Chapter 1

Introduction

- In recent years, face recognition has attracted much attention and its research has rapidly expanded by not only engineers but also neuroscientists, since it has many potential applications in computer vision communication and automatic access control system.
- However, face detection is not straightforward because it has lots of variations of image appearance, such as pose variation (front, non-front), occlusion, image orientation, illuminating condition and facial expression.
- This project presents a face detection technique mainly based on the colour segmentation, image segmentation and template matching methods.
- Face recognition is the technique in which the identity of a human being can be identified using one's individual face. Such kind of systems can be used in photos, videos, or in real time machines.
- The objective of this article is to provide a simpler and easy method in machine technology. With the help of such a technology one can easily detect the face by the help of dataset in similar matching appearance of a person.
- This method is useful in many fields such as the military, for security, schools, colleges and universities, airlines, banking, online web applications, gaming etc.
- This system uses powerful python algorithm through which the detection and recognition of face is very easy and efficient.

Chapter 2

System Analysis

- **EXISTING SYSTEM:** INCLUDES 5 DIFFERENT SYSTEMS.

- **Principal Component Analysis, or PCA**, is a dimensionality-reduction method that is often used to reduce the dimensionality of large data sets, by transforming a large set of variables into a smaller one that still contains most of the information in the large set. Reducing the number of variables of a data set naturally comes at the expense of accuracy, but the trick in dimensionality reduction is to trade a little accuracy for simplicity. Because smaller data sets are easier to explore and visualize and make analyzing data much easier and faster for machine learning algorithms without extraneous variables to process. So to sum up, the idea of PCA is simple, reduce the number of variables of a data set, while preserving as much information as possible.
- **Linear Discriminant Analysis (LDA)**, is most commonly used as dimensionality reduction technique in the pre-processing step for pattern-classification and machine learning applications. The goal is to project a dataset onto a lower-dimensional space with good class-separability in order avoid overfitting ("curse of dimensionality") and also reduce computational costs.
- **Skin colour**, Face detection based on skin color extraction scheme can eliminate a large number of non-face background and determine the face area. Then, use the AdaBoost algorithm to detect the face accurately, so as to improve the speed and accuracy of the face detection system.
- **Wavelets**, M−Band wavelets are used to decompose face images into M2 frequency levels. The efficiency of this approach is tested using the eigenface procedure for face recognition. Along with signal decomposition, Wavelet methods allow for data compression, thus reducing computational efforts.

- **Artificial Neural Network (ANN),** The term "Artificial Neural Network" is derived from Biological neural networks that develop the structure of a human brain. Similar to the human brain that has neurons interconnected to one another, artificial neural networks also have neurons that are interconnected to one another in various layers of the networks. These neurons are known as nodes.

    - **PROPOSED SYSTEM:** INCLUDES 3 STEPS**.**

1. Overview of Face Detection:

What if the machine is able to detect objects automatically in an image without human involvement? Let us see: Face detection can be such a problem where we detect human faces in an image. There might be slight differences in human faces, but after all, it is safe to say that there are specific features that are associated with all human faces. Various face detection algorithms are there but the Viola-Jones Algorithm is the oldest method that is also used today. Face detection is generally the first step towards many face-related applications like face recognition or face verification. But, face detection has very useful applications. One of the most successful applications of face detection is probably "photo-taking".

Example: When you click a photo of your friends, the camera in which the face detection algorithm has built-in detects where the faces are and adjusts focus accordingly.

2. Overview of Face Recognition:

Now we have seen our algorithms can detect faces but can we also recognize whose faces are there? And what if an algorithm is able to recognize faces? Generally, Face Recognition is a method of identifying or verifying the identity of an individual by using their face. Various algorithms are there for face recognition but their accuracy might vary. Here we are going to discuss with you that how we can do face recognition using OpenCV. Here we use face embeddings in which every face is converted into a vector. The technique of converting the face into a vector is called deep metric learning. Let me divide this process into three simple steps for better and easy understanding:

- **Face Detection:**

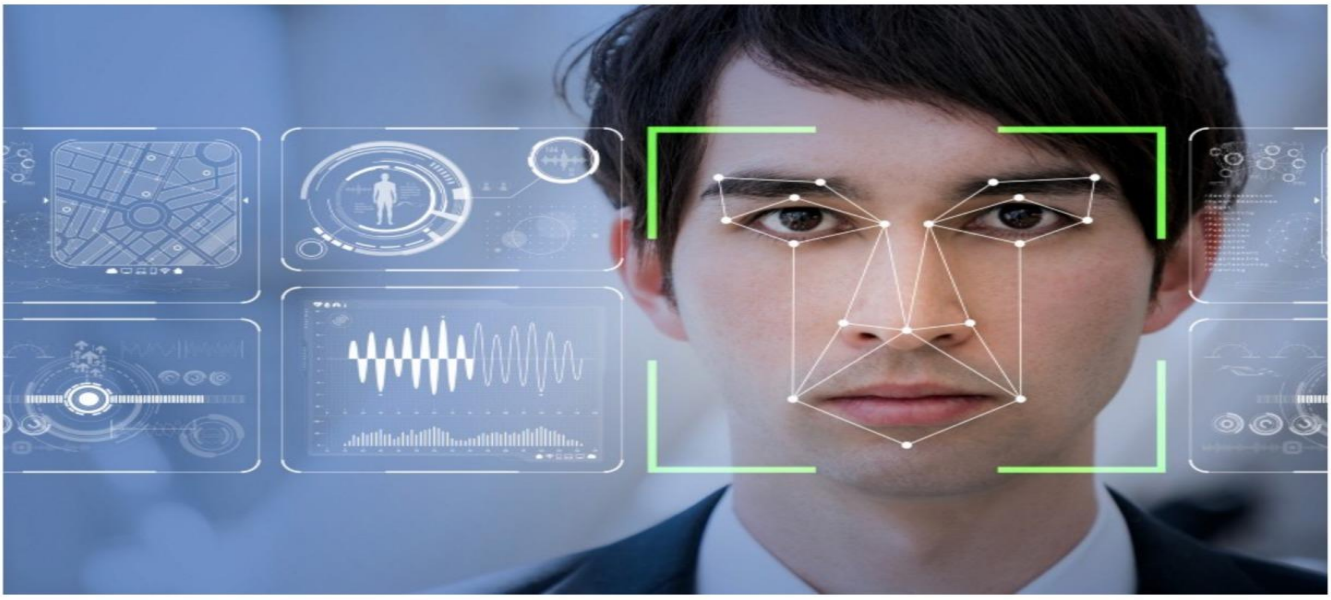
The first task that we perform is detecting faces in the image(photograph) or video stream. Now we know that the exact coordinates/location of the face, so we extract this face for further processing.

