SMART ATTENDENCE SYSTEM USING FACE RECOGNITION

A MAJOR PROJECT REPORT

Submitted to

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY,HYDERABAD

In partial fulfilment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY IN COMPUTER SCIENCE AND ENGINEERING

Submitted by

BOMMAGANI RAJKUMAR	**19UK1A05F1**
VYDHYULA ACHALA	**19UK1A05F3**
ANDRU BHAVYA	**19UK1A05J7**
RAJIDI CHANDRA PRAKASH REDDY	**19UK1A05G9**

Under the esteemed guidance of

Ms.T.SRAVANTHI

(Assistant professor)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

VAAGDEVI ENGINEERING COLLEGE

(Affiliated to JNTUH, Hyderabad)

Bollikunta, Warangal – 506005

2019– 2023

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
VAAGDEVI ENGINEERING COLLEGE
BOLLIKUNTA, WARANGAL – 506005
2019 – 2023

CERTIFICATE OF COMPLETION

MAJOR PROJECT REPORT

This is to certify that the Major project entitled **"SMART ATTENDENCE SYSTEM USING FACE RECOGNITION"** is being submitted by **B.RAJKUMAR(H.T.NO:19UK1A05F1),VYDHYULA ACHALA(H.T.NO:19UK1A05F3),ANDRU BHAVYA(H.T.NO:19UK1A05J7),RAJIDI CHANDRA PRAKASH REDDY(H.T.NO:19UK1A05G9)** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering** to **Jawaharlal Nehru Technological University Hyderabad** during the academic year **2022-2023**, is a record of work carried out by them under the guidance and supervision.

Project Guide
Ms.T.Sravanthi
(Assistant Professor)

Head of the Department
Dr. R. Naveen Kumar
(Professor)

ACKNOWLEDGEMENT

We wish to take this opportunity to express our sincere gratitude and deep sense of respect to our beloved **Dr.P.PRASAD RAO,** Principal, Vaagdevi Engineering College for making us available all the required assistance and for his support and inspiration to carry out this Major project in the institute.

We extend our heartfelt thanks to **Dr.R.NAVEEN KUMAR**, Head of the Department of CSE, Vaagdevi Engineering College for providing us necessary infrastructure and thereby giving us freedom to carry out the Major project.

We express heartfelt thanks to Smart Bridge Educational Services Private Limited, for their constant supervision as well as for providing necessary information regarding the Major project and for their support in completing the Major project.

We express heartfelt thanks to the guide, **Ms.T.Sravanthi** (Assistant professor), Department of CSE for his constant support and giving necessary guidance for completion of this Major project.

Finally, we express our sincere thanks and gratitude to my family members, friends for their encouragement and outpouring their knowledge and experience throughout the thesis.

B.RAJKUMAR (19UK1A05F1)
V.ACHALA (19UK1A05F3)
A.BHAVYA (19UK1A05J7)
R.CHADRA PRAKASH (19UK1A05G9)

ABSTRACT

Face is the crucial part of the human body that uniquely identifies a person. Using the face characteristics as biometric, the face recognition system can be implemented. The most demanding task in any organization is attendance marking. In traditional attendance system, the students are called out by the teachers and their presence or absence is marked accordingly. However, these traditional techniques are time consuming and tedious. In this project, the Open CV based face recognition approach has been proposed. This model integrates a camera that captures an input image, an algorithm for detecting face from an input image, encoding and identifying the face, marking the attendance in a spreadsheet and converting it into PDF file. The training database is created by training the system with the faces of the authorized students. The cropped images are then stored as a database with respective labels. The features are extracted using LBPH algorithm.

Facial recognition represents a system for automatic recognition or identification of person based on digital pictures of their face.There are multiple algorithms that the recognition software uses but it all comes down to comparison of input picture,that is the comparsion of the biometric face features with the selected face pictures in the database.Facial recognition is mostly used in systems that are connected with safety and usually in two ways:recognition and verification.When it comes to recognizing the face parameters are compared with other faces in the database.For the verification,the face parameters of the person that is using another way of verification in the given moment,are compared with the information of that person in the database.This is done for sake of making sure that the person who is requesting access is actually a person with allowed access.Even though facial recognition is less precise compared to fingerprint scanning or iris scanning,it is very well accepted because of its non physical and non invasive way of use.

TABLE CONTENTS

LIST OF FIGURES :

Smart Attendence – Face Recognition

1. INTRODUCTION

Nowadays Educational institutions are concerned about regularity of student attendance. Thisis mainly due to students' overall academic performance is affected by his or her attendance in the institute. Mainly there are two conventional methods of marking attendance which are calling out the roll call or by taking student sign on paper. Theyboth were more time consuming and difficult. Hence, there is a requirement of computer-based student attendance management system which will assist the faculty for maintaining attendance record automatically. In this project we have implemented the automated attendance system using PYTHON. We have projected our ideas to implement "Automated Attendance System Based on Facial Recognition", in which it imbibes large applications. The application includes face identification, which saves time and eliminates chances of proxy attendance because of the face authorization.

