

# Facial Recognition Attendance System

Rohit Sharma<sup>1</sup>, Sanvida Badhe<sup>2</sup>, Achal Patil<sup>3</sup>, Akshata Prabhu<sup>4</sup>

<sup>1,2,3</sup>U.G. Student, Electronics and Telecommunication Engineering, K.J. Somaiya College of Engineering, Mumbai, India

<sup>4</sup>Assistant Professor, Electronics and Telecommunication Engineering, K.J. Somaiya College of Engineering, Mumbai, India

\*\*\*

**Abstract** - A critical event in the education sector, such as attendance, should not be handled in the old boring conventional way in this age of rapidly evolving new technologies. For both the teaching staff and the students, an attendance management system would save a lot of time and resources. The face recognition algorithm will monitor attendance by distinguishing only the students' faces from the rest of the objects and then labelling them as present. The device will be pre-loaded with photographs of all of the students enrolled in the class, and the algorithm will use this data to detect the students who are present and compare their features to those already saved in the database.

**Key Words:** Face Recognition, Attendance Monitoring System, Django database, Histogram of Oriented Gradients (HOG), SVM Classifier Algorithm

## 1. INTRODUCTION

These days, technology strives to impart enormous knowledge-based technological advances. The traditional method of recording attendance consists of students sitting in a class and the teacher/professor recording attendance with a pen or paper. This is inconvenient, increases the use of paper, and exposes students' proxies, so to address this, a new attendance technology is being implemented that will use students' faces to label attendance. With the advent in technology, the computer now automatically senses the students' attendance output and keeps a record of the data obtained. The use of face recognition for attendance marking is a clever way of implementing an attendance management scheme. Face recognition, among other methods, is more reliable and quicker, and it decreases the risk of proxy attendance. Face recognition allows for passive identification, which means that the person being recognised does not have to take any action in order to be identified. An idea of building a technology that can help in making attendance marking systems fully automated and even in the darkest of times like a pandemic situation present now where social distancing is required, contactless attendance can be maintained[2].

In our university, we currently use a manual attendance tracking system. As we all know, the manual student attendance control system has a number of flaws. The following are some of the issues:

- Disturbance to both the professor and the students
- Possibility of misplacing the attendance sheets
- Difficulty in analysing
- Collecting attendance from a large number of students is difficult[1].

So, to overcome these problems, an automated attendance system using face recognition will be very helpful and effective.

The main aims to carry forward in this project are:

- To make a fully automated and more efficient attendance marking system.
- To maintain social distancing and contactless attendance.
- To make a login website for students to view their attendance and for a professor to maintain the record of it.

## 2. LITERATURE REVIEW

### 2.1 Face Recognition

Automatic Attendance System using Face Recognition proposes that the system is based on face detection and recognition algorithms, which is used to automatically detect the student face when he/she enters the class and the system is capable of marking the attendance by recognizing him. The algorithm used for detection and recognition is HOG (Histogram of Oriented Gradients).

The Histogram of Oriented Gradients (HOG) is a function descriptor for object detection in computer vision and image processing. The technique counts the number of times a gradient orientation appears in a specific area of an image. Edge orientation histograms, scale-invariant feature transform descriptors, and shape contexts are all related methods, but this one differs in that it is computed on a dense grid of uniformly spaced cells and uses overlapping local contrast normalization for better accuracy.

HOG working is as follows:

- Finding the faces: Histogram of Oriented Gradients(HOG). Images are converted to black and white. Image is broken into 16\*16 pixels each to count gradient point in each major direction. This is how we detect faces.
- Posing and Projecting faces: It warp images. Algorithm used in this is face landmark estimation. We use 68 landmark points to warp faces.
- Encoding faces: In this, 128 measurements for each face, like distance between eyes, nose, mouth, etc. are taken. These 128 measurements are called an embedding.
- Finding person's name from encoding: This is done using SVM classifier algorithm. It compares the encodings of each face captured in the camera feed with the faces or images which are already stored in the database. After this matching process, the name of the respective person is returned[3].

## 2.2 Django based approach

Django is a high-level Python web framework that enables rapid development of secure and maintainable websites. Built by experienced developers, Django takes care of much of the hassle of web development, so you can focus on writing your app without needing to reinvent the wheel. It is free and open source, has a thriving and active community, great documentation, and many options for free and paid-for support[4].