- **Feature Extraction:**

Now see we have cropped out the face from the image, so we extract specific features from it. Here we are going to see how to use face embeddings to extract these features of the face. As we know a neural network takes an image of the face of the person as input and outputs a vector that represents the most important features of a face! In machine learning, this vector is nothing but called embedding and hence we call this vector face embedding. Now how this will help in recognizing the faces of different people? When we train the neural network, the network learns to output similar vectors for faces that look similar. Let us consider an example, if I have multiple images of faces within different timelapse, it's obvious that some features may change but not too much. So in this problem, the vectors associated with the faces are similar or we can say they are very close in the vector space. Up to this point, we came to know how this network works, let us see how to use this network on our own data. Here we pass all the images in our data to this pre-trained network to get the respective embeddings and save these embeddings in a file for the next step.

- **Comparing Faces: (FIG:2.1)**

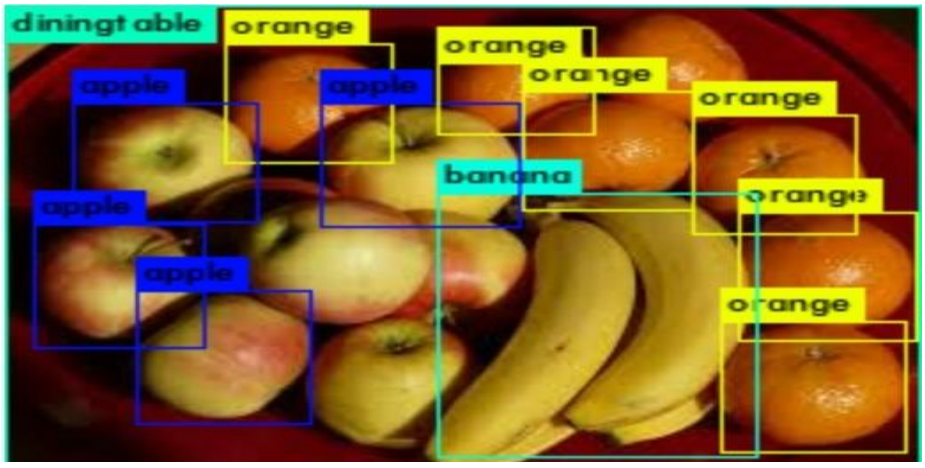
We have face embeddings for each face in our data saved in a file, the next step is to recognize a new image that is not in our data. Hence the first step is to compute the face embedding for the image using the same network we used earlier and then compare this embedding with the rest of the embeddings that we have. We recognize the face if the generated embedding is closer or similar to any other embedding.



3. Understand what is OpenCV: (FIG:2.2)

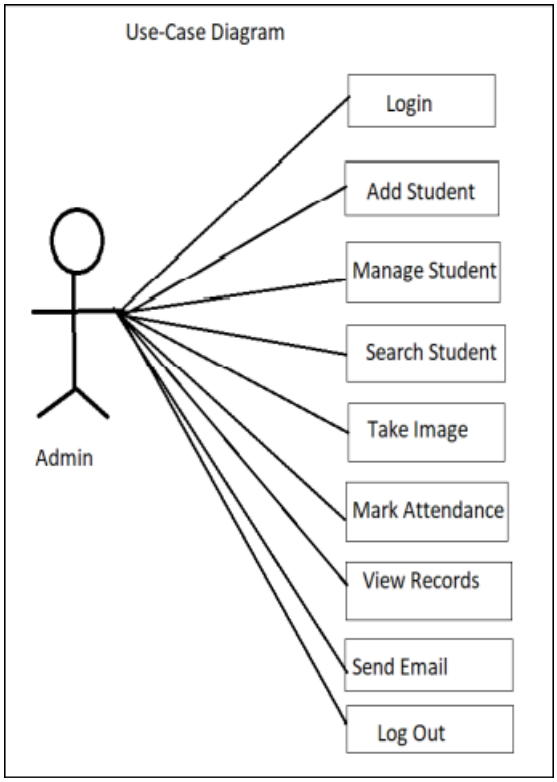
In the Artificial Intelligence field, Computer Vision is one of the most interesting and challenging tasks. Computer Vision acts as a bridge between Computer Software and visualizations. Computer Vision allows computer software to understand and learn about the visualizations in the surroundings.

Let us understand an example: Based on the shape, colour, and size that determines the fruit. This task is very easy for the human brain but in the Computer Vision pipeline, first, we need to gather the data, then we perform the data processing operations, and then we train and teach the model to understand how to distinguish between the fruits based on its size, shape, and colour of the fruit.



Chapter 3

System Design



- **OVERVIEW LAYOUT. (ER DIAGRAM 3.1 & 3.2)**

Step 1: Taking the image.

Step 2: Detecting the total faces in the image.

Step3: Cropping the image into total faces.

Step4: Applying pre-processing algorithms.

Step 5: Classification of faces as known and unknown faces.

- **Module for registration/Data feeding into system.**

Step 1: Admin feeding details.

Step 2: Admin requesting for system resource.

Step 3: Capture image for database.

