



**DEPARTMENT OF,  
SHIVAJI UNIVERSITY, KOLHAPUR**

Academic Year 2021-22

A

Mini Project Report

On

**“Face Recognition Attendance  
System using Python”**

A Project Report Submitted

To

**Shivaji University, Kolhapur.**

**Submitted by:**

**Mr. Mahamadabrar Dastagir Maneri (41)**

**Ms. Aishwarya Sandip Patil (57)**

**Ms. Amisha Ashok Patil (58)**

# CERTIFICATE

This is to certify that as a partial fulfilment of the curriculum B. TECH, Department Of Technology Shivaji University, Kolhapur, the student, **Mr. Mahamadabrar Dastagir Maneri (41), Ms. Aishwarya Sandip Patil (57), Ms. Amisha Ashok Patil (58)** successfully completed the project entitled “**Face Recognition Attendance System using Python**” during the academic year 2021-2022.

**Place:** Kolhapur

**Date:**

**INCHARGE**

**CO - ORDINATOR**

**DIRECTOR**

## ACKNOWLEDGEMENT

I would like to express my special thanks of gratitude to my teacher Mr. Hemant A. Tirmare who gave me the golden opportunity to do this wonderful project on the topic “**Face Recognition Attendance System using Python**” and also guide me to complete this project, which also helped me in doing a lot of research and I came to know about so many new things about Face Recognition. My interest in this subject increased while researching about the project. I am really thankful to our **Mini Project** teacher **Mr. Hemant A. Tirmare**.

# INDEX

**Class:** T.Y CST

**Subject:** Mini Project

Sr. No.	Content	Page No.
1	Abstract	4
2	Introduction	5
3	Literature Review	6
4	Proposed System	7
5	Problem Statement	8
6	Methodology	9
7	Software Development	10
8	Overview	12
9	Project Code	13
10	Results	15
11	Conclusion	17
12	References	18

## Abstract

The conventional attendance system consists of registers marked by teachers which leads to human error and a lot of maintenance. Time consumption is an important point of concern in this system. We have thought of revolutionize it using available digital tools in the modern era i.e., FACE RECOGNITION. Our project will ensure more precision and negligible manual work. The project is revolutionized in order to overcome the problems of conventional system. Face recognition and then marking the attendance is our project all about. The database of all the students in the class is stored in a folder and when the face of the individual student matches with one of the faces stored image, attendance is marked else the face is ignored and attendance not marked. In our project, face recognition (Machine Learning) technology is used. Inside this Histogram of Oriented Gradient for face detection and SVM Classifier for name recognition is used. The model has an accuracy of 99.38% on the Labelled Faces in the Wild benchmark.

Keywords- Face Detection, Face Recognition, OpenCV, Tkinter etc.

## Introduction

In this Project, you will learn how to build a face-recognition system using Python. Face recognition is a step further to face detection. In face detection, we only detect the location of the human face in an image but in face recognition, we make a system that can identify humans.

Attendance plays an important role in any organization whether it be educational institutions or companies. So, it is very important to keep record of the attendance. The problem arises when one has to manually take the attendance which is not only time consuming but exhausting as well.

So, an automatic attendance system can solve such problem.

Basically, there are two kinds of system:

- 1) Manual Attendance System (MAS)
- 2) Automated Attendance System (AAS)

One of AAS system is biometric technique using finger prints, though it is automatic and a step ahead of traditional method it fails to meet the time and hygiene constraint. But using the biometric features of face solves such problem. Our projects emphasize on the features of the face like ears, nose etc. We used a method invented in 2005 called Histogram of Oriented Gradients (HOG) for face detection. For identifying the name of person simple linear SVM Classifier is used. All we need to do is train a classifier that can take in the measurements from a new test image and tells which known person is the closest match. The result is the name of person which is used to mark attendance.

## Literature Review

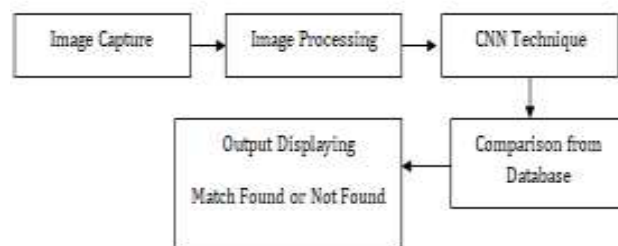
Approach for Face Detection and Attendance Using OpenCV and Machine learning The Face detection has been implemented Using a Method Called Histogram of Oriented Gradients In this system students' images are stored in database folder With Students name. when Any person comes in front of camera it captures the image of person and compares the captured image with images present in database Folder if images match with any of the image in database folder, then the attendance of the student will be marked and stored in CSV file.

