PROJECT EXHIBITION
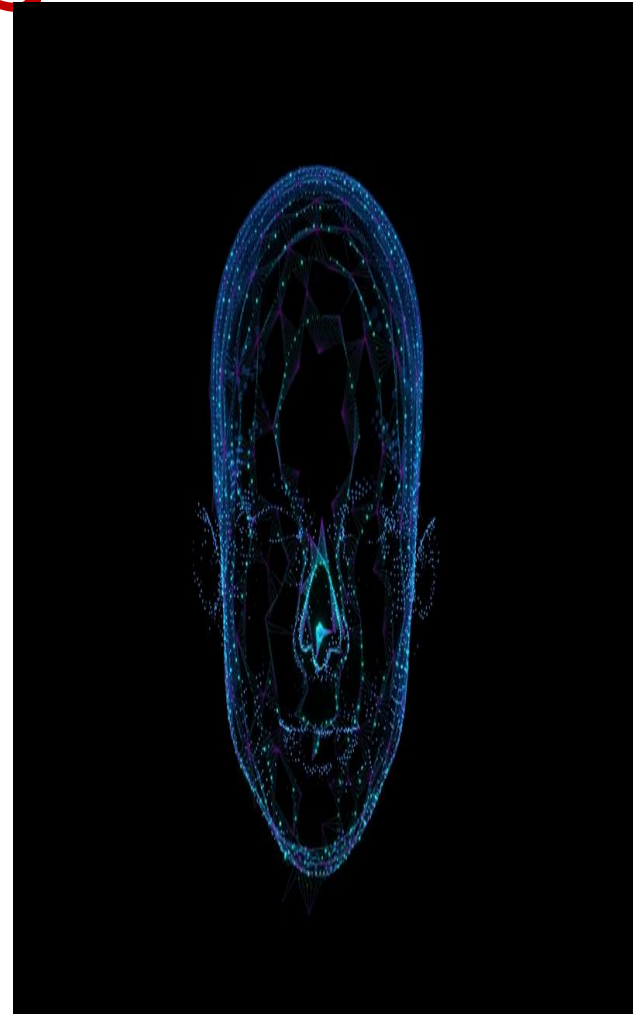
Project Administrator:
Dr. Pandi Murugan

GROUP MEMBERS:

- ❖ MANSI SAINI
  19BCE10053

- ❖ SANJANA SINGH
  19BCE10274

- ❖ RACHITA JHA
  19BCE10283

# TOPIC:

3-D vision attendance system

INTRODUCTION

- Face recognition is a passive, non-invasive method for verifying the identity of a person.

- Face Recognition is a way of identifying a human face through a picture or video stream.

- A facial recognition system uses technology to recognize a face.

- Every face has different facial features, different shapes, different peaks, and valleys. These are known as nodal points and every face has approximately 80 nodal points.

15

2019/07/15

FACE RECOGNITION

Major Objectives

- Save Images To Database
- Detect Faces
- Match detected faces to Database
- Recognize Faces
- Provides accurate information about them

Image → Face Detection → Feature Extraction → Face Recognition → Verification/Identification
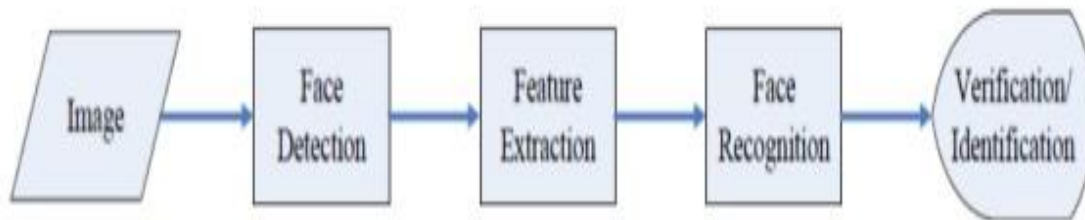
The objective of this project is to provide enhanced security by implementing a simple face recognition system.