Hence, this system can be implemented in a field where attendance plays an important role. The system is designed using PYTHON platform. The proposed system uses Principal Component Analysis (PCA) algorithm which is based on eigenface approach. This algorithm compares the test image and training image and determines students who are present and absent. The attendance record is maintained in an excel sheet which is updated automatically in the system.

Problem Statement Attendances of every student are being maintained by every school, college and university. Empirical evidences have shown that there is a significant correlation between students' attendances and their academic performances. There was also a claim stated that the students who have poor attendance records will generally link to poor retention. Therefore, faculty has to maintain proper record for the attendance. The manual attendance record system is not efficient and requires more time to arrange record and to calculate the average attendance of each student. Hence there is a requirement of a system that will solve the problem of student record arrangement and student average attendance calculation. One alternative to make student attendance system automatic is provided by facial recognition.

Face recognition can be applied for a wide variety of problems like image and film processing, human-computer interaction, criminal identification etc. This has motivated researchers to develop computational models to identify the faces, which are relatively simple and easy to implement. The existing system represents some face space with higher dimensionality and it is not effective too. The important fact which is considered is that although these face images have high dimensionality, in reality they span very low dimensional space. So instead of considering whole face space with high dimensionality, it is better to consider only a subspace with lower dimensionality to represent this face space. The goal is to implement the system (model) for a particular face and distinguish it from a large

number of stored faces with some real-time variations as well. The Eigenface approach uses Principal Component Analysis (PCA) algorithm for the recognition of the images. It gives us efficient way to find the lower dimensional space.

“Eigenfaces for recognition’’ (Mathew Turk and Alex Pentland) [1], here they have developed a near-real time computer system that can locate and track a subject’s head, and then recognize the person by comparing characteristics of the face to those of known individuals. The computational approach taken in this system is motivated by both physiology and information theory, as well as by the practical requirements of near-real time performance and accuracy. This approach treats the face recognition problem as an intrinsically two-dimensional recognition problem rather than requiring recovery of three- dimensional geometry, taking advantage of the fact that these faces are normally upright and thus may be described by a small set of two-dimensional characteristic views. Their experiments show that the eigenface technique can be made to perform at very high accuracy, although with a substantial “unknown “rejection rate and thus potentially well suited to these applications. The future scope of this project was-in addition to recognizing face, to use eigenface analysisto determine the gender of the subject and to interpret facial expressions. “Fast face recognition using eigenfaces” (Arun Vyas and Rajbala Tokas) [2], their approach signifies face recognition as a two-dimensional problem. In this approach, face reorganization is done by Principal Component Analysis (PCA). Face images are faced onto a space that encodes best difference among known face images.

The face space is created by eigenface methods which are eigenvectors of the set of faces, which may not link to general facial features such as eyes, nose, and lips. The eigenface method uses the PCA for recognition of the images. The system performs by facing pre-extracted face image onto a set of face space that shows significant difference among known face images. Face will be categorized as known or unknown face after imitating it with the present database. From the obtained results, it was concluded that, for recognition, it is sufficient to take about 10% eigenfaces with the highest eigenvalues. It is also clear thatthe recognition rate increases with the number of training images.

2. SYSTEM ANALYSIS

2.1 PROPOSED SYSTEM

The present system of attendance marking i.e., manually calling out the roll call by the faculty have quite satisfactorily served the purpose. With the change in the educational system with the introduction of new technologies in classroom such as virtual classroom, the traditional way of taking attendance may not be viable anymore. Even with rising number of course of study offered by universities, processing of attendance manually could be time consuming. Hence, in our project we aim at creating a system to take attendance using facial recognition technology in classrooms and creating an efficient database to record them.

Block diagram

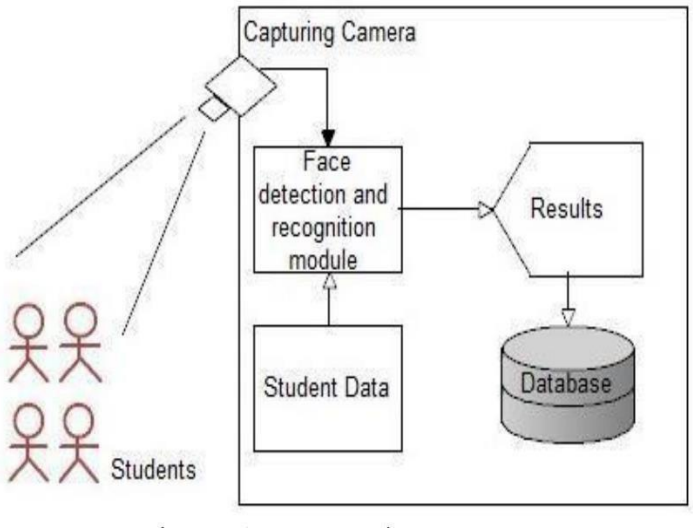


Figure 1 Proposed system

The block diagram in figure 1 describes the proposed system for Face Recognition based Classroom attendance system. The system requires a camera installed in the classroom at a position where it could capture all the students in the classroom and thus capture their images effectively