Step 4: Training the Dataset and storing into database

- **SYSTEM ARCHITECTURE:**

- **This system can be divided into four main modules:**

1. **Face detection.**

The appropriate and effective facial detection algorithm constantly improves facial recognition. Several facial algorithms such as face-to-face geometry, construction methods, Face geometry-based methods, Feature Invariant methods, Figure 2: System Diagram Machine learning based methods. Out of all these methods Viola and Jones proposed a framework that gives a high detection rate and is also fast. Viola-Jones detection algorithm is fast and robust. So we chose Viola-Jones face detection algorithm, which uses Integral Image and AdaBoost learning algorithm as classier. We have observed that this algorithm yields better results in a variety of lighting conditions.

2. **Pre-processing.**

Extracting the face features it is called pre-processing. This pre-processes step involves specifying the extracted facial image and transforms to 100x100. Histogram Equalization is the most commonly used Histogram Normalization technique. This improves the contrast of the image as it extends beyond the intensity of the image, making it even more clear and constraint.

3. **Database development.**

As we choose biometric based system every individual is required. This database development phase consists of an image capture of each individual and extracting the biometric feature, and then it is enhanced using preprocessing techniques and stored in the database.

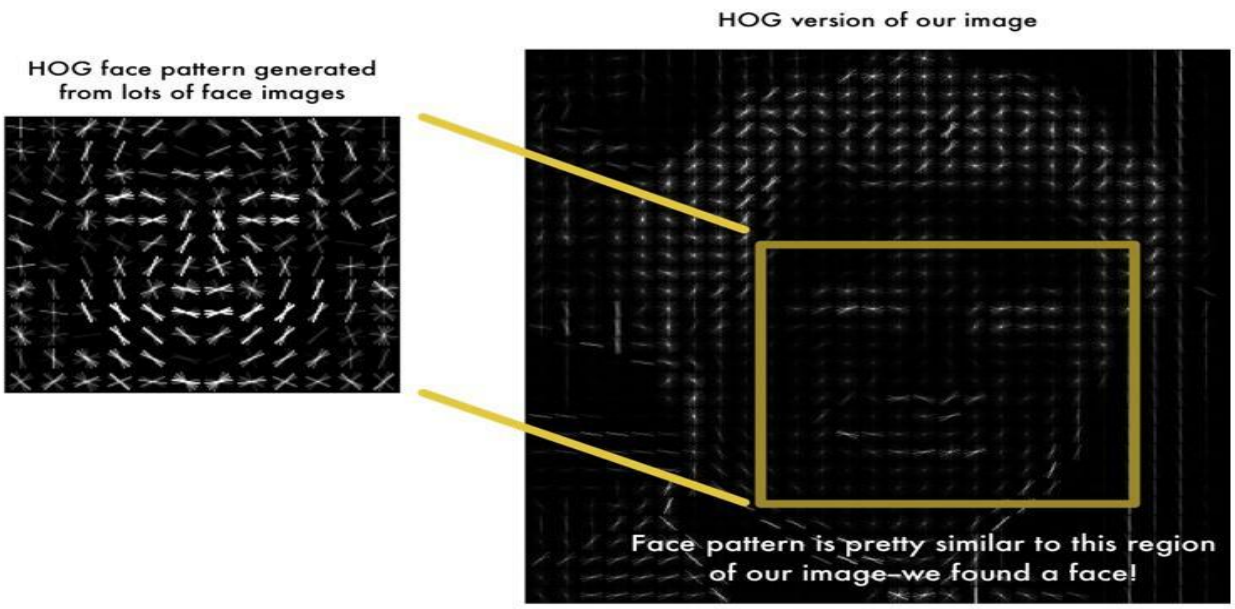
4. **Post-processing.**

In the proposed system, after recognizing the faces of the person, the names are show into a video output. The result is generated by exporting mechanism present in the database system. These generated records can be seen in real time video. This ensures that person

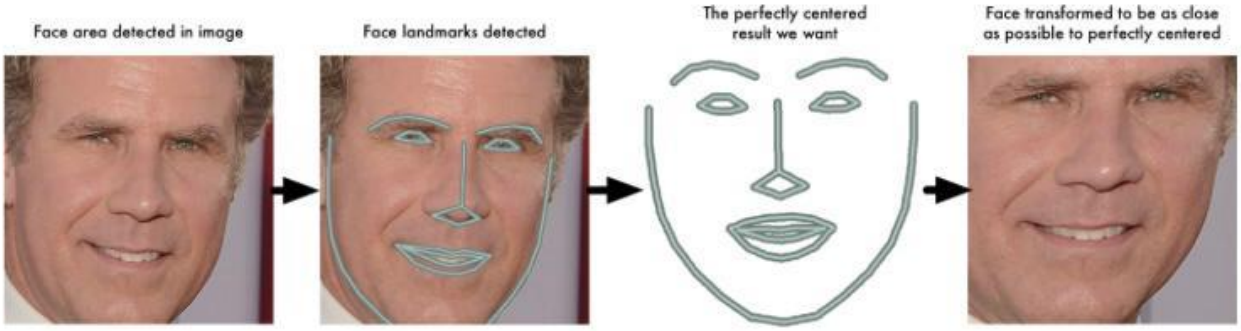
whose faces are not recognized correctly by the system have to check in database. And thus, giving them the ability to correct the system and make it more stable and accurate.

