***PROJECT REPORT***

***Data Science (CSL-487)***

**

***BS (IT)-6(A)***

***Project Title:***

***FACE RECOGNITION ATTENDANCE SYSTEM***

*Group Members*

|  |  |
| --- | --- |
| ***Name*** | ***Enrollment*** |
| *1.Samiya Komal* | ***02-235201-005*** |
| *2.Muhammad Umar* | ***02-235201-004*** |
| *3.Muhammad Umer Rasheed* | ***02-235201-011*** |

***Submitted to:***

***MAJID RIAZ***

***BAHRIA UNIVERSITY KARACHI CAMPUS***

*Department of Computer Science*

Table of Contents

Contents

[Abstract 3](#_Toc125023767)

[Introduction 4](#_Toc125023768)

[1.1. Introduction 5](#_Toc125023769)

[1.2. Background 5](#_Toc125023770)

[1.3. Problem Statement 6](#_Toc125023771)

[1.4. Aims and Objectives 7](#_Toc125023772)

[1.5. Scope of the project 7](#_Toc125023773)

[Literature Review 8](#_Toc125023774)

[2.1. Student Attendance System 9](#_Toc125023775)

[2.2. Difference between Face Detection and Face Recognition 9](#_Toc125023776)

[Model Implementation & analysis 11](#_Toc125023777)

[3.1. Introduction 12](#_Toc125023778)

[3.2. Design Requirements 12](#_Toc125023779)

[3.3. Experimental Results 14](#_Toc125023780)

[4.1. Future Scope of Work 15](#_Toc125023781)

[Code Implementation 17](#_Toc125023782)

[4.1. Code Implementation 18](#_Toc125023783)

[4.1.1. main\_python\_file.ipynb 18](#_Toc125023784)

[4.1.2. Mail function 37](#_Toc125023785)

[System Outputs 38](#_Toc125023786)

### Abstract

Uniqueness or individuality of an individual face is the representation of one’s identity. In this project face of an individual is used for the purpose of attendance making automatically. Attendance of the student is very important for every college, universities and school and also the most difficult task in any organization. Conventional methodology for taking attendance is by calling the name or roll number of the student and the attendance is recorded. Time consumption for this purpose is an important point of concern. Assume that the duration for one subject is around 60 minutes or 1 hour & to record attendance takes 5 to 10 minutes. For every tutor this is consumption of time. To stay away from these losses, an automatic process is used in this project which is based on image processing.

In this project face detection and face recognition is used. Face detection is used to locate the position of face region and face recognition is used for marking the attendance. The database of all the students in the class is stored and when the face of the individual student matches with one of the faces stored in the database then the attendance is recorded. Face recognition is a biometric technique that determines whether the image of a person’s face matches any of the face images stored in a database. The facial recognition feature embedded in the attendance monitoring system not only ensures accurate attendance but also eliminates flaws. Using a system to overcome defects not only saves resources but also reduces human intervention in the overall process by delegating all complex tasks to the system.

Chapter 1

# Introduction

## Introduction

Attendance is prime important for both the teacher and student of an educational organization. So it is very important to keep record of the attendance. The problem arises when we think about the traditional process of taking attendance in class room.

Calling name or roll number of the student for attendance is not only a problem of time consumption but also it needs energy. So an automatic attendance system can solve all above problems.

There are some automatic attendances making system which are currently used by much institution. One of such system is biometric technique and RFID system. Although it is automatic and a step ahead of traditional method it fails to meet the time constraint. The student has to wait in queue for giving attendance, which is time taking.

This project introduces an involuntary attendance marking system, devoid of any kind of interference with the normal teaching procedure. The system can be also implemented during exam sessions or in other teaching activities where attendance is highly essential. This system eliminates classical student identification such as calling name of the student, or checking respective identification cards of the student, which can not only interfere with the ongoing teaching process, but also can be stressful for students during examination sessions. In addition, the students have to register in the database to be recognized. The enrolment can be done on the spot through the user-friendly interface.

## Background

Face recognition is crucial in daily life in order to identify family, friends or someone we are familiar with. We might not perceive that several steps have actually taken in order to identify human faces. Human intelligence allows us to receive information and interpret the information in the recognition process.

After visual processing done by the human visual system, we actually classify shape, size, contour and the texture of the object in order to analyze the information. The analyzed information will be compared to other representations of objects or face that exist in our memory to recognize. In fact, it is a hard challenge to build an automated system to have the same capability as a human to recognize faces. However, we need large memory to recognize different faces, for example, in the Universities, there are a lot of students with different race and gender, it is impossible to remember every face of the individual without making mistakes. In order to overcome human limitations, computers with almost limitless memory, high processing speed and power are used in face recognition systems.

