PROJECT REPORT

on

Face Recognition Attendance System (CSE V Semester Mini Project)

2021-2022

Submitted to:

Mr. Akash Chauhan

(CC-CSE-K-V-Sem)

Guided by:

Mr. Chandradeep

(Resource Person)

Submitted by:

Ajay Negi

Roll. No.: 1918851

CSE-K-V-Sem

Session: 2021-2022

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

GRAPHIC ERA HILL UNVERSITY, DEHRADUN

CERTIFICATE

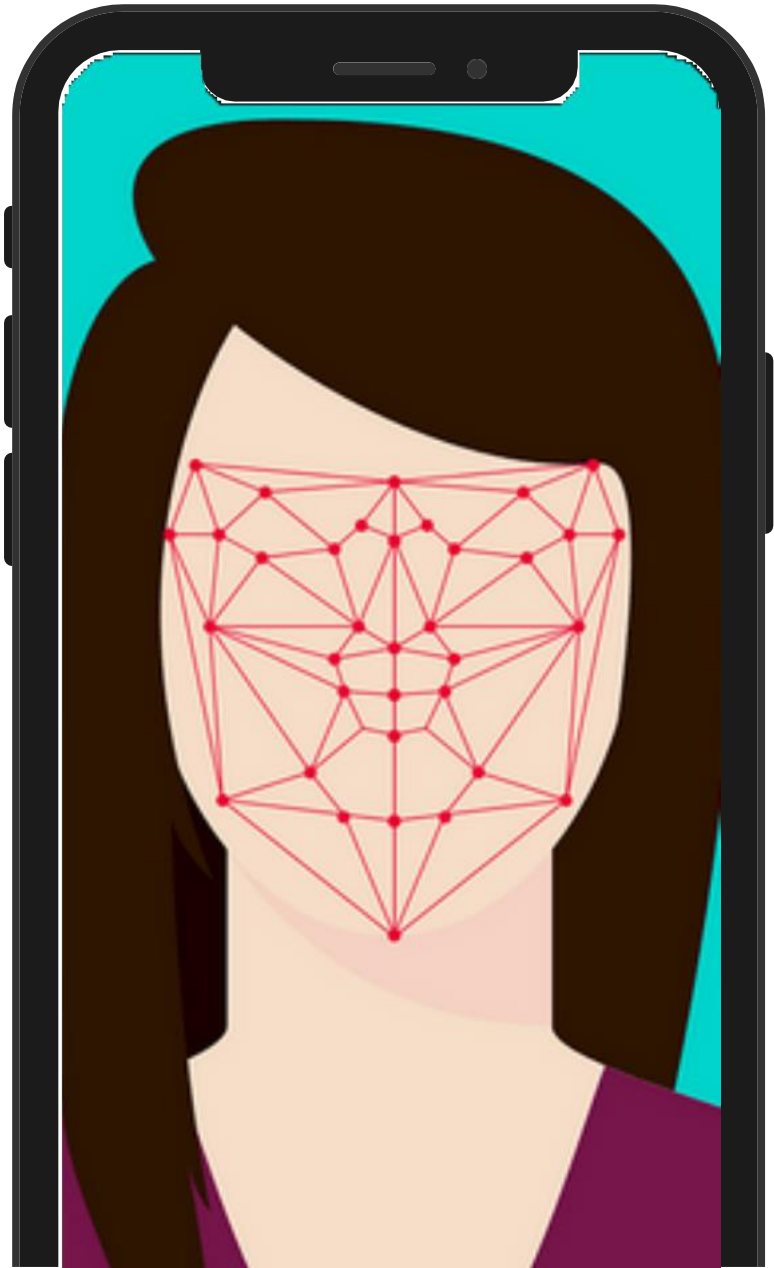
Certified that Ajay Negi (Roll No.- 1918851) has developed mini project on “Face Recognition Attendance System” for the CS V Semester Mini Project Lab (CSP-501) in Graphic Era Hill University, Dehradun. The project carried out by Students is their own work as best of my knowledge.