- **PICTORIAL REPRESENTATION:**

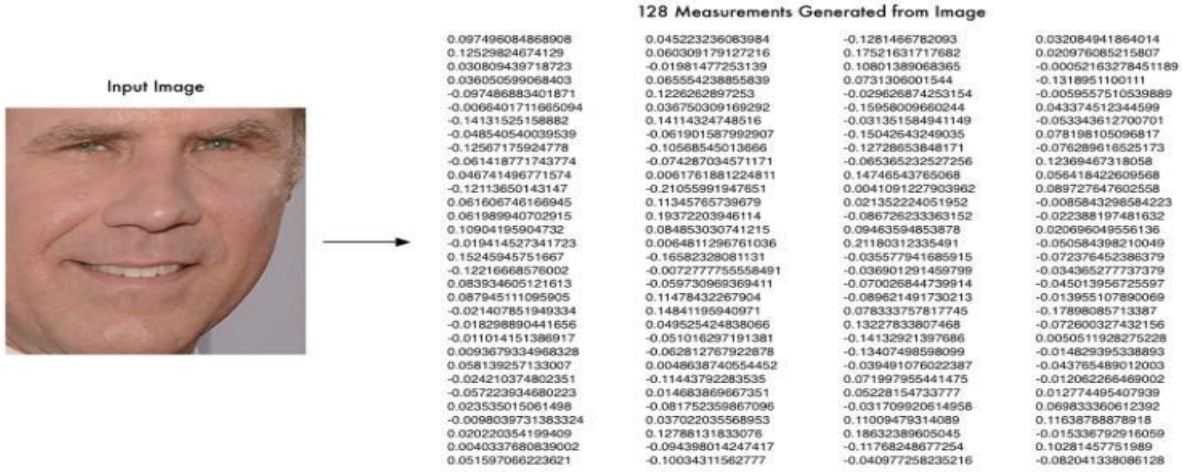
- **DETECTING FACE. (FIG:3.3)**



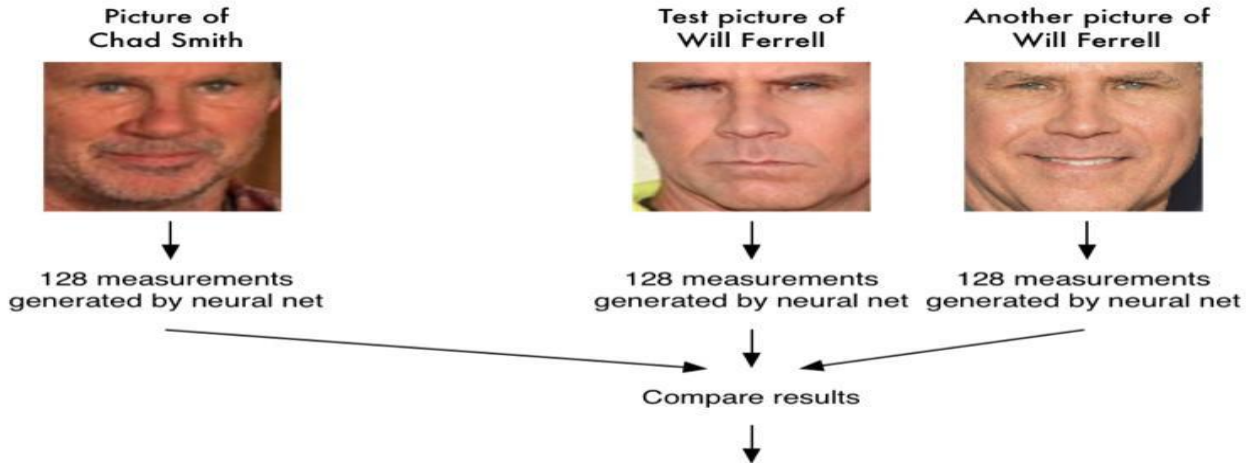
- **HOG (histogram of oriented gradients) OF THE IMAGE. (FIG:3.4)**



- **CENTERED IMAGE. (FIG:3.5)**



- **ENCODED FACE. (FIG:3.6)**



- **COMPARISON. (FIG:3.7)**

Chapter 4

System Implementation

- **WE HAVE USED 6 DIFFERENT LIBRARIES:**

1. **Cmake:**

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

2. **Dlib:**

Dlib is a modern C++ toolkit containing machine learning algorithms and tools for creating complex software in C++ to solve real world problems. It is used in both industry and academia in a wide range of domains including robotics, embedded devices, mobile phones, and large high performance computing environments. Dlib's open source licensing allows you to use it in any application, free of charge.

3. **Numpy:**

NumPy is a Python library used for working with arrays. It also has functions for working in domain of linear algebra, fourier transform, and matrices. NumPy was created in 2005 by Travis Oliphant. It is an open source project and you can use it freely. NumPy stands for Numerical Python.

4. **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.

5. **OpenCv:**

OpenCV is the most popular library for computer vision. Originally written in C/C++, it now provides bindings for Python.OpenCV uses machine learning algorithms to search for faces within a picture. Because faces are so complicated, there isn't one simple test that will tell you if it found a face or not. Instead, there are thousands of small patterns and features that must be matched. The algorithms break the task of identifying the face into thousands of smaller, bite-sized tasks, each of which is easy to solve. These tasks are also called classifiers.

Advantages of OpenCV:

1. Open CV is free of cost and an open-source library.
2. Open CV is fast as it is written in C/C++ language as compared to others
3. With less system RAM, OpenCV works better.
4. It supports most of the operating systems like Windows, Linux, and macOS.

6. **OS:**

The OS module in Python provides functions for creating and removing a directory (folder), fetching its contents, changing and identifying the current directory, etc.

- **COMPONENTS USED ARE:**

- **HARDWARE:**

- Laptop (acer swift 3)
- External Webcam (1080 pixels, 1920 x 1080 resolution)
- Focal Length- F/2.0.

