



# **MAJOR PROJECT**

## **FACE RECOGNITION BASED ATTENDENCE SYSTEM**

### **Diploma in Electronics Engineering**

By

**Harsh Raja**

**20DPEL206**

**GM4262**

**Ritik Upadhyay**

**20DPEL212**

**GM4345**

**Mohd. Saqib**

**20DPEL216**

**GM4353**

**Mohd. Azfaar**

**20DPEL228**

**GM4337**

**Pushkar Varshney**

**20DPEL230**

**GM4222**

**Eshaw Saxena**

**20DPEL231**

**GM4331**

**UNDER THE GUIDANCE OF**

**Mr. Tanveer Hasan**

**E-mail:-thasan@myamu.ac.in**

**(Associate professor, University Polytechnic AMU)**



# **DIPLOMA IN ENGINEERING**

**University Polytechnic AMU**

## **CERTIFICATE**

This is to certify that the Major project entitled "**Face Recognition based Attendance system**" being submitted by **Harsh Raja, Ritik Upadhyay, Mohd. Saqib, Mohd. Azfaar khan, Pushkar Varshney, Eshav Saxena** from Diploma in Electronics Engineering, University Polytechnic have successfully completed their project under our supervision and guidance during session 2022-2023.

---

**Mr. Tanveer Hasan**

**(Associate professor, University Polytechnic AMU)**

## ACKNOWLEDGEMENT

This project is done as a semester major project, as a part course titled “Project and Seminar Lab-II PC0694C”. We are really thankful to our course the Principal Prof. Arshad Umar and the HOD Ms. Kaushar Jahan, Electrical Engineering Section, University Polytechnic, Aligarh Muslim University, Aligarh for their invaluable guidance and assistance, without which the accomplishment of the task would have never been possible.

We also thank Mr.Tanveer Hasan for giving this opportunity to explore into the real world and realize the interrelation without which a Project can never progress. In our present project we have chosen the topic: “FACE RECOGNITION BASED ATTENDANCE SYSTEM”.

We are also thankful to parents, friends and all staff of Electrical Engineering Department, for providing us relevant information and necessary clarifications, and great support.

We are also thankful to parents, friends and all staff of **Electrical Engineering** Department, for providing us relevant information and necessary clarifications, and great support.

## **PURPOSE AND GOAL**

- 1. To create a system that can automatically take attendance in a business or classroom.**
- 2. To increase attendance records' accuracy and dependability by doing away with manual record-keeping.**
- 3. To make taking attendance easier and less expensive by reducing the time and resources needed.**
- 4. To offer a practical and simple solution that can manage big classrooms or workplaces.**
- 5. To use machine learning and computer vision techniques to instantly track attendance and recognise faces.**

The "**Face Recognition Based Attendance System**" project's goal is to create an automated system for taking attendance that can recognise faces and log attendance in real-time.

The system's goal is to offer a trustworthy, accurate, and effective substitute for manual attendance-taking techniques, which can be time-consuming, prone to error, and resource-intensive.

- By doing away with the necessity for manual record-keeping and minimising the time and resources needed for attendance-taking, the system attempts to streamline the taking of attendance process.
- By utilising computer vision and machine learning techniques to recognise faces and record attendance in real-time, the system also intends to increase the precision and dependability of attendance records.

- By utilising computer vision and machine learning techniques to recognise faces and record attendance in real-time, the system also intends to increase the precision and dependability of attendance records.

The system's goal is to offer a convenient and user-friendly system that can effectively manage huge courses or workplaces.

- By using suitable data protection mechanisms, the system also strives to safeguard the security and privacy of personal information.

## THE SCOPE OF THE PROJECT

The "Face Recognition Based Attendance System" project has a broad scope that covers a variety of topics, including software engineering, machine learning, and computer vision. The project's primary scopes are as follows:

 **Automation of the attendance process:** The system's goal is to automate the taking of attendance, therefore minimising the need for manual record-keeping and increasing the precision and effectiveness of attendance records.