Date: 17-12-2021

(Mr.Akash Chauhan)	(Mr. Chandradeep)
Project Co-ordinator	**Project Guide**
CC-CSE-K-V-Sem	**Resource Person**
(CSE Department)	(CSE Department)
GEHU Dehradun	GEHU Dehradun

What is a Face recognition system?

A facial recognition system is a technology capable of matching a human face from a digital image or a video frame against a database of faces, typically employed to authenticate users through ID verification services, works by pinpointing and measuring facial features from a given image.



ALGORITHM

Encoding Faces

This algorithm uses Deep Convolutional Neural Network. But instead of training the network to recognize pictures objects, the model will generate 128 measurements for each face.

In Machine learning, 128 measurements of each face are called **embedding**. The idea of reducing complicated raw data like a picture into a list of computer-generated numbers comes up a lot in machine learning.

```
[-0.16597123  0.11949642  0.10122138 -0.12254387 -0.11838274 -0.02759848
  0.01384261 -0.07288168  0.06527785 -0.13963827  0.17559825 -0.08519795
 -0.25020906 -0.0102928  -0.01079375  0.21948609 -0.21553843 -0.19072869
 -0.13214052 -0.10856642  0.01477439  0.07908501 -0.05431245  0.11583221
 -0.1715932  -0.23868942 -0.10927209 -0.01475592  0.01510147 -0.08837818
 -0.00252565  0.03029939 -0.17696191  0.00717495  0.04937807  0.12862331
 -0.01091681 -0.09640069  0.1934087   0.01389313 -0.23863317  0.01766105
  0.15954995  0.26412117  0.21739453 -0.00336392 -0.03702538 -0.05539587
  0.10163558 -0.28075284  0.05129996  0.20316276  0.02803321  0.16820049
 -0.02042791 -0.12237336 -0.00515839  0.22946236 -0.21929839  0.03012473
  0.10334508 -0.19385433  0.010464   -0.1046315   0.159848    0.14616895
 -0.19181257 -0.22495562  0.19514428 -0.23470432 -0.11755507  0.07461189
 -0.0839375  -0.09592333 -0.35081255  0.03915626  0.32692659  0.20959754
 -0.10889891 -0.01162777 -0.09144997 -0.02747694  0.00839157  0.15324235
 -0.02081387 -0.0425766  -0.06370837  0.0559925   0.19876704  0.04837478
  0.00898859  0.25267416  0.04914283  0.0009048   0.00353496  0.14037418
 -0.15166469 -0.07702    -0.10442869  0.03603458 -0.10455432 -0.05006842
  0.01399595  0.18746354 -0.10759938  0.24044091 -0.03882578 -0.01190661
 -0.12068785  0.03638472 -0.02094681 -0.02857181  0.10984001 -0.22919109
  0.14735889  0.20415246 -0.00772358  0.09734803  0.10852756  0.12523061
 -0.03349054 -0.08732443 -0.17586364 -0.09488015  0.04729999 -0.0654182
  0.10575557  0.07663471]
```

Libraries Used

OpenCV

OpenCV is a library of programming functions mainly aimed at real-time computer vision.

NumPy

NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.

Face_recognition

Recognize and manipulate faces from Python or from the command line with the world's simplest face recognition library.
Built using dlib's state-of-the-art face recognition built with deep learning. The model has an accuracy of 99.38% on the Labeled Faces in the Wild benchmark.

Datetime

The datetime module supplies classes for manipulating dates and times.

OS

This module provides a portable way of using operating system dependent functionality.