- **SOFTWARE:**

- IDE-PYCHARM

Chapter 5

System Testing

1. The system is tested with Various changes in the view, both the camera view and facial positions

2. The system was first tested with a series of celebrity images, when all of the test images were detected with the correct output we tested the system with the images of our college staff

3. Both the comparisons where done, where the first comparison was between the base image and test image which were both pre-loaded in the system and the second comparison was between the base image and image from the live camera

4. Both single and Simultaneous image testing was done, where a group of people simultaneously come in the camera frame to get the faces detected

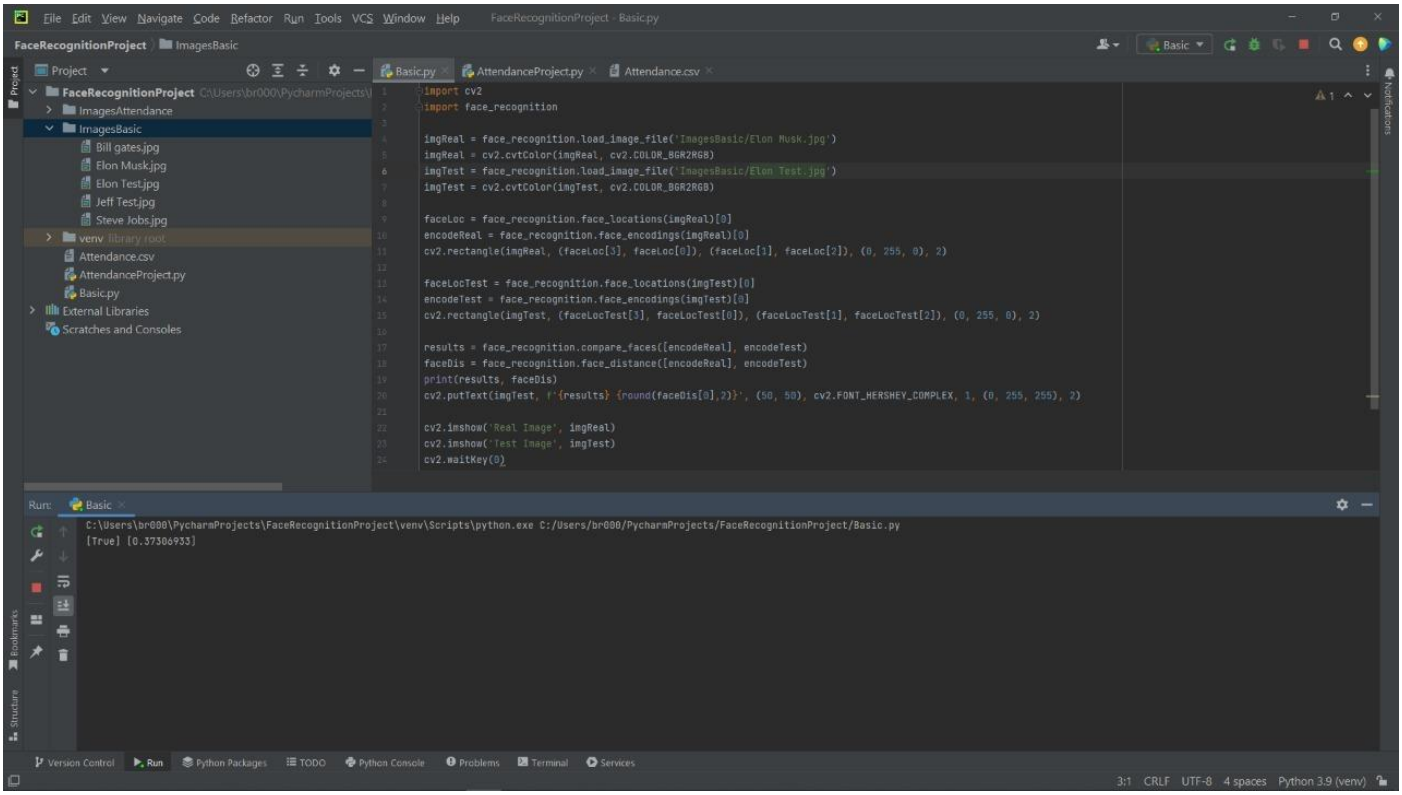
5. Changes in the size of the base image was also done to check for the accuracy.

6. We even tested the system with minor changes in the facial structure of the same person like with and without the beard and also with and without the spectacles which both gave the correct outputs.