 **Face recognition:** The system recognises faces and compares them to faces in its database using computer vision and machine learning algorithms. The capabilities of the system include locating and extracting facial characteristics, creating a deep learning model, and real-time face recognition.

Integrating the system with other systems and technologies, such as learning management systems and biometric identification systems, is part of the system's scope.

The "Face Recognition Based Attendance System" project's overall goal is to develop an automated attendance-taking system that is trustworthy, accurate, and efficient and makes use of several computer vision and machine learning algorithms, Python libraries, and software engineering concepts. Additionally, the system's scope entails safeguarding the security and privacy of user data and offering a user-friendly interface that can adapt to various classroom and office environments.

## **INTEGRATED DEVELOPMENT ENVIRONMENT:-**

An **integrated development environment (IDE)** is a [software application](#) that provides comprehensive facilities to [computer programmers](#) for [software development](#). An IDE normally consists of at least a [source code editor](#), [build automation](#) tools, and a [debugger](#). Some IDEs, such as [NetBeans](#) and [Eclipse](#), contain the necessary [compiler](#), [interpreter](#), or both; others, such as [SharpDevelop](#) and [Lazarus](#), do not.

An integrated development environment (IDE) is a software application that provides comprehensive facilities to computer programmers for software development. An IDE normally consists of at least a source code editor, build automation tools, and a debugger.

The boundary between an IDE and other parts of the broader software development environment is not well-defined. Many modern IDEs also have a class browser, an object browser, and a class hierarchy diagram for use in object oriented software development.

## THE IDE USED

### VISUAL STUDIO CODE :-

Visual Studio Code, also commonly referred to as VS Code,<sup>[9]</sup> is a [source-code editor](#) made by [Microsoft](#) for [Windows](#), [Linux](#) and [macOS](#).<sup>[10]</sup> Features include support for [debugging](#), [syntax highlighting](#), [intelligent code completion](#), [snippets](#), [code refactoring](#), and embedded [Git](#).

Users can change the [theme](#), [keyboard shortcuts](#), preferences, and install [extensions](#) that add functionality. Visual Studio Code is a source-code editor that can be used with a variety of programming languages, including [C](#), [C#](#), [C++](#), [Fortran](#), [Go](#), [Java](#), [JavaScript](#), [Node.js](#), [Python](#). Out of the box, Visual Studio Code includes basic support for most common programming languages.

This basic support includes [syntax highlighting](#), [bracket matching](#), [code folding](#), and configurable snippets. Visual Studio Code also ships with [IntelliSense](#) for JavaScript, TypeScript, [JSON](#), [CSS](#), and [HTML](#), as well as debugging support for Node.js. Support for additional languages can

## THE CODE

```
import face_recognition  
import cv2  
import numpy as np  
import csv  
import os  
from datetime import datetime  
  
video_capture = cv2.VideoCapture(0)  
  
harsh2_image =  
face_recognition.load_image_file("photos/harsh2.jpg")  
harsh2_encoding =  
face_recognition.face_encodings(harsh2_image)[0]  
  
harsh_image =  
face_recognition.load_image_file("photos/harsh.jpg")  
harsh_encoding =  
face_recognition.face_encodings(harsh_image)[0]
```

```
ratan_tata_image =  
face_recognition.load_image_file("photos/ratan_tata.jpg"  
)
```

```
ratan_tata_encoding =  
face_recognition.face_encodings(ratan_tata_image)[0]
```

```
known_face_encoding =
```

```
[  
    harsh2_encoding,  
    harsh_encoding,  
    ratan_tata_encoding  
]
```

```
known_faces_names = ["harsh2", "harsh", "ratan_tata"]
```

```
students = known_faces_names.copy()
```

```
face_locations = []
```

```
face_encodings = []
```

```
face_names = []
```

```
s = True
```

```
now = datetime.now()
current_date = now.strftime("%Y-%m-%d")

f = open(current_date+'.csv','w+',newline="")
lnwriter = csv.writer(f)

while True:
    frame = video_capture.read()
    small_frame = cv2.resize(frame,(0,0),fx = 0.25,fy = 0.25)
    rgb_small_frame = small_frame[:, :, :: -1]

    if s :
        face_locations=
        face_recognition.face_locations(rgb_small_frame)
        face_encodings=
        face_recognition.face_encodings(rgb_small_frame,face_l
ocations)

        face_names = []
        for face_encoding in face_encodings:
            matches=
            face_recognition.compare_faces(known_face_encoding,f
ace_encoding)
```

```
name = ""

face_distance=
face_recognition.face_distance(known_face_encoding,
face_encoding)

best_match_index = np.argmin(face_distance)

if matches[best_match_index]:
    name= known_faces_names[best_match_index]

face_names.append(name)

if name is known_faces_names:
    if name in students:
        students.remove(name)
    print(students)

current_time = now.strftime("%H-%M-%S")

lnwriter.writerow([name,current_time])

cv2.imshow("attendance system",frame)

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