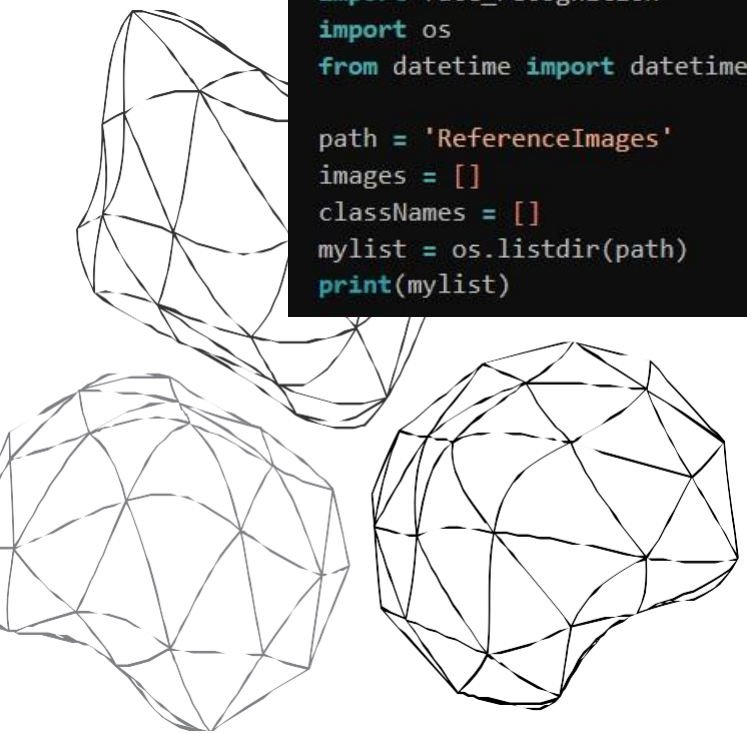
Set the path ofthe database

After importing libraries I have created a folder named ReferenceImages in which I have stored the images which will be used to take measurements.

In the folder ReferenceImages I have stored images.

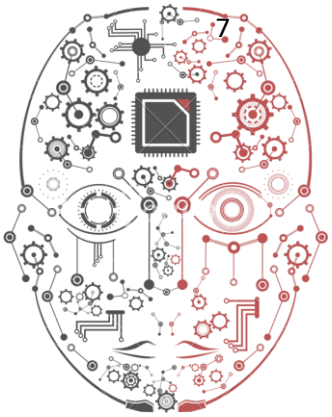
There are two lists defined here which will store the file name and the name of the person in the image.

```
import cv2
import numpy as np
import face_recognition
import os
from datetime import datetime

path = 'ReferenceImages'
images = []
classNames = []
mylist = os.listdir(path)
print(mylist)
```



Finding Encodings

MAKINGAFUNCTIONTOENCODEFACES



In the function findEncodings(images) first, we are converting the BGR image to RGB .

```
def findEncodings(images):
    encodelist = []
    for img in images:
        img = cv2.cvtColor(img,cv2.COLOR_BGR2RGB)
        encode = face_recognition.face_encodings(img)[0]
        encodelist.append(encode)
    return encodelist
```

This function will also return a list of 128-dimensional face encodings (one for each face in the image) which is named encodelist.
After executing this function, the face encoding is completed.