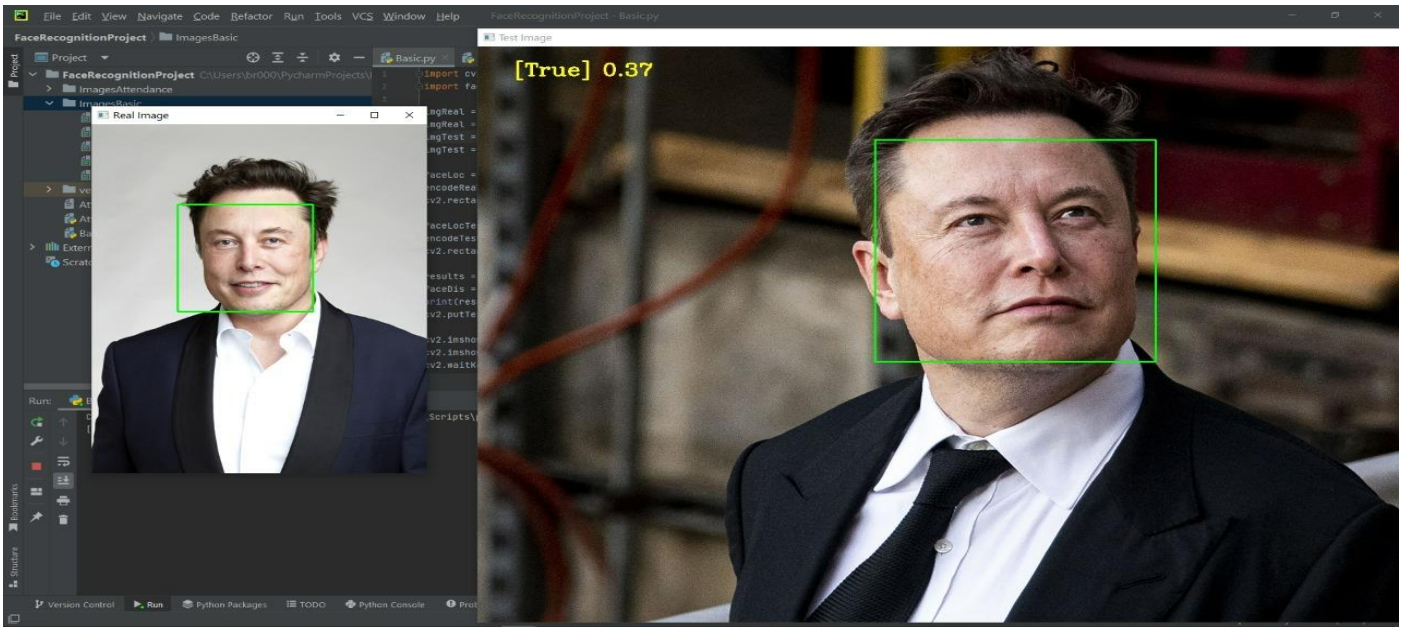
Chapter 6

Results

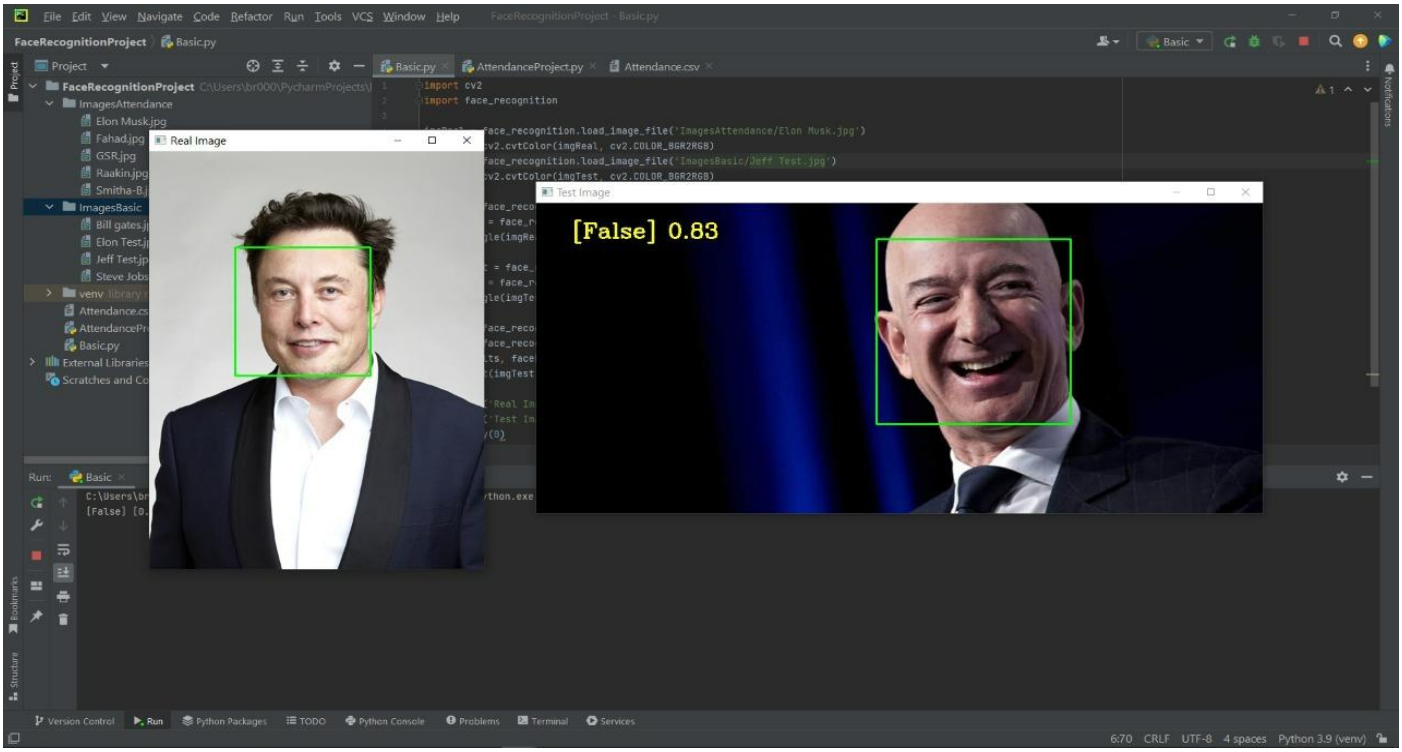
- **PART 1: BASIC.PY (OUTPUT AS TRUE OR FALSE) (FIG:6.1)**