The human face is a unique representation of individual identity. Thus, face recognition is defined as a biometric method in which identification of an individual is performed by comparing real-time capture image with stored images in the database of that person.

Nowadays, face recognition system is prevalent due to its simplicity and awesome performance. For instance, airport protection systems and FBI use face recognition for criminal investigations by tracking suspects, missing children and drug activities. Apart from that, Facebook which is a popular social networking website implement face recognition to allow the users to tag their friends in the photo for entertainment purposes. Furthermore, Intel Company allows the users to use face recognition to get access to their online account. Apple allows the users to unlock their mobile phone, iPhone X by using face recognition.

The work on face recognition began in 1960. Woody Bledsoe, Helen Chan Wolf and Charles Bisson had introduced a system which required the administrator to locate eyes, ears, nose and mouth from images. The distance and ratios between the located features and the common reference points are then calculated and compared. The studies are further enhanced by Goldstein, Harmon, and Lesk in 1970 by using other features such as hair colour and lip thickness to automate the recognition. In 1988, Kirby and Sirovich first suggested principle component analysis (PCA) to solve face recognition problem. Many studies on face recognition were then conducted continuously until today.

## Problem Statement

Traditional student attendance marking technique is often facing a lot of trouble. The face recognition student attendance system emphasizes its simplicity by eliminating classical student attendance marking technique such as calling student names or checking respective identification cards. There are not only disturbing the teaching process but also causes distraction for students during exam sessions. Apart from calling names, attendance sheet is passed around the classroom during the lecture sessions. The lecture class especially the class with a large number of students might find it difficult to have the attendance sheet being passed around the class. Thus, face recognition attendance system is proposed in order to replace the manual signing of the presence of students which are burdensome and causes students get distracted in order to sign for their attendance. Furthermore, the face recognition based automated student attendance system able to overcome the problem of fraudulent approach and lecturers does not have to count the number of students several times to ensure the presence of the students.

Different projects of face recognition has listed the difficulties of facial identification. One of the difficulties of facial identification is the identification between known and unknown images. In addition, the training process for face recognition student attendance system is slow and time-consuming. In addition, the different lighting and head poses are often the problems that could degrade the performance of face recognition based student attendance system.

Hence, there is a need to develop a real time operating student attendance system which means the identification process must be done within defined time constraints to prevent omission. The extracted features from facial images which represent the identity of the students have to be consistent towards a change in background, illumination, pose and expression. High accuracy and fast computation time will be the evaluation points of the performance.

## Aims and Objectives

The objective of this project is to develop face recognition attendance system. Expected achievements in order to fulfill the objectives are:

* + - To detect the face segment from the video frame.
    - To extract the useful features from the face detected.
    - To classify the features in order to recognize the face detected.
    - To record the attendance of the identified student.

## Scope of the project

We are setting up to design a system comprising of four modules.

1. Check Camera
2. Mark Attendance
3. Manage Students
4. Manage Teachers

The first module Check Camera is basically a camera application that captures student faces and stores them in a file using computer vision and face extraction techniques. The second module Mark Attendance does face recognition of the captured images (faces) in the file, marks the attendance by comparing the details with the trained data and then stores the results in a database for future analysis. The third module is Manage Students by which you can View all Students, Add new Student and Remove Students. The last module is Manage Teachers in which there is a functionality to View all Teachers and Add and Remove Teacher.

Chapter 2

# Literature Review

## Student Attendance System

There are also may different methods for attendance including RFID (Radio Frequency Identification) card system, fingerprint system and iris recognition system but they are not as efficient as face recognition attendance system. RFID card system is implemented due to its simplicity. However, the user tends to help their friends to check in as long as they have their friend’s ID card. The fingerprint system is indeed effective but not efficient because it takes time for the verification process so the user has to line up and perform the verification one by one. However for face recognition, the human face is always exposed and contains less information compared to iris. Iris recognition system which contains more detail might invade the privacy of the user. Voice recognition is available, but it is less accurate compared to other methods. Hence, face recognition system is suggested to be implemented in the student attendance system.

|  |  |  |
| --- | --- | --- |
| **System Type** | **Advantage** | **Disadvantages** |
| RFID card system | Simple | Fraudulent usage |
| Fingerprint system | Accurate | Time-consuming |
| Voice recognition system |  | Less accurate compared to Others |
| Iris recognition system | Accurate | Privacy Invasion |

Table 2.1: Advantages & Disadvantages of Different Biometric System

## Difference between Face Detection and Face Recognition

Face detection answers the question, Where is the face? It identifies an object as a “face” and locates it in the input image. Face Recognition on the other hand answers the question who is this? Or whose face is it? It decides if the detected face is someone known or unknown based on the database of faces it uses to validate this input image. It can therefore be seen that face detections output (the detected face) is the input to the face recognizer and the face Recognition’s output is the final decision i.e. face known or face unknown.