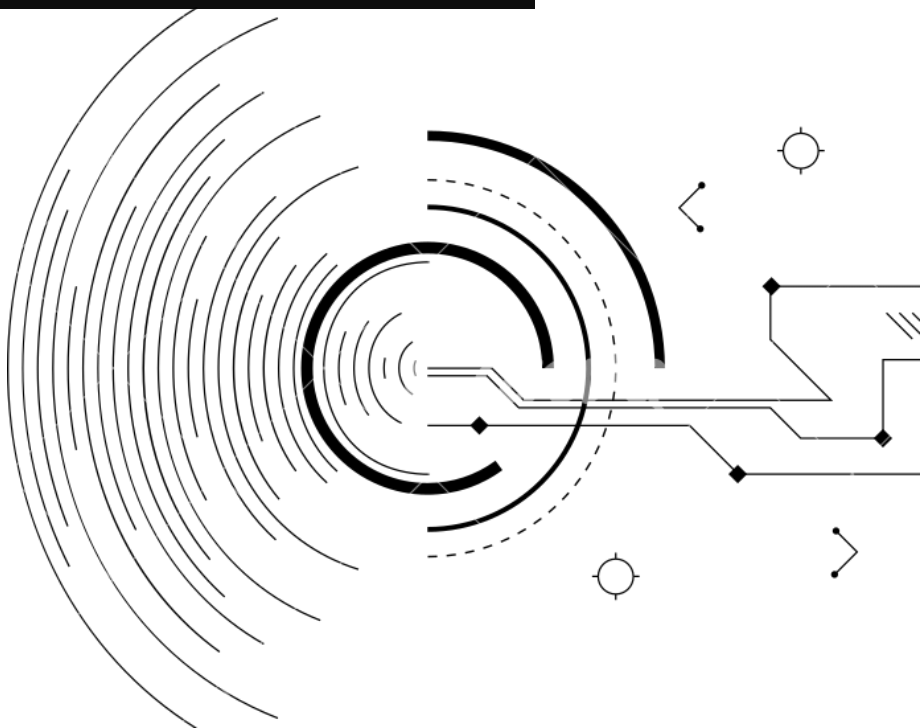
```
[-0.16597123  0.11949642  0.10122138 -0.12254387 -0.11838274 -0.02759848
  0.01384261 -0.07288168  0.06527785 -0.13963827  0.17559825 -0.08519795
 -0.25020906 -0.0102928  -0.01079375  0.21948609 -0.21553843 -0.19072869
 -0.13214052 -0.10856642  0.01477439  0.07908501 -0.05431245  0.11583221
 -0.1715932  -0.23868942 -0.10927209 -0.01475592  0.01510147 -0.08837818
 -0.00252565  0.03029939 -0.17696191  0.00717495  0.04937807  0.12862331
 -0.01091681 -0.09640069  0.1934087   0.01389313 -0.23863317  0.01766105
  0.15954995  0.26412117  0.21739453 -0.00336392 -0.03702538 -0.05539587
  0.10163558 -0.28075284  0.05129996  0.20316276  0.02803321  0.16820049
 -0.02042791 -0.12237336 -0.00515839  0.22946236 -0.21929839  0.03012473
  0.10334508 -0.19385433  0.010464   -0.1046315   0.159848    0.14616895
 -0.19181257 -0.22495562  0.19514428 -0.23470432 -0.11755507  0.07461189
 -0.0839375  -0.09592333 -0.35081255  0.03915626  0.32692659  0.20959754
 -0.10889891 -0.01162777 -0.09144997 -0.02747694  0.00839157  0.15324235
 -0.02081387 -0.0425766  -0.06370837  0.0559925   0.19876704  0.04837478
  0.00898859  0.25267416  0.04914283  0.0009048   0.00353496  0.14037418
 -0.15166469 -0.07702    -0.10442869  0.03603458 -0.10455432 -0.05006842
  0.01399595  0.18746354 -0.10759938  0.24044091 -0.03882578 -0.01190661
 -0.12068785  0.03638472 -0.02094681 -0.02857181  0.10984001 -0.22919109
  0.14735889  0.20415246 -0.00772358  0.09734803  0.10852756  0.12523061
 -0.03349054 -0.08732443 -0.17586364 -0.09488015  0.04729999 -0.0654182
  0.10575557  0.07663471]
```

Insertion of name and time in a csv file

CREATING A FUNCTION MARK ATTENDANCE IN A CSV FILE

- Creating a .csv file and opening it in both readand write mode.
- Creating a list namelist which will get the names.
- The names will be checked and if they are not present in the namelist then, that name will be added in the .csv file with the time at which the face is recognised.

```
def markAttendence(name):
    with open('Attendence.csv','r+') as f:
        myDatalist = f.readlines()
        namelist = []
        for line in myDatalist:
            entry = line.split(',')
            namelist.append(entry[0])
        if name not in namelist:
            now = datetime.now()
            dtString = now.strftime("%H-%M:%S")
            f.writelines(f'\n{name},{dtString}')
```