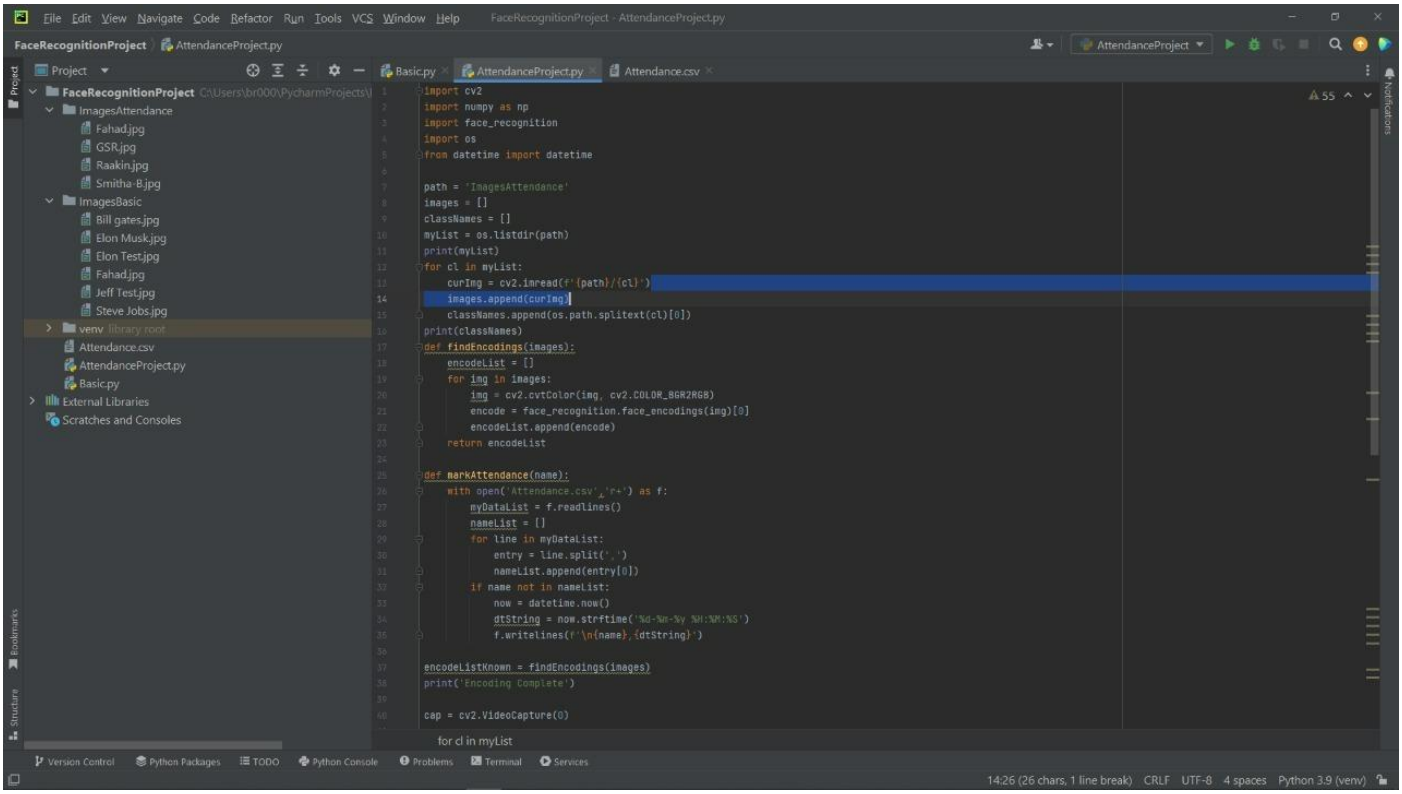
- **OUTPUT AS TRUE. (FIG:6.2)**



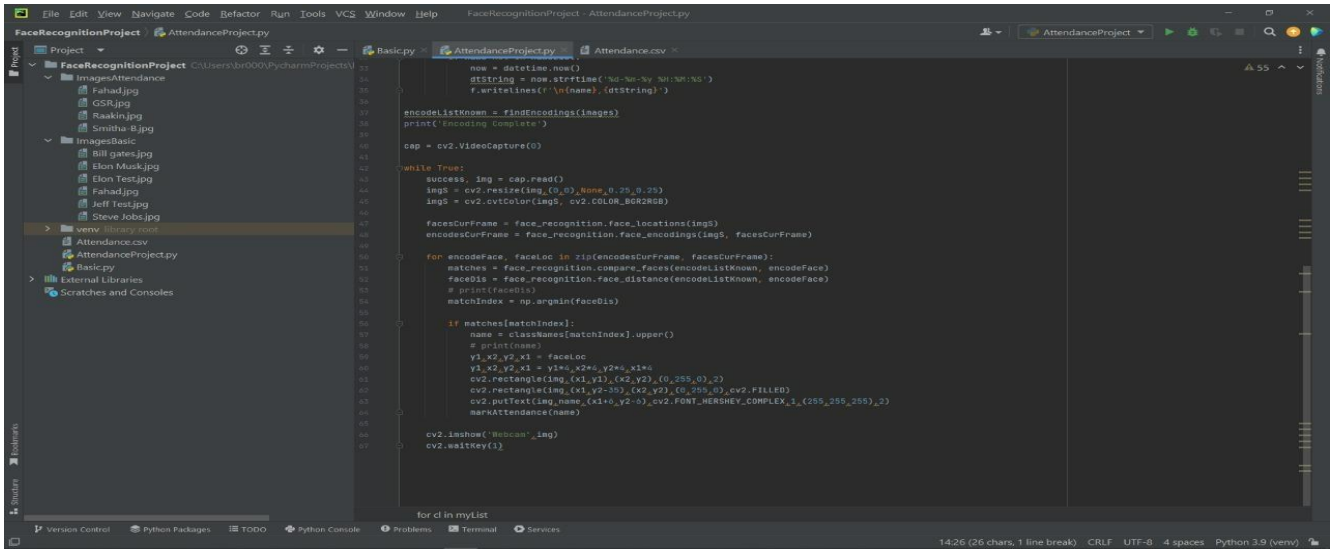
- **OUTPUT AS FALSE. (FIG:6.3)**



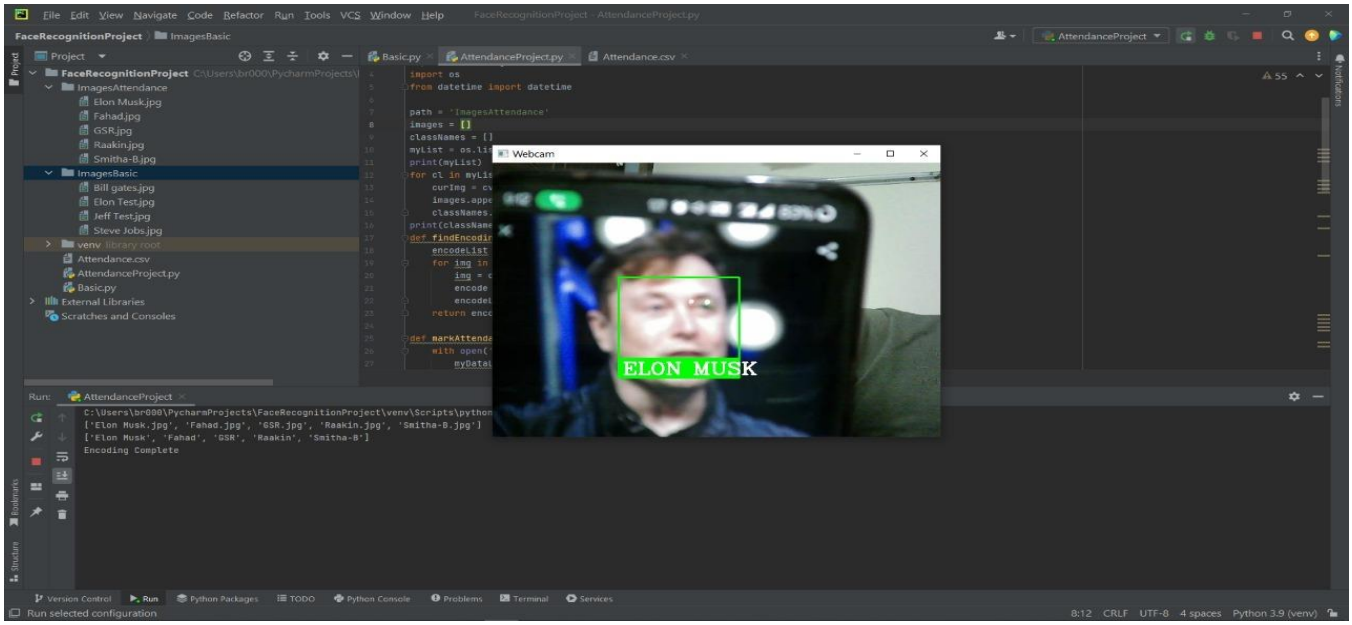
- **PART 2: ATTENDANCE REGISTRATION. (FIG:6.4)**