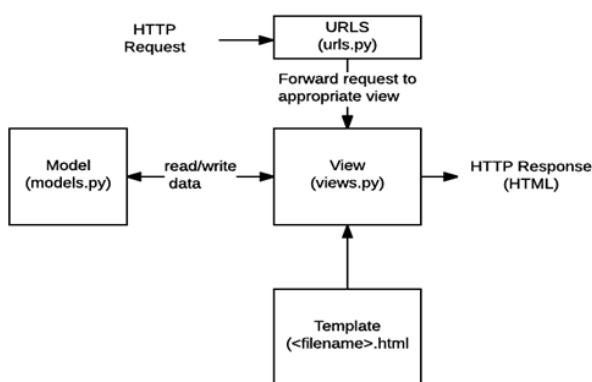


Fig 1. Django Architecture

## 2.3 Raspberry Pi and Pi Camera

RASPBERRY PI 3B is a development board in PI series. It can be considered as a single board computer that works on the LINUX operating system. The board not only has tons of

features it also has terrific processing speed making it suitable for advanced applications. The simplest use for a Raspberry Pi is as a desktop computer. Along with the Pi itself, the microSD card, and power supply, you'll need a HDMI cable and a suitable display. As with a traditional computer, you'll also need a USB keyboard and mouse. The Raspberry Pi 3B and later has Wi-Fi and Bluetooth built in[5].

The Pi camera module is a portable lightweight camera that supports Raspberry Pi. It communicates with Pi using the MIPI camera serial interface protocol. It is normally used in image processing, machine learning or in surveillance projects. It is commonly used in surveillance drones since the payload of the camera is very less. Apart from these modules Pi can also use normal USB webcams that are used along with computers[6].

## 2.4 Visual Studio 2019

Microsoft Visual Studio is the company's integrated production environment (IDE). It's used to make blogs, web applications, web services, and smartphone apps, among other things. Windows API, Windows Shapes, Windows Presentation Foundation, Windows Store, and Microsoft Silverlight are some of the Microsoft software development tools used by Visual Studio. It has the ability to generate both native and controlled code.

Visual Studio comes with a code editor that supports IntelliSense (code completion) and code refactoring. The integrated debugger can be used as both a source-level and a machine-level debugger. A code profiler, a designer for creating GUI applications, a web designer, a class designer, and a database schema designer are among the other built-in tools[7].

## 2.5 HTML

The HyperText Markup Language, or HTML is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript.

Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document[8].

## 2.6 CSS

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language such as HTML. CSS is a cornerstone

technology of the World Wide Web, alongside HTML and JavaScript.

CSS is designed to enable the separation of presentation and content, including layout, colours and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file which reduces complexity and repetition in the structural content as well as enabling the .css file to be cached to improve the page load speed between the pages that share the file and its formatting[9].

### 3. PROJECT DESIGN

As the idea behind this project is to make our traditional attendance system fully automated, we will be requiring both hardware and software. Hardware will contain the Raspberry Pi and Pi camera module or else some other camera which includes the laptop in-built camera or web-cam. Software used here is Django, Visual Studio 2019 and VNC Viewer. With the help of these hardware components and software, we will be able to achieve our goal of making an automated attendance marking system.

#### 3.1 Block Diagram

The functioning of the entire system is as follows:

- First the lecture is started by the professor.
- The camera then detects whether the person's face is present or not.
- After detection, the detected face is being compared with the images in the database.
- When the face is recognized, student's attendance is marked and it is updated in the back-end.
- Students can then see his/her attendance marked in the login portal.
- The faculty can download the attendance sheet after logging into the portal whenever required.