A face Detector has to tell whether an image of arbitrary size contains a human face and if so, where it is. Face detection can be performed based on several cues: skin color (for faces in color images and videos, motion (for faces in videos), facial/head shape, facial appearance or a combination of these parameters. Most face detection algorithms are appearance based without using other cues. An input image is scanned at all possible locations and scales by a sub window. Face detection is posed as classifying the pattern in the sub window either as a face or a non-face. The face/nonface classifier is learned from face and non-face training examples using statistical learning methods. Most modern algorithms are based on the Viola Jones object detection framework, which is based on Haar Cascades.

|  |  |  |
| --- | --- | --- |
| **Face Detection Method** | **Advantages** | **Disadvantages** |
| Viola Jones Algorithm | 1. High detection Speed. 2. High Accuracy. | 1. Long Training Time.  2. Limited Head Pose.  3. Not able to detect dark faces. |
| Local Binary Pattern Histogram | 1. Simple computation. 2.High tolerance against the monotonic | 1. Only used for binary and grey images.  2. Overall performance is inaccurate compared to Viola-Jones Algorithm. |

Chapter 3

# Model Implementation & analysis

## Introduction

Face detection involves separating image windows into two classes; one containing faces (turning the background (clutter). It is difficult because although commonalities exist between faces, they can vary considerably in terms of age, skin color and facial expression. The problem is further complicated by differing lighting conditions, image qualities and geometries, as well as the possibility of partial occlusion and disguise. An ideal face detector would therefore be able to detect the presence of any face under any set of lighting conditions, upon any background. The face detection task can be broken down into two steps. The first step is a classification task that takes some arbitrary image as input and outputs a binary value of yes or no, indicating whether there are any faces present in the image. The second step is the face localization task that aims to take an image as input and output the location of any face or faces within that image as some bounding box with (x, y, width, height).After taking the picture, the system will compare the equality of the pictures in its database and give the most related result.

We will use windows operating system, Google Colab, open CV platform and will do the coding in python language.

The main components used in the implementation approach are open source computer vision library (OpenCV). One of OpenCV’s goals is to provide a simple-to-use computer vision infrastructure that helps people build fairly sophisticated vision applications quickly. OpenCV library contains over 500 functions that span many areas in vision. The primary technology behind Face recognition is OpenCV. The user stands in front of the camera keeping a minimum distance of 50cm and his image is taken as an input. The frontal face is extracted from the image then converted to gray scale and stored. The Principal component Analysis (PCA) algorithm is performed on the images and the eigen values are stored in an xml file. When a user requests for recognition the

frontal face is extracted from the captured video frame through the camera. The eigen

value is re-calculatedfor the test face and it is matched with the stored data for the

closest neighbor.

## Design Requirements

We used some tools to build the desired system. Without the help of these tools it would not be possible to make it done. Here we will discuss about the most important one.

#### Software Implementation

* + - 1. **OpenCV:** We used OpenCV 2 dependency for python . OpenCV is library where there are lots of image processing functions are available. This is very useful library for image processing. Even one can get expected outcome without writing a single code. The library is cross-platform and free for use under the open-source license. Example of some supported functions are given bellow:
         * **Derivation**: Gradient / laplacian computing, contours delimitation
         * **Hough transforms:** lines, segments, circles, and geometrical shapes detection
         * **Histograms**: computing, equalization, and object localization with back projection algorithm
         * **Segmentation**: thresholding, distance transform, foreground

/ background detection, watershed segmentation

* + - * + **Filtering**: linear and nonlinear filters, morphological operations
        + **Cascade detectors**: detection of face, eye, car plates
        + **Interest points**: detection and matching
        + **Video processing:** optical flow, background subtraction, camshaft (object tracking)
        + **Photography**: panoramas realization, high definition imaging (HDR), image in painting.

So it was very important to install OpenCV. But installing OpenCV 2 is a complex process. How we did it is given below:

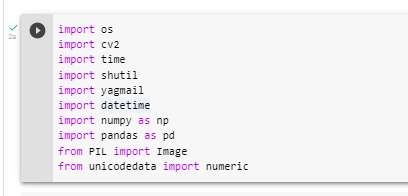


Fig: Installing OpenCV (cv2)

* + - 1. **Python IDE:** There are lots of IDEs for python. Some of them are PyCharm, Thonny, Ninja, Spyder etc. Ninja and Spyder both are very excellent and free but we used Google Colab as it is an online free platform.
      2. **Webcam:**

A webcam is a digital video device commonly built into a computer. Its main function is to transmit pictures over the Internet. It is popularly used with instant messaging services and for recording images.