Marking attendance using face recognition Automated Attendance system using Face recognition proposes that the system is based on face detection and recognition algorithm which is used to detect the student face when he/she come in front of camera and then compare the face with the images present in the folder if the match is found it will mark the attendance. This system has advantage over the traditional system as it saves time and there is no chance of proxy (that is no other student will mark the attendance of his/her friends).

## Proposed System

**Proposed System** The task of the proposed system is to capture the face of each student and to store it in the database for their attendance. The face of the student needs to be captured in such a manner that all the feature of the students' face needs to be detected. There is no need for the teacher to manually take attendance in the class because the system records a video and through further processing steps the face is being recognized and the attendance database is updated. This system is developed using python OpenCV.

**OpenCV:** OpenCV (Open-Source Computer Vision Library) is a library of programming functions mainly aimed at real-time computer vision.



**Figure 2.1: Flowchart of the Proposed System**

**Image Capture:** We need some HD camera in order to get results. We can capture the images from the video stream or by capturing each and every image from the webcam manually. Doing the frame capture from the stream of video will give us results in less time but we won't be able to capture the face properly in case we lose light or something and if the face is not captured properly.

**Image Processing:** Digital image processing is the use of a digital computer to process digital images through an algorithm. As a subcategory or field of digital signal processing, digital image processing has many advantages over analog image processing.



## Problem Statement

Attendance is an important part of daily classroom evaluation. At the beginning and ending of class, it is usually checked by the teacher, but it may appear that a teacher may miss someone or some students answer multiple times. Face recognition-based attendance system is a problem of recognizing face for taking attendance by using face recognition technology based on high-definition monitor video and other information technology.

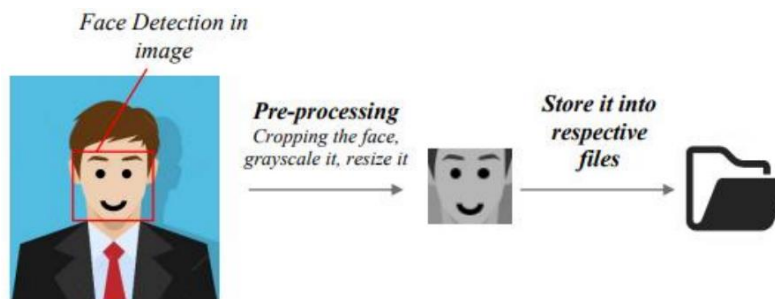
The concept of face recognition is to give a computer system the ability of finding and recognizing human faces fast and precisely in images or videos. Numerous algorithms and techniques have been developed for improving the performance of face recognition. Recently Deep learning has been highly explored for computer vision applications. Human brain can automatically and instantly detect and recognize multiple faces. But when it comes to computer, it is very difficult to do all the challenging tasks on the level of human brain. The face recognition is an integral part of biometrics. In biometrics, basic traits of human are matched to the existing data. Facial features are extracted and implemented through algorithms, which are efficient and some modifications are done to improve the existing algorithm models. Computers that detect and recognize faces could be applied to a wide variety of practical applications including criminal identification, security systems, identity verification etc.

The face recognition system generally involves two stages:

- Face Detection – where the input image is searched to find any face, then image processing cleans up the facial image for easier recognition.
- Face Recognition – where the detected and processed face is compared to the database of known faces to decide who that person is.

## Methodology

Before the attendance management system can work, there are a set of data needed to be inputted into the system which essentially consist of the individual's basic information which is their ID and their faces. The first procedure of portrait acquisition can be done by using the Camera to capture the faces of the individual. In this process the system will first detect the presence of a face in the captured image, if there are no face detected, the system will prompt the user to capture their face again until it meets certain number of portraits which will be 10 required portraits in this project for each student. The decision of storing only 10 portrait per student is due to the consideration of the limited storage space in the raspberry pi because the total amount of students in the university is considered heavy. Then, the images will undergo several pre-processing procedures to obtain a grayscale image and cropped faces of equal sized images because those are the prerequisites of using the Eigenfaces Recognizer. Both of the processes mentioned above can be represented in the diagram below.