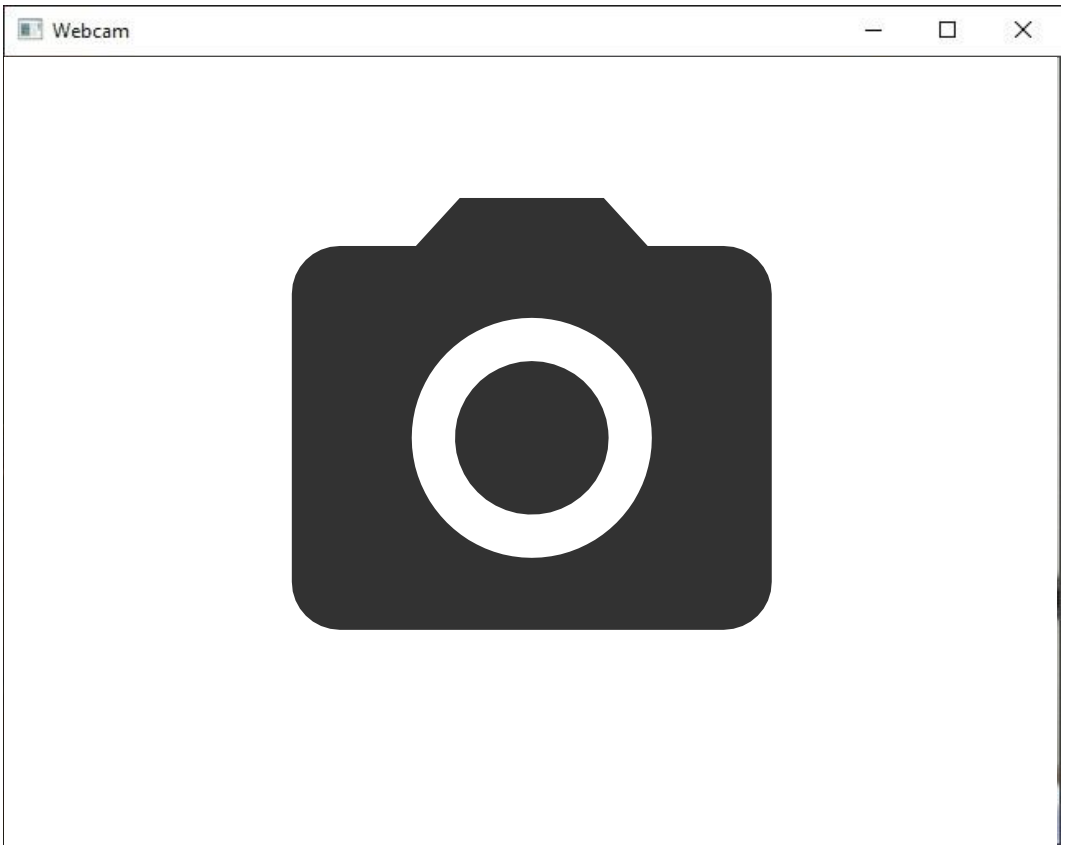
Opening the camera

WITHTHEHELPOFOPENCV

I have declared a variable named cap that will capture the images from the camera, from which we will be able to get the test images.

```
cap = cv2.VideoCapture(0)
```

The camera looks like this:



Reading the faces

Now we will read the frames from the camera with the help of the variable cap.

We'll resize the image in a square of dimensions (0.25x0.25) and also covert the BGR image into RBG

We'll also do the following things: Find the

- face location.
- Compare the encodings of the images in the frame and in the database.
- Find the least value of encodings in the array facedis.