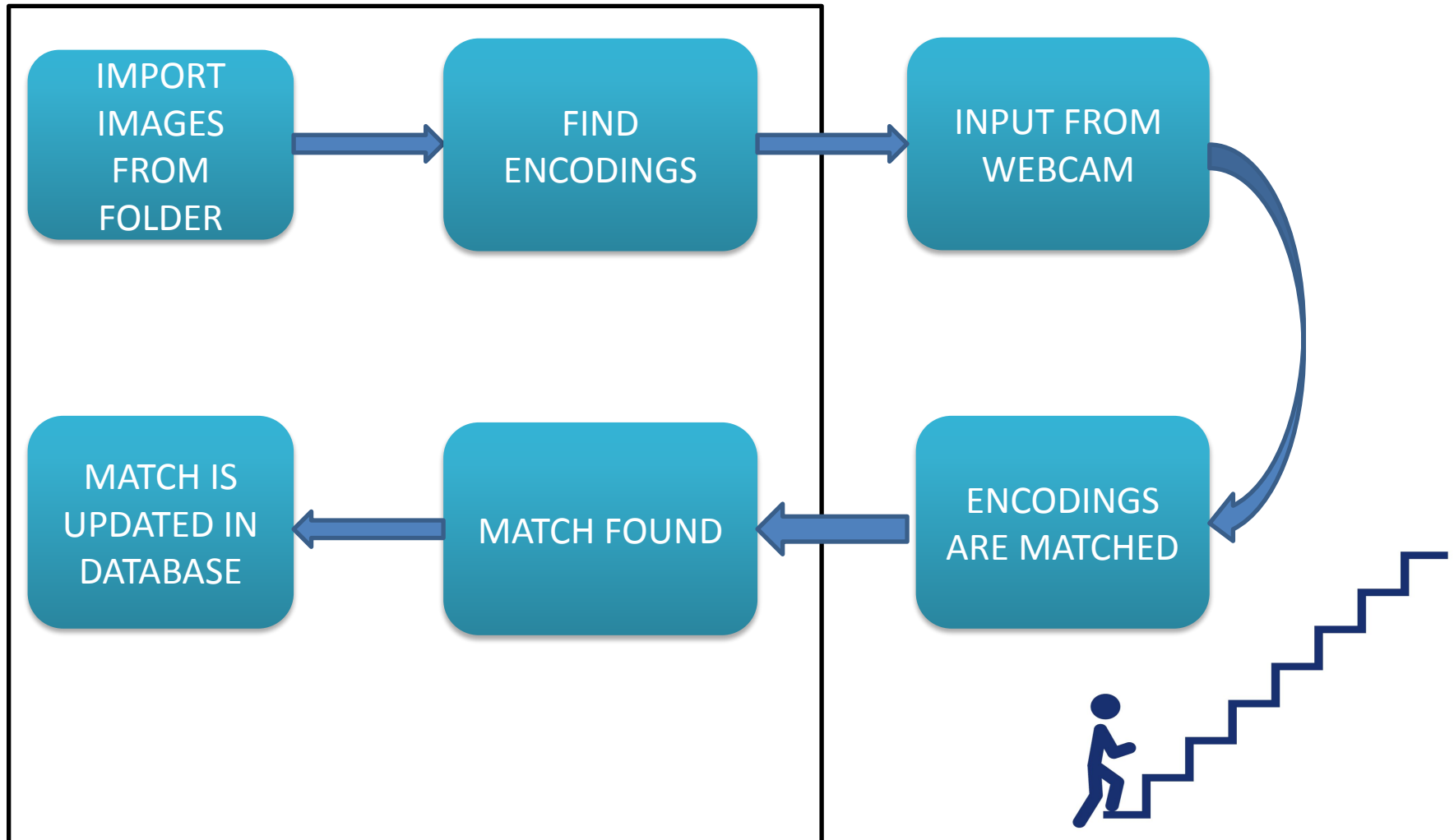Another main objective is to use this technology for taking attendance.