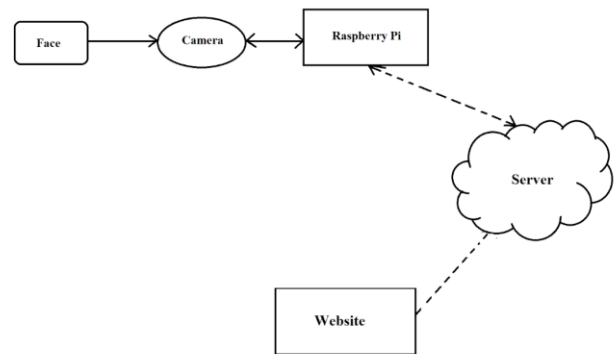


Fig 2. Block Diagram

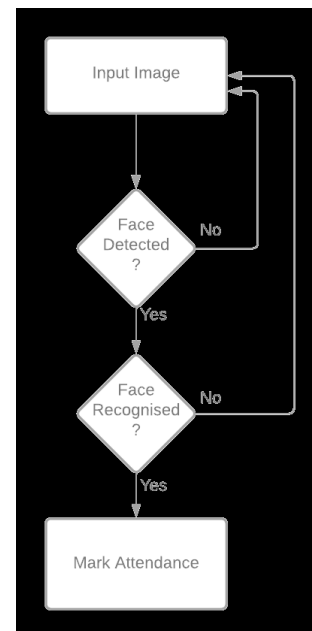


Fig 3. Flowchart

#### 3.2 Objectives

- Face Detection - To detect faces correctly
- Face Recognition - To compare detected faces with database and give accurate results
- Mark Attendance - To update the attendance in database and website
- Login Portal - To make portal for students as well as professors for viewing and managing attendance

## 4. IMPLEMENTATION AND EXECUTION

### 4.1 Face Recognition Using Python

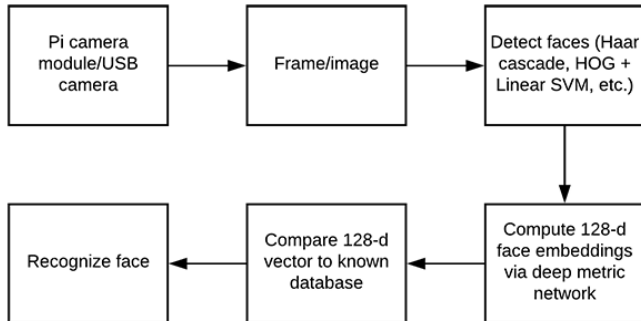


Fig 4. Face Recognition Flowchart

By looking at the flowchart, we get to know how a face is being recognised. The important part of facial recognition is finding encodings of the face and then comparing it with the encodings of images in the database.

#### 4.1.1 Training dataset with encodings

To make the process of comparing encodings faster, we have trained our dataset with encodings of images already. Below is the pseudo code of how we trained our dataset with encodings.

```

def findEncodings(images):
    encodeList=[]
    count = 0
    for img in images:
        try:
            img=cv2.cvtColor(img,cv2.COLOR_BGR2RGB)
            encode=fc.face_encodings(img)[0]
            encodeList.append(encode)
            print(count)
        except:
            print("Problem with image",count)
            count+=1
    return encodeList

encodeListKnown=findEncodings(images)

data = {"encodings": encodeListKnown, "names": classNames}
f = open("face_enc", "wb")
f.write(pickle.dumps(data))
f.close()
print('Encoding Complete')
  
```

Fig 5. Training Dataset Pseudo Code

#### 4.1.2 Face Recognition Implementation

After training the dataset, now we have to recognize faces and compare it with the database. Pseudo code is given below.

```

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

    faceCurFrame=fc.face_locations(imgS)
    encodeCurFrame=fc.face_encodings(imgS,faceCurFrame)

    for encodeFace,faceLoc in zip(encodeCurFrame,faceCurFrame):
        matches=fc.compare_faces(encodeListKnown,encodeFace,tolerance=0.4)
        faceDis=fc.face_distance(encodeListKnown,encodeFace)
        matchIndex=np.argmax(faceDis)

        if matches[matchIndex]:
            id,name=classNames[matchIndex].upper().split('_')
            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)
            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(id,name)
  