```
video_capture.release()  
cv2.destroyAllWindows()  
f.close()
```

## EXPLANATION

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

### 1. Importing the Necessary Libraries:

**Import** `face_recognition`, **import** `cv2`, **import** `numpy as np`, **import** `csv`, **import** `os` from `datetime`, **import** `datetime` in Python.

The code first imports the required libraries:  
`'face_recognition'` is a library used for facial recognition tasks.

⊕ `'cv2'`(OpenCV) is used for processing images and videos.

⊕ `'numpy'` is a library for performing numerical calculations.

⊕ The `'csv'` command is used to read and write CSV files.

⊕ `'os'` offers a method of interacting with the operating system.

- ✿ 'datetime' is used to get the time and date right now.

## **2. Using Python, set up video capture using cv2.VideoCapture(0).**

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

Here, a video capture object is made to have access to the default camera (number 0), which will record video frames.

## **3. Loading and Encoding Face Images:**

```
harsh2_image = face_recognition.load_image_file("photos/harsh2.jpg")
harsh2_encoding = face_recognition.face_encodings(harsh2_image)[0]

harsh_image = face_recognition.load_image_file("photos/harsh.jpg")
harsh_encoding = face_recognition.face_encodings(harsh_image)[0]

ratan_tata_image = face_recognition.load_image_file("photos/ratan_tata.jpg")
ratan_tata_encoding = face_recognition.face_encodings(ratan_tata_image)[0]
```

The 'face\_recognition' library is used in this section's code to load facial images of recognised people from files and encode them into numerical feature vectors. These feature vectors, which reflect each face's distinctive features, enable face comparison and recognition.

As we have used one of our Companion Harsh Raja's Photo.

#### **4. Creating Lists of Known Face Encodings and Names:**

```
known_face_encoding = [  
    harsh2_encoding,  
    harsh_encoding,  
    ratan_tata_encoding  
]  
  
known_faces_names = [  
    "harsh2",  
    "harsh",  
    "ratan_tata"  
]
```

Here, the code generates two lists: "known\_face\_encoding" and "known\_faces\_names," which both contain the names of each face encoding and the face encodings of known individuals.

#### **5. Initializing Variables:**

```
students = known_faces_names.copy()  
  
face_locations = []  
face_encodings = []  
face_names = []  
s = True
```

Variables for recording attendance are initialised in the code. The list of students who have not yet been marked present is represented by the word "students," which is a copy of "known\_faces\_names."

The results of face detection and names that were identified for each frame are stored in the variables "face\_locations," "face\_encodings," and "face\_names." Variable's 's' represents a flag.

## **6. Creating and Opening CSV File:**

```
now = datetime.now()
current_date = now.strftime("%Y-%m-%d")

f = open(current_date+'.csv', 'w+', newline='')
lnwriter = csv.writer(f)
```

The code uses the '[datetime](#)' module to get the current date and time. '[current\\_date](#)' is formatted as "YYYY-MM-DD". A CSV file is created with the name based on the current date, and a writer object '[lnwriter](#)' is initialized for writing rows to the CSV file.

## **7. Main Loop: Face Recognition and Attendance Tracking**