Existing Work with Limitations

- **Poor Image Quality Limits Facial Recognition's Effectiveness-** The quality of the reference image plays an important role in the identification process. If the resolution of the said image is not high enough, it can cause cameras to be tricked into believing that the person being scanned is not the same as in the photo.

- **Small Image Sizes Make Facial Recognition More Difficult-** Small size of image can sometimes create a problem in detection process as the relative size of the face as compared with the enrolled image size affects how well the face will be recognized.

- **Different Face Angles Can Throw Off Facial Recognition's Reliability-** Many non-premium facial recognition systems cannot account for faces that are captured at angles other than straight into the capturing camera. The disadvantage of this is that it makes the attendance marking process slower and less efficient.

- **Data Processing and Storage Can Limit Facial Recognition Technology-** Depending on the quality of the input data, a system would need an appropriate amount of storage. This could be troublesome if the data collected is of high quality and requires large amounts of storage space especially for events with a large expected attendance.

BACKEND
Steps followed to solve the defined problem

```
┌─────────────┐      ┌─────────────┐      ┌─────────────┐
│ IMPORT      │      │             │      │ INPUT FROM  │
│ IMAGES      │ ───► │ FIND        │ ───► │ WEBCAM      │
│ FROM        │      │ ENCODINGS   │      │             │
│ FOLDER      │      │             │      │             │
└─────────────┘      └─────────────┘      └─────────────┘

┌─────────────┐      ┌─────────────┐      ┌─────────────┐
│ MATCH IS    │      │             │      │ ENCODINGS   │
│ UPDATED IN  │ ◄─── │ MATCH FOUND │ ◄─── │ ARE MATCHED │
│ DATABASE    │      │             │      │             │
└─────────────┘      └─────────────┘      └─────────────┘
```

Proposed System

There are two parts- **Enrollment and recognition**. In enrolment, the students have to register themselves by entering details. In pre-processing phase, detection of faces, extraction of the features, detection of extracted features will be processed and will be stored in data base. And in recognition phase, after the detection of extracted features, a canonical form of the facial surface is computed and recognition itself is performed. Faces will be detected from live video/photo of people present there and then

Novelty of this project

- **The main novelty of this approach is the ability to compare surfaces independent of natural deformations resulting from facial expressions.**

- **First, the range image and the texture of the face are acquired.**

- **Next, the range image is preprocessed by removing certain parts such as hair, which can complicate the recognition process.**

- **Finally, a canonical form of the facial surface is computed and recognition itself is performed on the canonical surfaces.**

Benefits of this project

- **Ease of enrollment**
- **Fast and Non-Invasive Identity Verification**
- **Better Worker Attendance Systems**
- **Face Recognition Furthers Enables Computer Vision**
- **High accuracy**
- **Suitable for large or small databases**
- **One-to-many searching**
- **Compatible with SQL server.**
- **Database of 5000 Indian and with other biometrics is completed**

REAL TIME USAGE

- Face recognition system is widely used to identify passports and driver's licenses carrying individuals even if they are not aware that a face recognition system is autonomously checking their identity.

- The system is being used to maintain discipline in the schools and is also helpful in detecting any kind of threats to the school.

- It is being used for identification of rare genetic diseases, also the symptoms for different disorders can be recognized and prescription is suggested accordingly.

- The new vehicles can predict when a driver is tired or falling asleep and can act if an accident could happen with the help of this system.

- Facial Recognition will be used at the entrance for security purposes instead of relying on ID cards for making an entry.

- AI-powered cameras are also being used for catching shoplifters, verifying bank employees and others.

HARDWARE REQUIREMENTS

- **6 GB RAM (Minimum)**
- **80 GB HDD (Hybrid Hard Drive)**
- **Dual Core processor**
- **(Compact disc read-only memory)CDROM (installation only)**
- **VGA(Video Graphics Array) resolution monitor**
- **Microsoft Windows 10**
- **SQL Server 2019. RTM version is 15.0. 2000.5**

Libraries to be used-

- **OpenCV** : OpenCV is a huge open-source library for computer vision, machine learning, and image processing. It can process images and videos to identify objects, faces, or even the handwriting of a human.