3. REQUIREMENT & SPECIFICATION

3.1 SCOPE OF THE SYSTEM

We are setting up to design a system comprising of two modules. The first module (face detector) is a mobile component, which is basically, a camera application that captures student faces and stores them in a file using computer vision face detection algorithms and face extraction techniques. The second module is a desktop application that does face recognition of the captured images (faces) in the file, marks the students register and then stores the results in a database for future analysis.

3.2 OBJECTIVE OF THE SYSTEM

The main objective of this project is to offer system that simplify and automate the process of recording and tracking students' attendance through face recognition technology. It is biometric technology to identify or verify a person from a digital image or surveillance video.

3.3 SYSTEM REQUIREMENTS

3.3.1 Software Requirements

❖ Operating system - Windows 7 Ultimate.

❖ Coding Language - Python.

❖ Front-End - Python.

3.3.2 Hardware Requirements

➢ Processor - Pentium –IV

➢ RAM - 8 GB (min)

➢ Hard Disk - 200 GB

➢ Key Board - Standard Windows Keyboard

➢ Mouse - Two or Three Button Mouse

➢ Monitor – SVGA

3.4 SYSTEM DESIGN

3.4.1 Input Design

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

➢ What data should be given as input?

➢ How the data should be arranged or coded?

➢ The dialog to guide the operating personnel in providing input.

➢ Methods for preparing input validations and steps to follow when error occur.

OBJECTIVES

Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.

It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.

When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow

3.4.2 Output Design

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decisionmaking. 1. Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.

4. DESIGN REQUIREMENT

We used some tools to build the HFR 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.

4.1 SOFTWARE IMPLEMENTATION

OpenCV: We used OpenCV 3 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 BSD 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, highdefinition imaging (HDR), image inpainting.

Image Processing Toolbox

The Image Processing Toolbox is a collection of functions that extend the capability of the PYTHON® numeric computing environment. The toolbox supports a wide range of image processing operations, including: • Spatial image transformations

• Morphological operations

• Neighborhood and block operations

• Linear filtering and filter design

• Transforms • Image analysis and enhancement

• Image registration

• De blurring

• Region of interest operations

Many of the toolbox functions are PYTHON M-files, a series of PYTHON statements that implement specialized image processing algorithms. You can extend the capabilities of the Image Processing Toolbox by writing your own M-files, or by using the toolbox in combination with other toolboxes, such as the Signal Processing Toolbox and the Wavelet Toolbox. The toolbox also includes a Simulink interface called the Image Acquisition Block set. This block set extends Simulink with a block that lets you bring live video data into a model.

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

5. TEST OBJECTIVES

• All field entries must work properly.

• Pages must be activated from the identified link.

• The entry screen, messages and responses must not be delayed.

5.1 TEST APPORACHES

Field testing will be performed manually and functional tests will be written in detail.

TYPES OF TESTING

5.2 UNIT TESTING

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

5.3 INTEGRATION TESTING

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

5.4 FUNCTIONAL TEST

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals. Functional testing is centered on the following items: Valid Input-identified classes of valid input must be accepted. Invalid Inputidentified classes of invalid input must be rejected. Functions-identified functions must be exercised. Output-identified classes of application outputs must be exercised. Systems/Procedures: interfacing systems or procedures must be invoked. Organization and preparation of functional tests is focused on requirements, key functions,