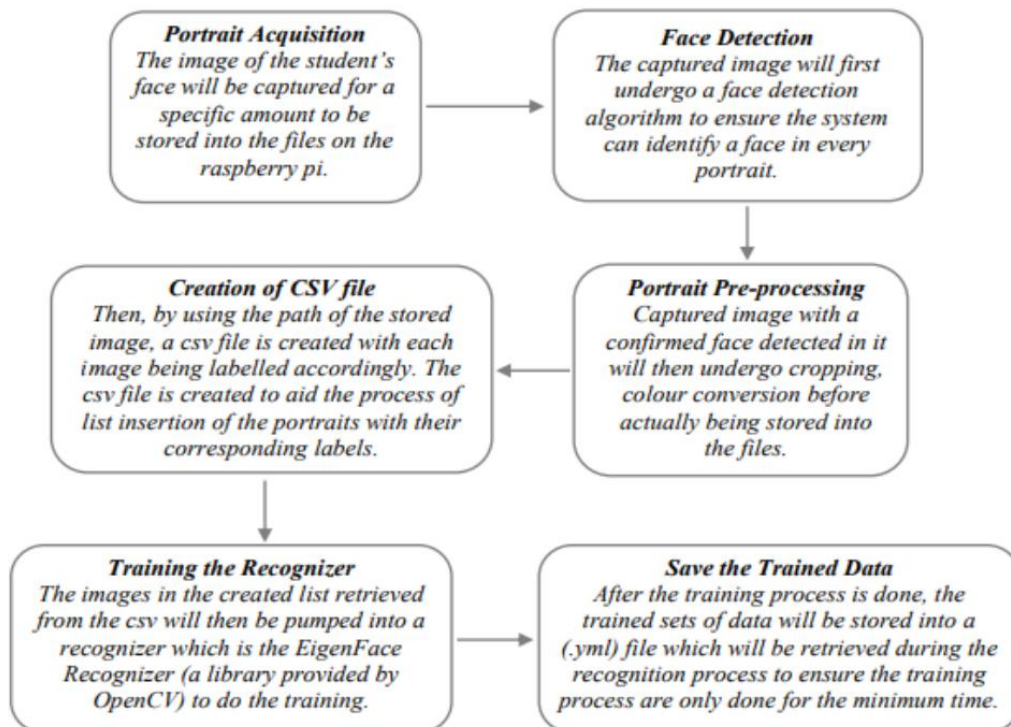
## Software Development

There are two major system flows in the software development section as shown below:

- The creation of the face database
- The process of attendance taking

Both processes mentioned above are essential because they made up the backbone of the attendance management system. In this section, the process of both flows will be briefly described. Meanwhile, their full functionality, specific requirements and also the methods/approach to accomplish such objectives will be discussed in the upcoming chapter.

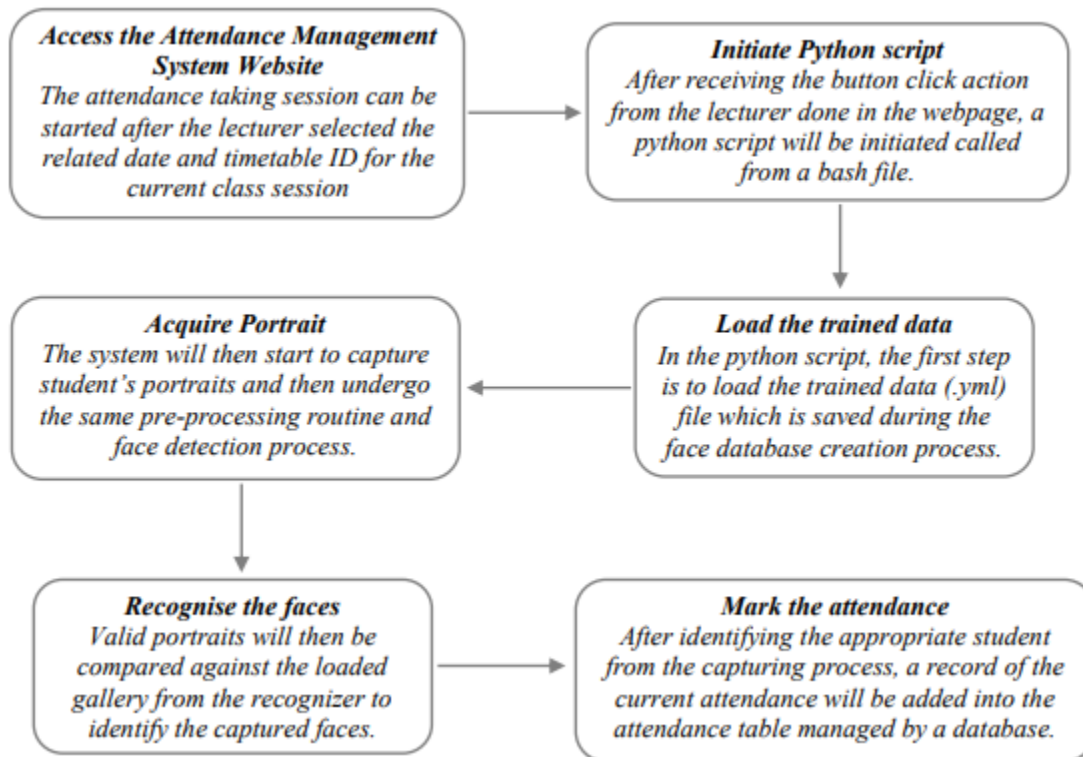
### The creation of the face database:



The face database is an important step to be done before any further process can be initiated. This is because the face database acts as a comparison factor during the recognition process which will be discussed in later section. In the process above, a csv file is created to aid the process of image labelling because there will be more than one

portrait stored for each student, thus, in order to group their portraits under the name of the same person, labels are used to distinguish them. After that, those images will be inserted into a recognizer to do its training. Since the training process is very time consuming as the face database grew larger, the training is only done right after there is a batch of new addition of student's portraits to ensure the training is done as minimum as possible

### The process of attendance taking:



## Overview

The proposed system is a software system which will mark attendance using facial recognition. In this project we used OpenCV module integrated with Python which will help the institution to make the attendance process easy and efficient. The system comprises of Computer, HD Video Camera and Wi-Fi module or Internet.

### Steps of Working:

- Initiate the firstpage.py python script.
- Create a DATASET of the student by entering his ID Number.
- Train the dataset, a yml file is created.
- A picture of the class is taken, and the RECOGNIZER python file is initiated.
- Attendance is taken by cropping the faces in the picture and comparing with the faces in the database.
- If a face is matched, the responding name with PRESENT status is marked in a EXCEL file with the current date and time.

## Project Code

### Main.py

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

# from PIL import ImageGrab

path = 'Training_images'
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('%H:%M:%S')
                f.writelines(f'\n{name},{dtString}')
```

```

#### FOR CAPTURING SCREEN RATHER THAN WEBCAM
# def captureScreen(bbox=(300,300,690+300,530+300)):
#     capScr = np.array(ImageGrab.grab(bbox))
#     capScr = cv2.cvtColor(capScr, cv2.COLOR_RGB2BGR)
#     return capScr

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

cap = cv2.VideoCapture(0)

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

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

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

        if matches[matchIndex]:
            name = classNames[matchIndex].upper()
            # print(name)
            y1, x2, y2, x1 = faceLoc
            y1, x2, y2, x1 = y1 * 4, x2 * 4, y2 * 4, x1 * 4
            cv2.rectangle(img, (x1, y1), (x2, y2), (0, 255, 0), 2)
            cv2.rectangle(img, (x1, y2 - 35), (x2, y2), (0, 255, 0), cv2.FILLED)
            cv2.putText(img, name, (x1 + 6, y2 - 6), cv2.FONT_HERSHEY_COMPLEX, 1, (255, 255,
255), 2)
            markAttendance(name)

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

```

## Results

First in all we need to register the person into the database. To do so, we need to give name and his/her registered number to store. Now we to get the pictures of the persons from the webcam or any other cams available like we used Logitech webcam here. Select the cam from which we need to take the image and start the camera. The camera is plotted in the axes and we can capture and save the images in the folder created automatically with the registered number we have entered. After this the data is stored in the database. Now let us capture a picture from the webcam and see the results. The camera starts and takes the image to give the results checking from the saved database.



## Conclusion

Capturing the images from camera or cc camera and applying techniques face detection and recognition can decrease the manual work from human and increase the security safety, taking the decision from this recognition result. Based on this face detection and recognition can used in implement so many application-like automatic attendances system based on face recognition, worker attendances, security, safety, police application like finding thief in image that help to catching thief. In this system we have implemented an attendance system for a lecture, section or laboratory by which lecturer or teaching assistant a record student's attendance. It saves time and effort, especially if it is a lecture with huge number of students. This attendance system shows the use of facial recognition techniques for the purpose of student attendance and for the further process this record of student can be used in exam related issues.



## References

Attendance System Based on Dynamic Face Recognition:

<https://ieeexplore.ieee.org/document/9258833>

Integration of Flask and Python on The Face Recognition Based Attendance System:

<https://ieeexplore.ieee.org/document/9590122>

Automated Student Attendance System Using Face Recognition:

<https://ieeexplore.ieee.org/document/9255052>

Face Recognition Based Smart Attendance System:

<https://ieeexplore.ieee.org/document/9160184>

OpenCV Documentation - <https://opencv.org>

NumPy - <https://numpy.org>