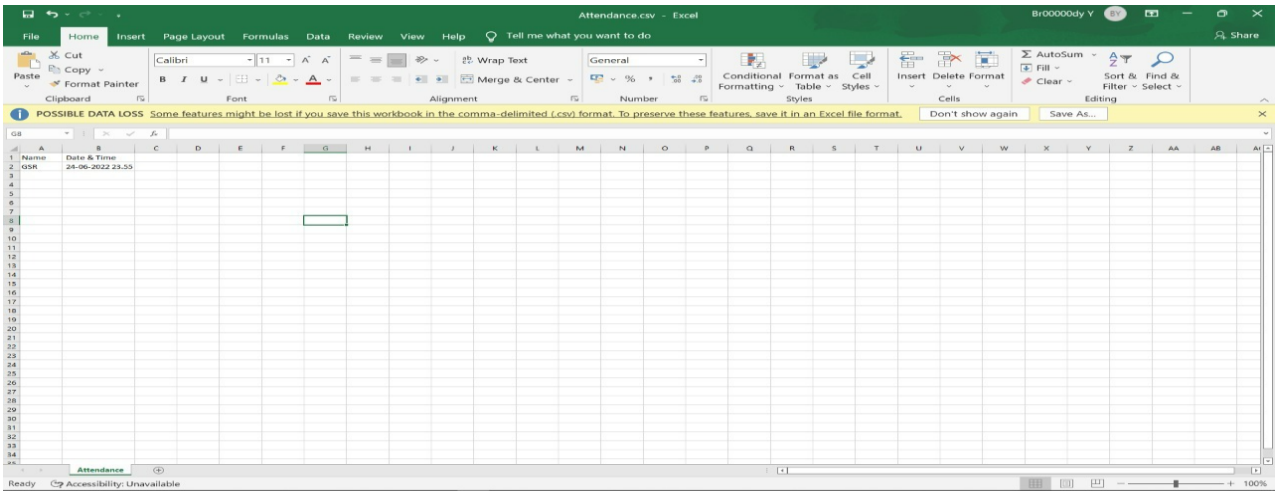
```
import cv2
import numpy as np
import face_recognition
import os
from datetime import datetime

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(',')
            nameList.append(entry[0])
        if name not in nameList:
            now = datetime.now()
            dtString = now.strftime('%d-%m-%y %H:%M:%S')
            f.writelines(f'\n{name},{dtString}')

encodeListKnown = findEncodings(images)
print('Encoding Complete')

cap = cv2.VideoCapture(0)
```



(FIG:6.5)



(FIG:6.6 and FIG:6.7)



Conclusion and Future Enhancements

- Face detection systems are currently associated with many top technological companies and industries making the work of face recognition easier.The use of python programming and OpenCV makes it an easier and handy tool or system which can be made by anyone according to their requirement. The proposed system discussed in this project will be helpful for many as it is user friendly and cost-efficient system. Hence by the use of python and OpenCV the face recognition system can be designed for various purposes.

- Application of front-end code to make the entire system more functional and easy for the user
- In a recent study scientists have shown that including someone's ears and its dimensions can increase the accuracy to about 6 percent. Hence this feature can also be improvised into our current system
- A new and separate database can also be created for attendees list to be stored and made accessible for various groups
- Automated email facility can also be added to the current system where the absence of a person will be verified.
- The current system can also be used in hospitals with some minor changes where the patients medical history can be displayed to the doctor making him aware of his allergies or other symptoms before the diagnosis.

References

[1]. “Attendance System Using NFC Technology with Embedded Camera on Mobile Device” (Bhise, Khichi, Korde,Lokare, 2015)

[2]. K.SenthamilSelvi, P.Chitrakala, A.AntonyJenitha, "Face Recognition Based Attendance Marking System", IJCSMC, Vol. 3, Issue.

[3]. Face Recognition Homepage, http://www.face�rec.org/a lgorithms/

[4]. “RFID based Student Attendance System” (Hussain, Dugar, Deka, Hannan, 2014)

[5]. OpenCvDocumentation - https://opencv.org

[6]. Numpy - https://numpy.org

[7]. https://www.analyticsvidhya.com/blog/2021/06/learn-how-to-implement-face-recognition-using-opencv-with-python/#h2_8

[8]. Face Detection and Tracking using OpenCV.

S.V.Viraktamath, Mukund Katti, Aditya Khatawkar, Pavan Kulkarni. 3, s.l.: SIJ, July-August 2013, The Standard International Journals (The SIJ) , Vol. 1, pp. 45-50. ISSN: 2321 – 2403