.*

2.) *Some factors like environmental changes and mild changes in appearance due to age or an incidence happen impact the technology to a greater degree than many expect.*

3.) *This one of the fast and convenient method of biometric identification and has greater*

\
use in society.

4.) *It will find the faces with exhaustively wasting time in searching of images.*

5.) *This system has widespread application in the smart environment.*

THANK YOU!!