```
while True:  
    _,frame = video_capture.read()  
    small_frame = cv2.resize(frame,(0,0),fx = 0.25,fy = 0.25)  
    rgb_small_frame = small_frame[:, :, ::-1]  
    if s:  
        face_locations = face_recognition.face_locations(rgb_small_frame)  
        face_encodings = face_recognition.face_encodings(rgb_small_frame,face_locations)  
        face_names = []  
        for face_encoding in face_encodings:  
            matches = face_recognition.compare_faces(known_face_encoding,face_encoding)  
            name = ""  
            face_distance = face_recognition.face_distance(known_face_encoding,face_encoding)  
            best_match_index = np.argmin(face_distance)  
            if matches[best_match_index]:  
                name = known_faces_names[best_match_index]  
  
                face_names.append(name)  
                if name in known_faces_names:  
                    if name in students:  
                        students.remove(name)  
        print(students)  
        current_time = now.strftime("%H-%M-%S")  
        lnwriter.writerow([name,current_time])
```

Face recognition and attendance monitoring are carried out by the code entering a loop in this part. As for what each component does:

- The video capture object ('video\_capture') retrieves a frame from the camera and assigns it to the variable 'frame'.
- In order to enhance face recognition performance, the captured frame is reduced in size using the 'cv2.resize' command. The resulting smaller frame is kept in the variable "small\_frame."

As required by the "face\_recognition" library, "small\_frame" is converted from BGR to RGB format to create "rgb\_small\_frame."

- + If "s," which is initially set to "True," is true, face detection and recognition are carried out on the "rgb\_small\_frame."
- + "face\_locations" is modified to reflect the faces that were found in the frame's top, right, bottom, and left places.
- + The face encodings for each recognised face are contained in the variable "face\_encodings."

## 8. Displaying the Video Frame:

```
cv2.imshow("attendance system",frame)
if cv2.waitKey(1) & 0xFF == ord('q'):
    break
```

This code displays the current video frame in a window titled "attendance system" using `cv2.imshow`. If the 'q' key is pressed, the loop is exited and the program terminates.

## **9. Releasing Resources and Closing the CSV File:**

```
video_capture.release()  
cv2.destroyAllWindows()  
f.close()
```

After exiting the loop, the video capture object is released using 'video\_capture.release()' to free the camera resources. The OpenCV windows are closed with 'cv2.destroyAllWindows()'. Finally, the CSV file is closed using 'f.close()'.

That's a breakdown of the code section by section, explaining its functionality and the purpose of each step.

## RESULTS OF EXECUTION

```
File Edit Selection View Go Run Terminal Help project3.py - python - Visual Studio Code

project3.py X

project3.py > ...
1 import face_recognition
2 import cv2
3 import numpy as np
4 import csv
5 import os
6 from datetime import datetime
7
8 video_capture = cv2.VideoCapture(0)
9
10 harsh_image = face_recognition.load_image_file("photos/harsh.jpg")
11 harsh_encoding = face_recognition.face_encodings(harsh_image)[0]
12
13 ratan_tata_image = face_recognition.load_image_file("photos/ratan_tata.jpg")
14 ratan_tata_encoding = face_recognition.face_encodings(ratan_tata_im
15

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

PS C:\python> python -u "c:\python\project2.py"
PS C:\python> python -u "c:\python\project3.py"
['ratan tata']
[]
```

The face of one of our group member is recognized as can be seen above. The code presents the name of the student or the Candidate in front of the camera and the attendance is also marked in a CSV file as below :-

A	B
1 harsh	14-46-07
2 ratan tata	14-46-07
3	
4	
5	
6	
7	
8	
9	

## REFERENCES

- <https://www.programiz.com/python-programming>
- <https://www.tutorialspoint.com/image-processing-in-python>
- <https://www.lystloc.com/blog/what-is-facial-recognition-attendance-system-and-its-top-benefits-in-the-workplace/>
- <https://www.tutorialspoint.com/dip/index.htm>