Fig: Webcam

## Experimental Results

The steps of the experiments process are given below:

##### Face Detection:

Start capturing images through web camera of the client side:

* Pre-process the captured image and extract face image
* calculate the eigen value of the captured face image and compared with eigen values of existing faces in the database.
* If eigen value does not matched with existing ones, save the new face image information to the face database.
* If eigen value matched with existing one then recognition step will done.

##### Face Recognition:

Using PCA algorithm the following steps would be followed in for face recognition:

* Find the face information of matched face image in from the database.
* Update the log table with corresponding face image and system time that makes completion of attendance for an individual students.

This section presents the results of the experiment conducted to capture the face into a grey scale image of 50x50 pixels.

|  |  |  |
| --- | --- | --- |
| **Face Orientations** | **Detection Rate** | **Recognition Rate** |
| O0 (Frontal face) | 98.7 % | 95% |
| 18º | 80.0 % | 78% |
| 54º | 59.2 % | 58% |
| 72º | 0.00 % | 0.00% |
| 90º(Profile face) | 0.00 % | 0.00% |

The results that we achieved throughout the course of using this system from initiation through conclusion of developing this system the following results have been achieved. They are as follows:

* The system can be administered by a non-IT technician.
* The system is market ready for commercial use.
* The system has the capacity to carry up to mark the attendance by recognizing face.
* The system can serve as much people as they want within an organization.

## Future Scope of Work

To increase the scope of this device we can add some new features. As technology is becoming more advance it will be mandatory to change the structure some day with better replacement and sometimes based on customer requirements. There are so many future scopes on this project. Some of them are, it

* + - Can improve security
    - Can use Neural Network for high accuracy
    - Can used in big factory or employee attendance
    - Can build on fully web base system.

There are a lot of possibilities with the designed device in future applications. The device may need some research for different applications, though the principle of the designed system will remain as it is.

Chapter 4

# Code Implementation

## Code Implementation

All our code is written in Python language. First here is our project directory structure and files.

### main\_python\_file.ipynb

All the work will be done here, Detect the face ,recognize the faces and take attendance.

pip install yagmail

pip install os

import os

import cv2

import time

import shutil

import yagmail

import datetime

import numpy as np

import pandas as pd

from PIL import Image

from unicodedata import numeric

#@title Load libraries

!pip install -q face\_recognition

!pip install -q fer

import matplotlib.pyplot as plt

from matplotlib.pyplot import imshow

import matplotlib.patches as patches

%matplotlib inline

import face\_recognition

import numpy as np

from PIL import ImageDraw

import PIL.Image

from io import BytesIO

from fer import FER

import cv2

import operator

import time

from google.colab import files

from google.colab.patches import cv2\_imshow

detector = FER()

# Function to display program's title

def title():

# Clear the screen

  os.system('cls')

# Title of the program

  print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

  print("\*\*\*\*\* Facial Recognition Attendance System \*\*\*\*\*")

  print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

title()

# Function to display main menu

def main\_menu():

  title()

  print(10 \* "-", "MAIN MENU", 10 \* "-")

  print("[0] Wants to Recognize Face and Emotion")

  print("[1] Check Camera")

  print("[2] Mark Attendance")

  print("[3] Manage Students")

  print("[4] Manage Teachers")

  print("[5] Quit")

  while True:

    try:

      choice = int(input("Enter Choice: "))

      if choice == 1:

        check\_camera()

        break

      elif choice == 2:

        mark\_attendance()

        break

      elif choice == 3:

        manage\_students()

        break

      elif choice == 4:

        manage\_teachers()

        break

      elif choice == 5:

        print("Thank you")

        break

      else:

        print("Invalid Choice. Try Again")

        main\_menu()

    except ValueError:

      print("Invalid Choice. Try Again")

  exit()

main\_menu()

#function to recognize group of people

#@title Upload image with faces

uploaded = files.upload() #https://unsplash.com/photos/1qfy-jDc\_jo

#@title Display image

image = face\_recognition.load\_image\_file(list(uploaded.keys())[0])

boundary = 24

im = PIL.Image.open(list(uploaded.keys())[0])

height = (im.size[1] / im.size[0]) \* 20

fig=plt.subplots(figsize=(20,height))

imshow(im)