CHAINGTHESIZEANDCOLOUR

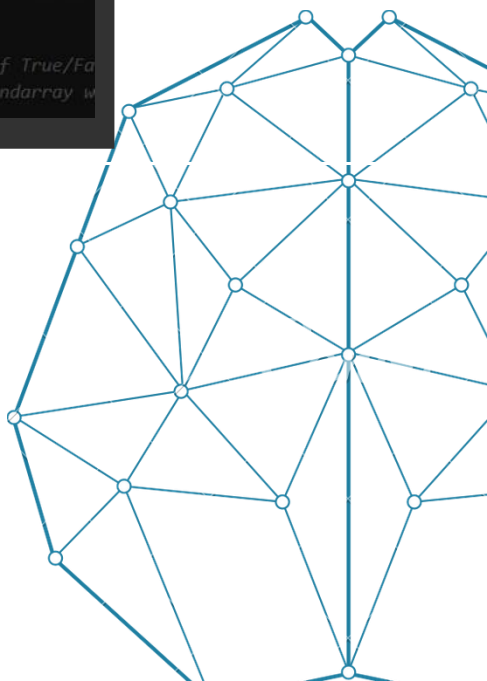
```
while True:
    sucess ,img = cap.read()
    imgS = cv2.resize(img,(0,0),None,0.25,0.25)
    imgS = cv2.cvtColor(imgS,cv2.COLOR_BGR2RGB)
```

ENCODINGSOFTHEIMAGES

```
faces_in_current_frame = face_recognition.face_locations(imgS)  #A list of tuples of found fac

encodes_current_frame = face_recognition.face_encodings(imgS,faces_in_current_frame)    #A lis

for encodeFaces , faceloc in zip(encodes_current_frame,faces_in_current_frame):
    matches = face_recognition.compare_faces(encodeListKnown,encodeFaces)    #A list of True/Fa
    facedis = face_recognition.face_distance(encodeListKnown,encodeFaces)    #A numpy ndarray w
```

MATCHING THE FACE DISTANCE

```
matchIndex =np.argmin(facedis)

if matches[matchIndex]:
    name = classNames[matchIndex].upper()
```



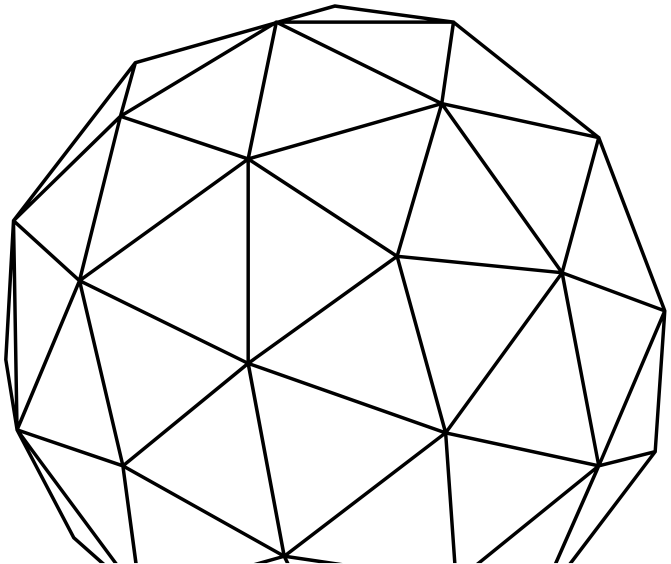
Conclusion

Face recognition technology is a very useful technology and has vast application in the real world. It can be used in an AI-enabled authentication scheme in IoT like a smart lock, smart homes, smart attendance system and many more.

Law enforcement agencies use facial recognition to find missing people, and they've also used it to find missing children. When combined with ageing software that shows how the child would look several years later, facial recognition can even help find someone who's been missing for years.

As facial recognition technology improves, its challenges will decrease. Other technology could impact its effectiveness, including recognizing body parts or how a person walks.

For the time being, though, the technology's inadequacies and people's reliance on it means facial recognition has room to grow and improve.



Reference

- www.google.com
  www.stakoverflow.com
- www.youtube.com
- www.wikipedia.com
  www.medium.com
- https://docs.python.org

- https://face-recognition.readthedocs.io/