- **NumPy** : NumPy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays. It is the fundamental package for scientific computing with Python.

- **CMake** : CMake is used to control the software compilation process using simple platform and compiler independent configuration files, and generate native make files and workspaces that can be used in the compiler environment of your choice.

- **Dlib** : A toolkit for making real world machine learning and data analysis applications.Dlib is a general purpose cross-platform software library written in the programming language C++. Its design is heavily influenced by ideas from design by contract and component-based software engineering. Thus it is, first and foremost, a set of independent software components.

- **DateandTime** : This package provides a Date Time data type, as known from Zope. Unless you need to communicate with Zope APIs, you're probably better off using Python's built-in datetime module.

- **Face –recognition** : The face
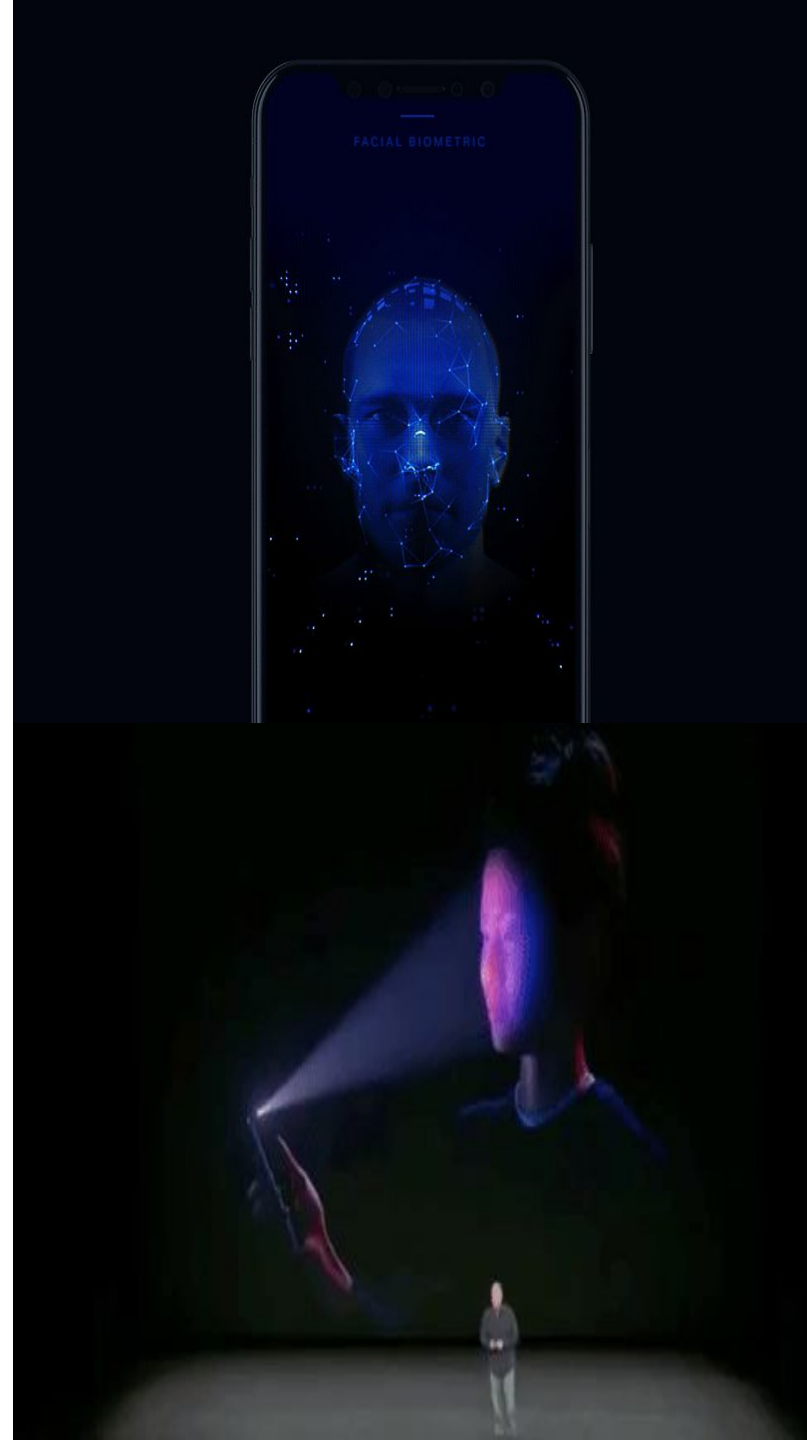
SYSTEM ACHITECTURE DIAGRAM
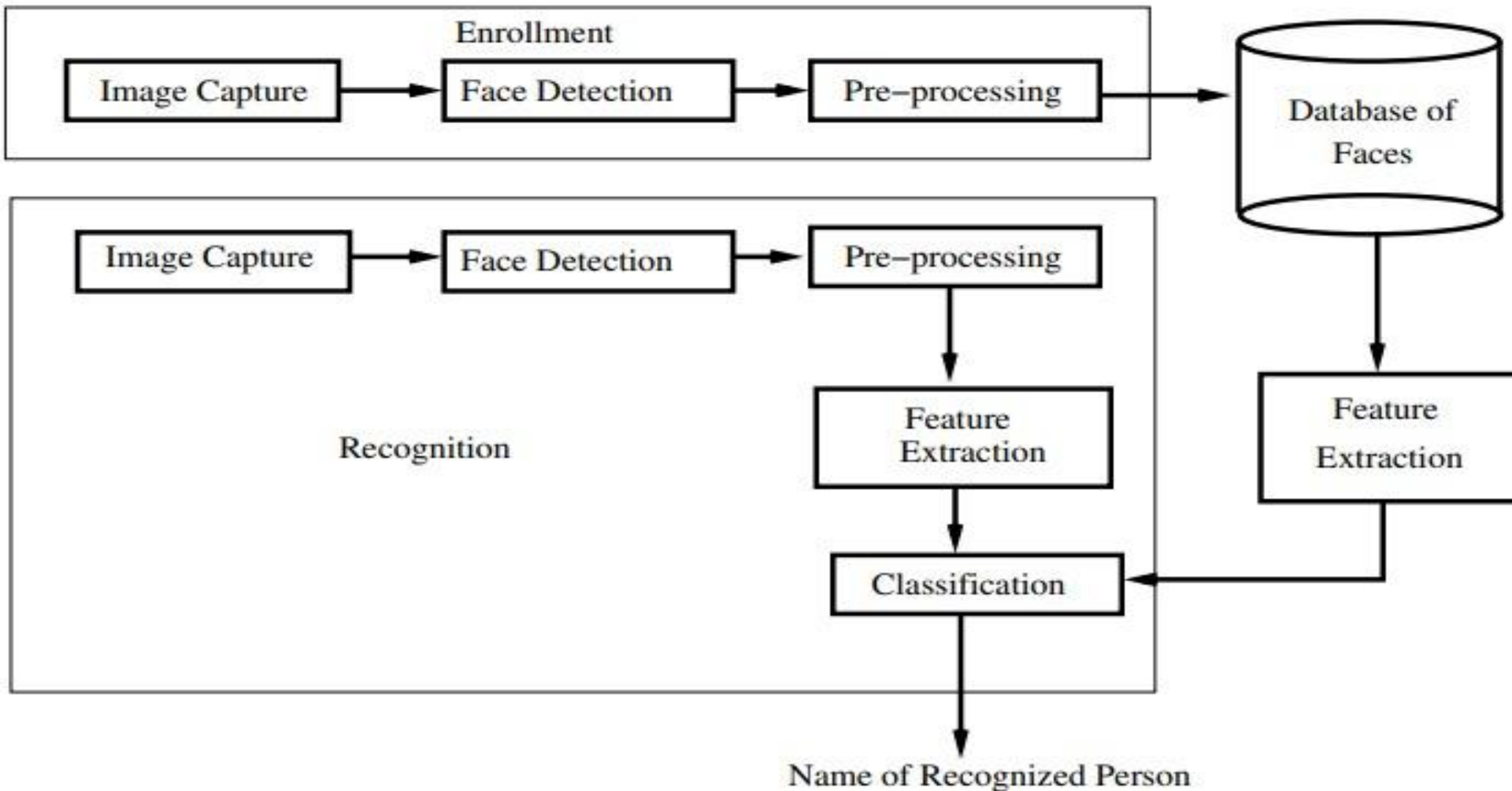
Literature Review