#@title Use 'Face Recognition' library and draw a red box around the faces

start = time.time()

face\_locations = face\_recognition.face\_locations(image)

fig,ax = plt.subplots(figsize=(20,height))

for face\_location in face\_locations:

    top, right, bottom, left = face\_location

    ax.imshow(im,aspect='auto')

    rect = patches.Rectangle((left,top), (right - left), (bottom-top),linewidth=3,edgecolor='r',facecolor='none')

    ax.add\_patch(rect)

end = time.time()

print("Number of faces:",len(face\_locations))

print("Time taken:",round(end-start,2)," seconds")

#@title Use 'Face Recognition' library and draw a red box around the faces as well as predicted emotion

start = time.time()

face\_locations = face\_recognition.face\_locations(image)

fig,ax = plt.subplots(figsize=(20,height))

for face\_location in face\_locations:

    top, right, bottom, left = face\_location

    ax.imshow(im,aspect='auto')

    rect = patches.Rectangle((left,top), (right - left), (bottom-top),linewidth=3,edgecolor='r',facecolor='none')

    ax.add\_patch(rect)

    face\_image = image[top-boundary:bottom+boundary, left-boundary:right+boundary]

    attribute = detector.detect\_emotions(face\_image)

    if (not attribute) == False:

      emotion = max(attribute[0]['emotions'].items(),key=operator.itemgetter(1))[0]

      #print(emotion)

      plt.text(left, top, emotion, fontsize=8, bbox=dict(fill=True, edgecolor='blue', linewidth=1))

end = time.time()

print("Number of faces:",len(face\_locations))

print("Time taken:",end-start," seconds")

# Function to check camera

def check\_camera():

 # Load the cascade

  face\_detector =cv2.CascadeClassifier('facial\_recognition/haarcascade\_frontalface\_default.xml')

  # To capture video from camera

  cap = cv2.VideoCapture(0, cv2.CAP\_DSHOW)

  if not cap.isOpened():

    print("There was an issue while opening the camera")

  while cap.isOpened():

    # Read the frame

    ret, img = cap.read()

    if ret:

      # Flips the original frame about y-axis

      img = cv2.flip(img, 1)

      # Convert to grayscale

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

      # Detect the faces

      faces = face\_detector.detectMultiScale(gray, 1.3, 5, minSize=(30, 30),

flags=cv2.CASCADE\_SCALE\_IMAGE)

      # Draw the rectangle around each face

      for (x, y, w, h) in faces:

        cv2.rectangle(img, (x, y), (x + w, y + h), (10, 159, 255), 2)

      # Display

      cv2.imshow('Camera Check', img)

      # Stop if escape key or 'q' is pressed

      if cv2.waitKey(1) & 0xFF == ord('q'):

        break

    else:

      break

  # Release the video capture object & destroy all windows

  cap.release()

  cv2.destroyAllWindows()

  key = input("Enter any key to return to main menu ")

  main\_menu()

# Function to check if input is a valid number

def is\_number(string: str):

  try:

    float(string)

    return True

  except ValueError:

    pass

  try:

    numeric(string)

    return True

  except (TypeError, ValueError):

    pass

# Function to return training images and labels

def imgs\_and\_labels(path: str):

  # Create empty list for faces

  faces = []

  # Create empty list for CMS IDs

  cms\_ids = []

  # Obtain a list of directories & files available inside the path

  \_, directories, \_ = next(os.walk(path))

  for directory in directories:

    # Obtain a list of files available within the subdirectory

    \_, \_, files = next(os.walk(path + '/' + directory))

    # Loop through each file within the subdirectory

    for file in files:

      # Load the image and convert it to gray scale

      pill\_img = Image.open(path + '/' + directory + '/' + file).convert('L')

      # Convert the PIL image into numpy array

      np\_img = np.array(pill\_img, 'uint8')

      # Get the CMS ID

      cms\_id = int(directory.split("\_")[-1])

      # Append the face to faces list

      faces.append(np\_img)

      # Append the cms\_id to CMS ids list

      cms\_ids.append(cms\_id)

  return faces, cms\_ids

# Function to view all students

def view\_students():

  if os.path.isfile("attendance/student\_details.csv"):

    student\_details = pd.read\_csv("attendance/student\_details.csv")

    if not student\_details.empty:

      print(student\_details)

    else:

      print("No students have been added yet.")

  else:

    print("No students have been added yet.")

# Function to add a student

def add\_student():

  cms\_id = input("CMS ID: ")

  while not is\_number(cms\_id):

    print("Please enter valid CMS ID")

    cms\_id = input("CMS ID: ")

  name = input("Name: ")

  print("Capturing the face...")