```

Fig 6. Find Faces

### 4.2 Database Using Django

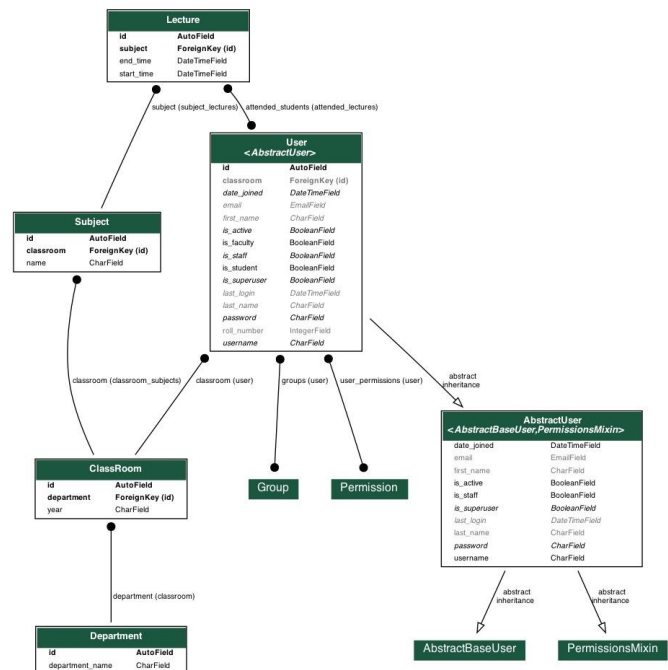


Fig 7. ER Diagram of Database

### 4.3 Web Portal

In our case, a web portal is used for students and faculties to login to their dashboard for viewing the attendance or managing the lecture schedule respectively.

#### 4.3.1 Login Page

Login page helps both faculties and students to login to their respective accounts for viewing and managing attendance. To get a better understanding, here is our login page:

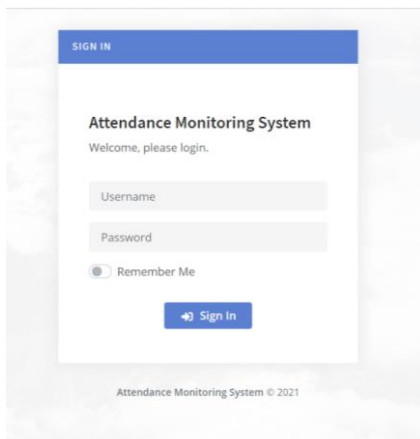


Fig 8. Login Portal

- Click on the subject which is to be conducted.
- Click on add new lecture.
- Enter the date, start time and end time of the lecture.
- Finally click on add lecture and the lecture has been created.

## 4.4 Results and Discussion

We will be creating a lecture of WN on May 4, 2021 from 6:30 p.m. to 7:30 p.m. and will see the results whether the camera is able to recognise faces and mark the attendance properly.

### 4.4.1 Creating Lecture

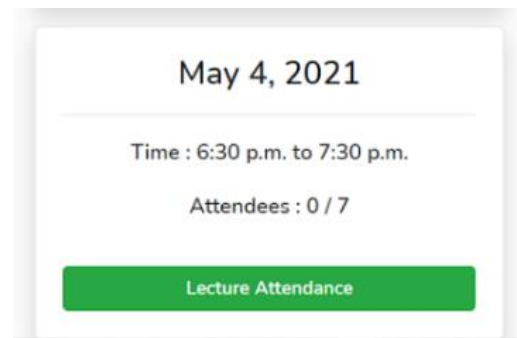


Fig 11. Created Lecture

### 4.3.2 Student Login

When a student enters his/her credentials, he/she is being redirected to the student dashboard. The student dashboard is as follows:

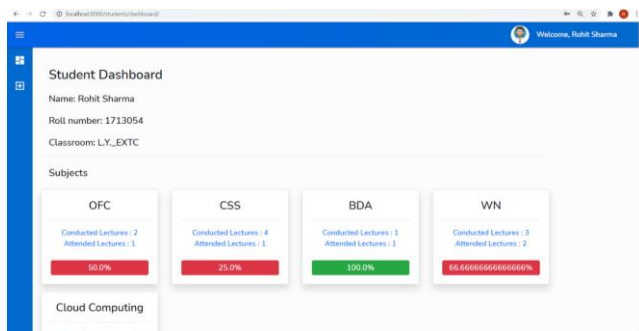


Fig 9. Student Dashboard

### 4.4.2 Detection of Faces

### 4.3.3 Faculty Login

When a faculty enters his/her credentials, he/she is being redirected to the faculty dashboard. The faculty dashboard is as follows:

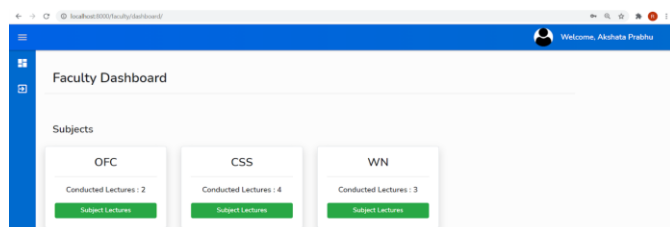


Fig 10. Faculty Dashboard

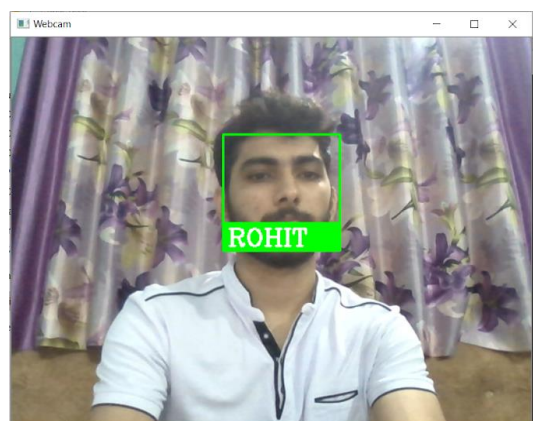


Fig 12. Face Recognized

The faculty dashboard is created in such a way that it should be easy for a faculty to access anyone's attendance with ease. The faculty can create the lecture very easily by following some steps:

### 4.4.3 Marking of Attendance



Fig 13. Attendance Marked Prompt

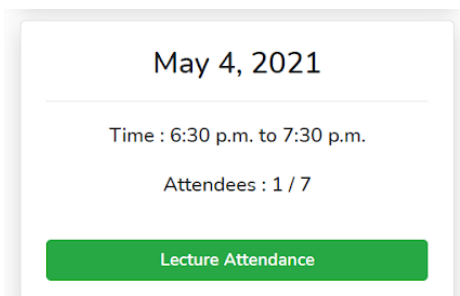


Fig 14. Attendance Marked on Faculty Portal

## 5. CONCLUSION

The Smart Attendance Management System was created to address the shortcomings of manual systems. We used the face recognition concept to monitor student attendance and improve the system. The method works well in a variety of poses and combinations. As a result, the aim of this project is to capture video of students, turn it into frames, link it to a database to ensure their participation or absence, and record attendance for each student. The Automated Attendance System aids in the improvement of accuracy and speed, resulting in high-precision real-time attendance to meet the demand for automatic classroom evaluation.

### Future Scope:

- The Automated Attendance System can be used in larger spaces, such as a seminar hall, to detect the presence of a large number of people.
- Image quality can be affected by bad lighting in the classroom, which can degrade device output indirectly. This can be solved in the later stages by enhancing the video quality or using certain algorithms.

- Make the code more flexible so that it can avoid discrepancies such as updating wrong names, or information in front of the student's name.

## ACKNOWLEDGEMENT

The successful implementation and final results of this project required thorough guidance and we are extremely privileged to have received that all along the duration of project completion. All that we have accomplished at the end of this tenure was only possible because of such supervision and assistance and we cannot forget to thank the ones who have directly or indirectly helped us through this journey.

We respect and thank our project guide Prof. Akshata Prabhu, who provided us with an opportunity to do the project work and gave us all the support and guidance which made us complete the project in due time and satisfactory manner.

## REFERENCES

- [1] Nandhini R, Duraimurugan N, S.P.Chokkalingam, "Face Recognition Based Attendance System", International Journal of Engineering and Advanced Technology (IJEAT), Volume-8, Issue-3S, February 2019.
- [2] Kaneez Laila Bhatti, Laraib Mughal, Faheem Yar Khuhawar, Sheeraz Ahmed Memon, "Smart Attendance Management System Using Face Recognition", EAI Endorsed Transactions on Creative Technologies, 2018, Volume 5 Issue 17, doi: 10.4108/eai.13-7-2018.159713.
- [3] <https://medium.com/@ageitgey/machine-learning-is-fun-part-4-modern-face-recognition-with-deep-learning-c3cffc121d78>
- [4] <https://developer.mozilla.org/en-US/docs/Learn/Server-side/Django/Introduction>
- [5] <https://components101.com/microcontrollers/raspberry-pi-3-pinout-features-datasheet>
- [6] <https://components101.com/misc/pi-camera-module>
- [7] [https://en.wikipedia.org/wiki/Microsoft\\_Visual\\_Studio](https://en.wikipedia.org/wiki/Microsoft_Visual_Studio)
- [8] <https://www.w3schools.com/html/>
- [9] <https://www.w3schools.com/css/>