Need of the system-

- As face provides a unique identity of a person, it can be used to identify a person and verify his/her identity.

- This system that we have developed has successfully able to accomplish the task of marking the attendance in the classroom automatically and output obtained in an excel sheet as desired in real time.

- Another important aspect where we can work is towards creating an online database of the attendance and its automatic updating.

- Face-recognition provides non-intrusive way to recognize a person.

- By using this system, the chances of fake attendance and proxies can be reduced.

Literature Review

My Understanding of the project-

Encode a picture using the HOG algorithm to create a simplified version of the image. Using this simplified image, find the part of the image that most looks like a generic HOG encoding of a face. Figure out the pose of the face by finding the main landmarks in the face. Once we find those landmarks, use them to warp the image so that the eyes and mouth are centered. Pass the centered face image through a neural network that knows how to measure features of the face. Save those 128 measurements. Looking at all the faces we've measured in the past, see which person has the closest measurements to our face's measurements and that will be our match. Best match is updated in our database with name of the person and time of joining.

MODULE DESCRIPTION

Maintaining attendance record of each and every person is a challenging task. Calling name of each student is time consuming and cumbersome to maintain and there is always a chance of proxy attendance. From schools to colleges, offices, IT companies, shops, face recognition is one of the mostly used biometrics, security, authentication and identification to reduce the dependency on ID cards only and to minimize proxy system. It has more advantages than fingerprint system and iris recognition as it is contactless, passive and non-invasive process. This system can also be used for attendance marking in schools, colleges, offices, etc. This project aims to build a class attendance system that uses the face recognition concept. To minimize the situations of proxy, the need for this system increases. Smart Attendance using Real-Time Face Recognition is a real-world solution of handling student attendance system This system consists of four modules- database creation, face detection, face recognition and attendance updation on excel sheet. Database contains the images of the students in class. It has used SVM classifier. Faces are detected and recognized from live streaming or movement of video/ photo excluding certain parts such as hair, which can complicate the recognition process. After the entry in excel sheet along with time, attendance can be mailed to the respective higher authorities.

MODULE WORK

- **Database creation**

- **Face Recognition**

- **Face Detection**

- **Attendance system**

Database Creation

First step is to create the database containing the pictures of the people whose attendance has to be taken. The power of modifying the folder/database is solely in the hands of the authority. Database is open to further future updations.

Face Detection

With the help of powerful libraries like open-cv, face-recognition etc. the face of the attendees from training images in database is detected through face landmarks and their encodings/embeddings is taken through inbuilt methods. Similar steps would be followed to detect the faces in real-time.

Face Recognition

As we have encodings of the training images and also that of the people in real time, face distance is found i.e. the basic difference between the training images and the real image through webcam. Image having the least face distance is the desired person. In this SVM classifier is used. Therefore in real time person's face is surrounded by rectangular box along with name. Its really efficient even in dim lights as it's using HOG.

Attendance Updation

Recognized attendee's name and time is updated in excel sheet, that could be further used by the authorities.