   # Make a folder for the student if it doesnot exist

  student = name.replace(" ", "\_") + "\_" + cms\_id

  if not os.path.isdir("facial\_recognition/faces/" + student):

    os.mkdir("facial\_recognition/faces/" + student)

  # Load the cascade

  face\_detector =cv2.CascadeClassifier("facial\_recognition/haarcascade\_frontalface\_default.xml")

  # To capture video from camera

  cap = cv2.VideoCapture(0, cv2.CAP\_DSHOW)

  if not cap.isOpened():

    print("There was an issue while opening the camera")

  sample\_num = 0

  while cap.isOpened():

   # Read the frame

    ret, img = cap.read()

    if ret:

      # Flips the original frame about y-axis

      img = cv2.flip(img, 1)

      # Convert to grayscale

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

      #detect the faces

      faces = face\_detector.detectMultiScale(gray, 1.3, 5, miniSize=(30, 30),flags=cv2.CASCADE\_SCALE\_IMAGE)

      for (x, y, w, h) in faces:

        # incement the sample number

        sample\_num += 1

        # Draw the rectangle around the face

        cv2.rectangle(img, (x, y), (x + w, y + h), (10, 159, 255), 2)

        cv2.imwrite("facial\_recognition/faces/" + student + "/" + str(sample\_num) + ".jpg", gray[y:y + h, x:x + w])

        # Display the frame

        cv2.imshow(student, img)

      # Stop after 100 milliseconds or if 'q' is pressed

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

        break

      # Exit if the number of samples is greater or equal to 100

      elif sample\_num >= 100:

        break

    else:

      break

  # Release the video capture object & destroy all windows

  cap.release()

  cv2.destroyAllWindows()

  # Save the student details

  if os.path.isfile("attendance/student\_details.csv"):

    student\_details = pd.read\_csv("attendance/student\_details.csv")

    student\_details = pd.concat([student\_details, pd.DataFrame({'CMS ID':[cms\_id], 'Name': [name]})],

                                ignore\_index=True,

                                axis=0)

    student\_details.drop\_duplicates(subset=['CMS ID'], inplace=True)

    student\_details.to\_csv('attendance/student\_details.csv', index=False)

  else:

    student\_details = pd.DataFrame(data={'CMS ID': [cms\_id], 'Name': [name]})

    student\_details.to\_csv('attendance/student\_details.csv', index=False)

  print("Training the model")

  # Train on the images & save the model

  recognizer = cv2.face\_LBPHFaceRecognizer.create()

  faces, cms\_ids = imgs\_and\_labels("facial\_recognition/faces")

  recognizer.train(faces, np.array(cms\_ids))

  recognizer.save("facial\_recognition/model.yml")

  print("Student Added!")

  key = input("Enter any key to return ")

  manage\_students

# Function to remove a student

def remove\_student():

  print("Enter CMS ID of the student to be removed")

  cms\_id = input("CMS ID: ")

  while not is\_number(cms\_id):

    print("Please enter valid CMS ID \n")

    cms\_id = input("CMS ID: ")

  # Read the student details

  students = pd.read\_csv("attendance/student\_details.csv")

  # If the student exists

  if not students[students['CMS ID'] == int(cms\_id)].empty:

    # Get the student's name

    name = students.loc[students['CMS ID'] == int(cms\_id)]['Name'].values[0]

    # Remove student's data from student\_details.csv

    students.drop(index=students[students["CMS ID"] == int(cms\_id)].index,

inplace=True)

    students.to\_csv("attendance/student\_details.csv", index=False)

    # Remove student's pictures

    try:

      shutil.rmtree("facial\_recognition/faces/" + name.replace(" ", "\_") + "\_" +

cms\_id)

    except OSError as e:

      print("Error: %s - %s." % (e.filename, e.strerror))

    # Retrain the model

    faces, cms\_ids = imgs\_and\_labels("facial\_recognition/faces")

    if len(faces) != 0 and len(cms\_ids) != 0:

      recognizer = cv2.face\_LBPHFaceRecognizer.create()

      recognizer.train(faces, np.array(cms\_ids))

      recognizer.save("facial\_recognition/model.yml")

    print("Student Removed!")

  else:

    print("Student not found")

  key = input("Enter any key to return ")

  manage\_students()

# Function to manage students

def manage\_students():

  print(10 \* "-", "MANAGE STUDENTS", 10 \* "-")

  print("[1] View all Students")

  print("[2] Add a Student")

  print("[3] Remove a Student")

  print("[4] Go to Main Menu")

  while True:

    try:

      choice = int(input("Enter Choice: "))

      if choice == 1:

        view\_students()

        manage\_students()

        break

      elif choice == 2:

        add\_student()

        break

      elif choice == 3:

        view\_students()

        remove\_student()

        break

      elif choice == 4:

        main\_menu()

        break

      else:

        print("Invalid Choice. Try Again!")

        main\_menu()

    except ValueError:

      print("Invalid Choice. Try Again!")

  exit()

# Function to view all teachers

def view\_teachers():

  if os.path.isfile("attendance/teacher\_details.csv"):

    teacher\_details = pd.read\_csv("attendance/teacher\_details.csv")

    if not teacher\_details.empty:

      print(teacher\_details)

    else:

      print("No teachers have been added yet.")

  else:

    print("No teachers have been added yet.")

# Function to add a teacher

def add\_teacher():

  name = input("Name: ")

  name = name.lower()

  email = input("Email Address: ")

  pwd = input("Set a password: ")

  # Save the teacher details

  if os.path.isfile("attendance/teacher\_details.csv"):

    teacher\_details = pd.read\_csv("attendance/teacher\_details.csv")

    teacher\_details = pd.concat([teacher\_details, pd.DataFrame({'Name': [name],

'Email Address': [email],

'Password': [pwd]})],

ignore\_index=True,

axis=0)

    teacher\_details.drop\_duplicates(subset=['Name'], inplace=True)

    teacher\_details.to\_csv('attendance/teacher\_details.csv', index=False)

  else:

    teacher\_details = pd.DataFrame(data={'Name': [name], 'Email Address':

[email], 'Password': [pwd]})

    teacher\_details.to\_csv('attendance/teacher\_details.csv', index=False)

  print("Teacher Added!")

  key = input("Enter any key to return ")

  manage\_teachers()

# Function to remove a teacher

def remove\_teacher():

  print("Enter the name of the teacher to be removed")

  name = input("Name: ")

  name = name.lower()

  # Read the student details

  teachers = pd.read\_csv("attendance/teacher\_details.csv")

  # If the student exists

  if not teachers[teachers['Name'] == name].empty:

    # Remove student's data from student\_details.csv

    teachers.drop(index=teachers[teachers["Name"] == name].index, inplace=True)

    teachers.to\_csv("attendance/teacher\_details.csv", index=False)

    print("Teacher Removed!")

  else:

    print("Teacher not found")

  key = input("Enter any key to return ")

  manage\_teachers()

# Function to manage teachers

def manage\_teachers():

  print(10 \* "-", "MANAGE TEACHERS", 10 \* "-")

  print("[1] View all Teachers")

  print("[2] Add a Teacher")

  print("[3] Remove a Teacher")

  print("[4] Go to Main Menu")

  while True:

    try:

      choice = int(input("Enter Choice: "))

      if choice == 1:

        view\_teachers()

        manage\_teachers()

        break

      elif choice == 2:

        add\_teacher()

        break

      elif choice == 3:

        view\_teachers()

        remove\_teacher()

        break

      elif choice == 4:

        main\_menu()

        break

      else:

        print("Invalid Choice. Try Again!")

        main\_menu()

    except ValueError:

      print("Invalid Choice. Try Again!")

  exit()

# Function to verify the teacher

def teacher\_verfication(name: str, pwd: str):

  if os.path.isfile("attendance/teacher\_details.csv"):

    teacher\_details = pd.read\_csv("attendance/teacher\_details.csv")

    if not teacher\_details[(teacher\_details['Name'] == name) &

(teacher\_details['Password'] == pwd)].empty:

      return True

    else:

      print("No teachers have been added yet.")

  else:

    print("No teachers have been added yet.")

# Function to email the attendance

def email\_attendance(teacher\_name: str, sheet\_path: str):

  teacher\_details = pd.read\_csv("attendance/teacher\_details.csv")

  receiver = teacher\_details.loc[teacher\_details['Name'] == teacher\_name]['EmailAddress'].values[0]

  body = "Attendance sheet is attached along with this email." # email body

  # Sender information

  yag = yagmail.SMTP("", "")

  # Send the email to teacher

  yag.send(

    to=receiver,

    subject=teacher\_name + " - Attendance Report", # email subject

    contents=body, # email body

    attachments=sheet\_path, # file attached

  )

  # Send the email to HOUGP EPE

  yag.send(

    to="ayahya@pnec.nust.edu.pk",

    subject=teacher\_name + " - Attendance Report", # email subject

    contents=body, # email body

    attachments=sheet\_path, # file attached

  )

# Function to mark attendance

def mark\_attendance():

  teacher\_name = input("Teacher Name: ")

  teacher\_name = teacher\_name.lower()

  pwd = input("Password: ")

  if teacher\_verfication(name=teacher\_name, pwd=pwd):

    # Load the face recognizer

    recognizer = cv2.face\_LBPHFaceRecognizer.create()

    recognizer.read("facial\_recognition/model.yml")