Code implementation

File Edit View Navigate Code Refactor Run Tools VCS Window Help

python > FaceRecognitionProject > AttendanceProject.py

Attendance.csv ×   AttendanceProject.py ×   Attendance.py ×   Mansi Saini.jpg ×   Basics.py ×

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


path='ImagesAttendance'
images=[]
classNames=[]
myList=os.listdir(path)
print(myList)
for cl in myList:
    curImg = cv2.imread(f'{path}/{cl}')
    images.append(curImg)
    classNames.append(os.path.splitext(cl)[0])
print(classNames)


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


def markAttendance(name):
    with open('Attendance.csv', 'r+') as f:
        myDataList = f.readlines()
        nameList=[]
        for line in myDataList:
            entry = line.split(',')
```

6: TODO   Terminal   Python Console

PyCharm 2019.3.5 available: // Update... (40 minutes ago)

40:1   CRLF   UTF-8   4 spaces   Python 3.8

File Edit View Navigate Code Refactor Run Tools VCS Window Help

python > FaceRecognitionProject > AttendanceProject.py

Attendance.csv   AttendanceProject.py   Attendance.py   Mansi Saini.jpg   Basics.py

```python
33            if name not in nameList:
34                now = datetime.now()
35                dtString = now.strftime('%H:%M:%S')
36                f.writelines(f'\n{name},{dtString}')
37
38
39    encodeListKnown = findEncodings(images)
40    |
41    print("ENCODING COMPLETE!!!")
42
43    cap = cv2.VideoCapture(0)
44
45    while True:
46        success, img = cap.read()
47        imgS = cv2.resize(img,(0,0),None,0.25,0.25)
48        imgS = cv2.cvtColor(imgS,cv2.COLOR_BGR2RGB)
49
50
51        facesCurFrame = face_recognition.face_locations(imgS)
52        encodesCurFrame = face_recognition.face_encodings(imgS, facesCurFrame)
53
54        for encodeFace,faceLoc in zip(encodesCurFrame,facesCurFrame):
55            matches = face_recognition.compare_faces(encodeListKnown, encodeFace)
56            faceDis = face_recognition.face_distance(encodeListKnown, encodeFace)
57            print(faceDis)
58            matchIndex = np.argmin(faceDis)
59
60            if matches[matchIndex]:
61                name = classNames[matchIndex].upper()
62                print(name)
63                y1, x2, y2, x1 = faceLoc
```
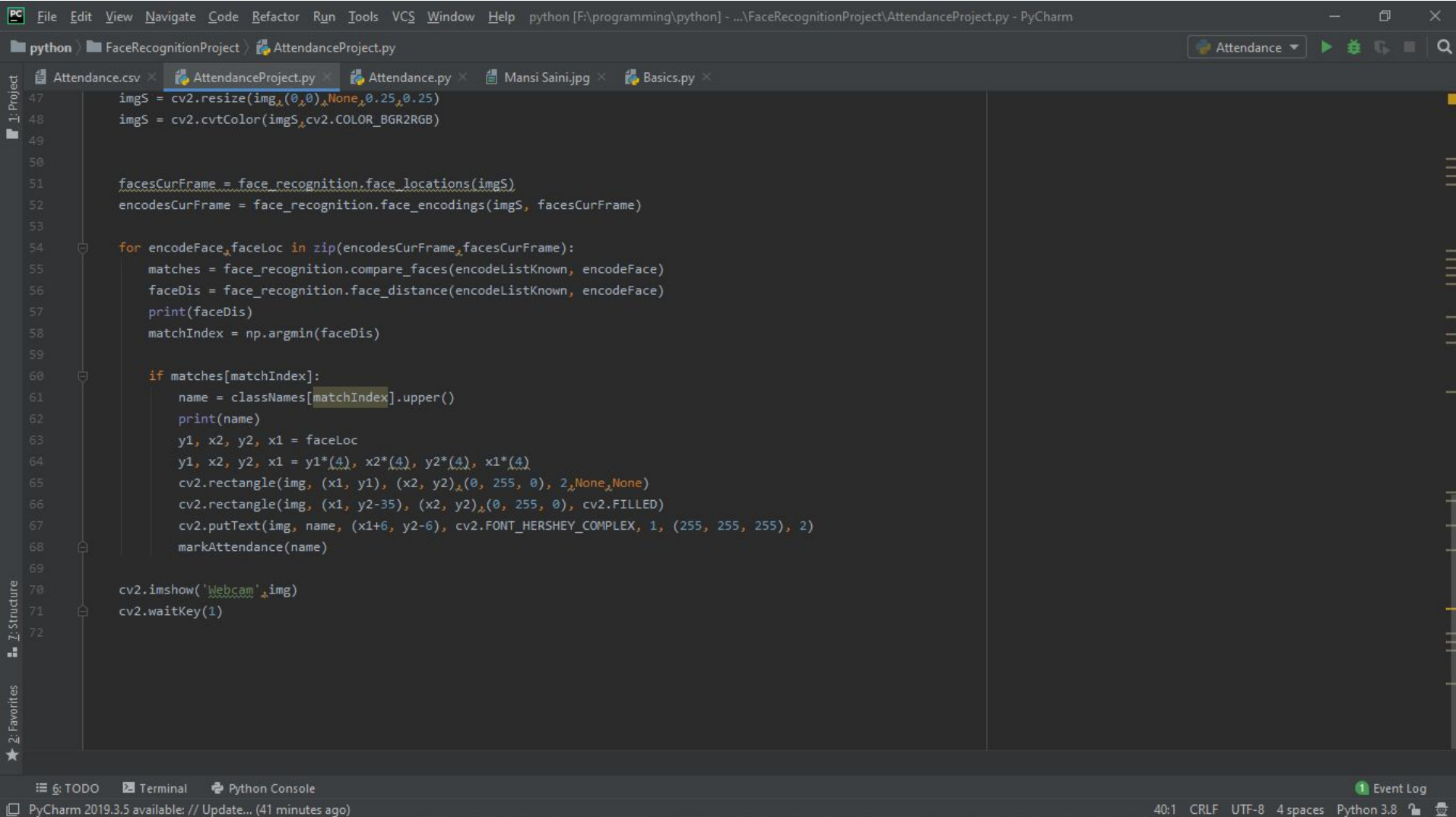
6: TODO   Terminal   Python Console

PyCharm 2019.3.5 available: // Update... (40 minutes ago)   40:1   CRLF   UTF-8   4 spaces   Python 3.8