    # Load the cascade

    face\_detector =cv2.CascadeClassifier("facial\_recognition/haarcascade\_frontalface\_default.xml")

    students = pd.read\_csv("attendance/student\_details.csv")

    col\_names = ['CMS ID', 'Name', 'Date', 'Time']

    attendance = pd.DataFrame(columns=col\_names)

    font = cv2.FONT\_HERSHEY\_SIMPLEX

    # To capture video from camera

    cap = cv2.VideoCapture(0, cv2.CAP\_DSHOW)

    cap.set(3, 640) # set video width

    cap.set(4, 480) # set video height

    # Min window size to be recognized as a face

    min\_w = 0.1 \* cap.get(3)

    min\_h = 0.1 \* cap.get(4)

    if not cap.isOpened():

      print("There was an issue while opening the camera")

    while cap.isOpened():

      # Read the frame

      ret, img = cap.read()

      if ret:

        # Flips the original frame about y-axis

          img = cv2.flip(img, 1)

          # Convert to grayscale

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

          # Detect the faces

          faces = face\_detector.detectMultiScale(gray, 1.2, 5,minSize=(int(min\_w), int(min\_h)),flags=cv2.CASCADE\_SCALE\_IMAGE)

          conf\_thresh = 65

          for (x, y, w, h) in faces:

            # Draw the rectangle around the face

            cv2.rectangle(img, (x, y), (x + w, y + h), (10, 159, 255), 2)

            cms\_id, conf = recognizer.predict(gray[y:y + h, x:x + w])

            if conf < 100:

              name = students.loc[students['CMS ID'] == cms\_id]['Name'].values

              confstr = " {0}%".format(round(100 - conf))

              student = str(cms\_id) + "-" + name

            else:

              cms\_id = ' Unknown '

              student = str(cms\_id)

              confstr = " {0}%".format(round(100 - conf))

            if (100 - conf) > conf\_thresh:

              ts = time.time()

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

              time\_stamp =datetime.datetime.fromtimestamp(ts).strftime('%H:%M:%S')

              name = str(name)[2:-2]

              attendance.loc[len(attendance)] = [cms\_id, name, date, time\_stamp]

            student = str(student)[2:-2]

            if (100 - conf) > conf\_thresh:

              cv2.putText(img, str(student), (x + 5, y - 5), font, 1, (255, 255,

255), 2)

            if (100 - conf) > conf\_thresh:

                cv2.putText(img, str(confstr), (x + 5, y + h - 5), font, 1, (0,

255, 0), 1)

            elif (100 - conf) > 50:

                cv2.putText(img, str(confstr), (x + 5, y + h - 5), font, 1, (0,

255, 255), 1)

            else:

              cv2.putText(img, str(confstr), (x + 5, y + h - 5), font, 1, (0, 0,

255), 1)

          attendance = attendance.drop\_duplicates(subset=['CMS ID'], keep='first')

          cv2.imshow('Attendance', img)

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

            break

      else:

          break

    ts = time.time()

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

    time\_stamp = datetime.datetime.fromtimestamp(ts).strftime('%H:%M:%S')

    hour, minute, second = time\_stamp.split(":")

    file\_name = "attendance/sheets/" + teacher\_name + "\_attendance\_" + date + "\_"+ hour + "-" + minute + ".csv"

    attendance.to\_csv(file\_name, index=False)

    cap.release()

    cv2.destroyAllWindows()

    # Email the attendance

    # email\_attendance(teacher\_name=teacher\_name, sheet\_path=file\_name)

    print("Attendance Successful!")

  key = input("Enter any key to return to main menu ")

  main\_menu()

if \_\_name\_\_ == "\_\_main\_\_":

  main\_menu()

**Extra Feature :**

### Mail function

It will send the attendance file to specific mail. Mail code given below:

# Function to email the attendance

def email\_attendance(teacher\_name: str, sheet\_path: str):

  teacher\_details = pd.read\_csv("attendance/teacher\_details.csv")

  receiver = teacher\_details.loc[teacher\_details['Name'] == teacher\_name]['EmailAddress'].values[0]

  body = "Attendance sheet is attached along with this email." # email body

  # Sender information

  yag = yagmail.SMTP("", "")

  # Send the email to teacher

  yag.send(

    to=receiver,

    subject=teacher\_name + " - Attendance Report", # email subject

    contents=body, # email body

    attachments=sheet\_path, # file attached

  )

  # Send the email to HOUGP EPE

  yag.send(

    to="ayahya@pnec.nust.edu.pk",

    subject=teacher\_name + " - Attendance Report", # email subject

    contents=body, # email body

    attachments=sheet\_path, # file attached

  )

Chapter 5

# System Outputs

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  | |
|  |  |
|  |  |
|  |  |
|  |  |
|  | |