```python
        imgS = cv2.resize(img,(0,0),None,0.25,0.25)
        imgS = cv2.cvtColor(imgS,cv2.COLOR_BGR2RGB)


        facesCurFrame = face_recognition.face_locations(imgS)
        encodesCurFrame = face_recognition.face_encodings(imgS, facesCurFrame)

        for encodeFace,faceLoc in zip(encodesCurFrame,facesCurFrame):
            matches = face_recognition.compare_faces(encodeListKnown, encodeFace)
            faceDis = face_recognition.face_distance(encodeListKnown, encodeFace)
            print(faceDis)
            matchIndex = np.argmin(faceDis)

            if matches[matchIndex]:
                name = classNames[matchIndex].upper()
                print(name)
                y1, x2, y2, x1 = faceLoc
                y1, x2, y2, x1 = y1*4, x2*4, y2*4, x1*4
                cv2.rectangle(img, (x1, y1), (x2, y2),(0, 255, 0), 2,None,None)
                cv2.rectangle(img, (x1, y2-35), (x2, y2),(0, 255, 0), cv2.FILLED)
                cv2.putText(img, name, (x1+6, y2-6), cv2.FONT_HERSHEY_COMPLEX, 1, (255, 255, 255), 2)
                markAttendance(name)

    cv2.imshow('Webcam',img)
    cv2.waitKey(1)
```

Demo Video

Testing

Input   **Output**

RESULT

Conclusion

Basically this system aims to build an effective class attendance system and improving this system in every domain like schools, colleges, organizations, institutions and companies using face recognition technique. Capturing live images from camera and applying different techniques of face detection and face recognition will reduce manual or traditional work and hence will save the time. In our solution, by creating interface, we generate the database. The proposed system will be able to mark the attendance using face Id and webcam and then recognize the faces. We trained the images using SVM classifier. When stored images and compared images matched then attendance sheet get updated automatically along with date and time in excel sheet. Through this system, it becomes easy for higher authorities to keep track on time of student or employees.

Thank you!