6. SYSTEM IMPLEMENTATION

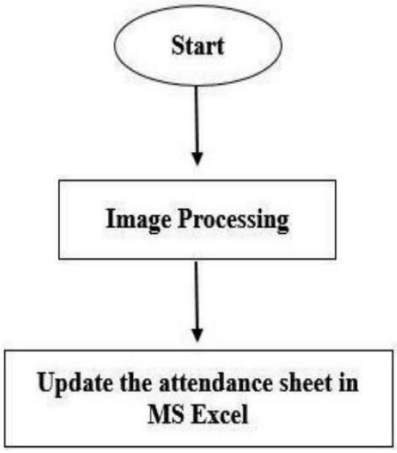


Figure 2 system flowchart

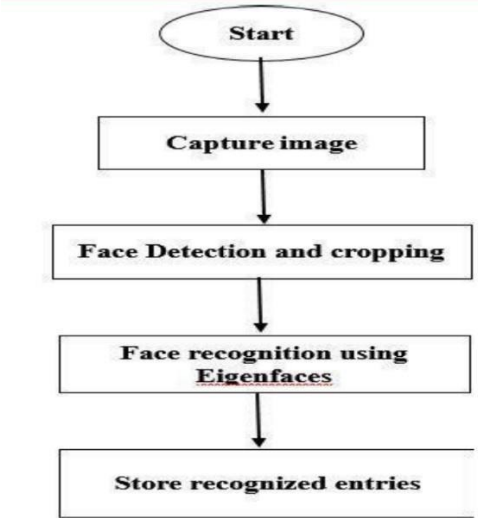


Figure 3 image processing procedure

7. RESULT AND ANALYSIS

Using the all the functions we have created; we have tested for output in using existing test images as well as in real-time. Following section, the screenshots of the output of different functions are given. We have tested the system with the help of four volunteers. Taking images from cam to create the test database

With the help of four volunteers, three images of each candidate is stored on the database as shown in the figure. For more accuracy we can increase the number of training images but with a compromise in the speed of calculation. However, for our application calculation speed variation won't be problem since a class timing is typically at least one hour and this period is just a lot more than the computation time takes by the algorithm.

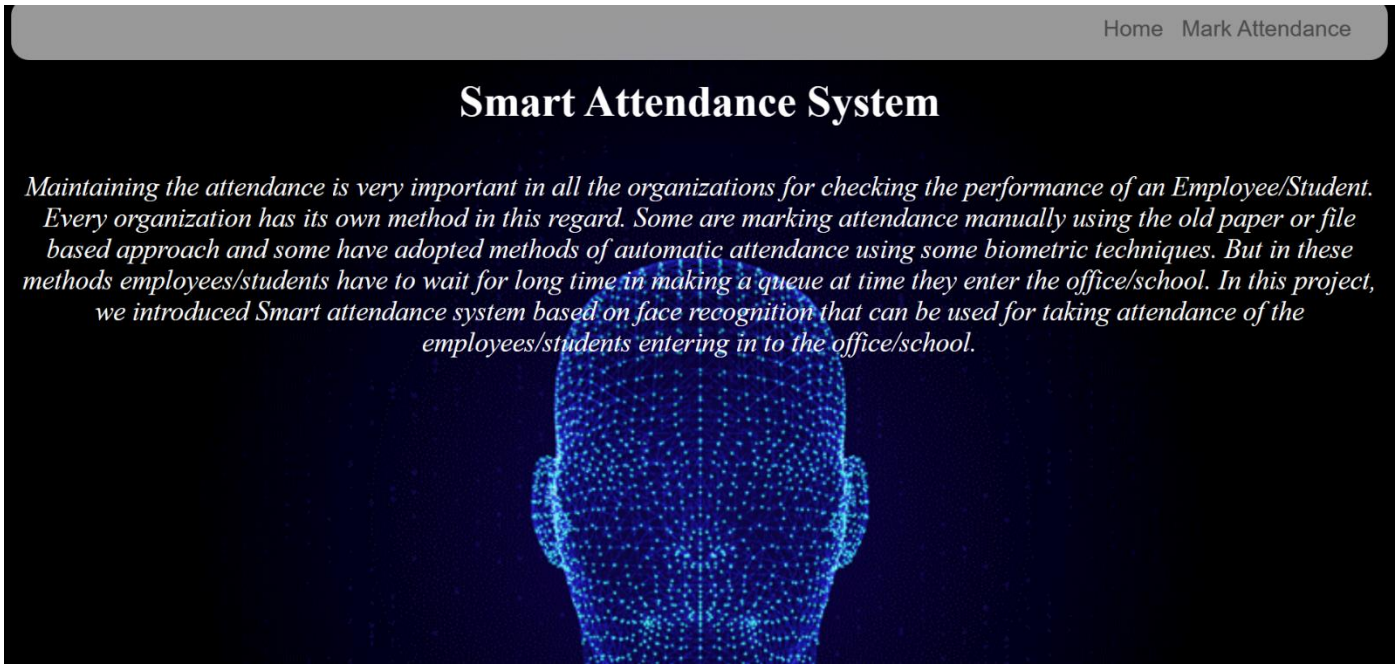
One thing we have keep in mind during this phase is to take the picture in ambient lighting and the frontal face must be clearly visible. Also there must be slight variation on the position or expression of the student in each captured image for better results. In normal lighting conditions and based on the proper sitting posture of the students the faces are efficiently captured. The classroom lighting has to be efficiently maintained. Also in case of blackouts appropriate alternatives have to be arranged.

All the detected faces which can be seen in figure, are cropped and saved in the Test Database folder. From this location the next algorithm read the image and further processing are carried out. The path of the folder must be exactly specified. Also the name each of the faces are given as numbers automatically. This helps in easier reading of the images from the folder.

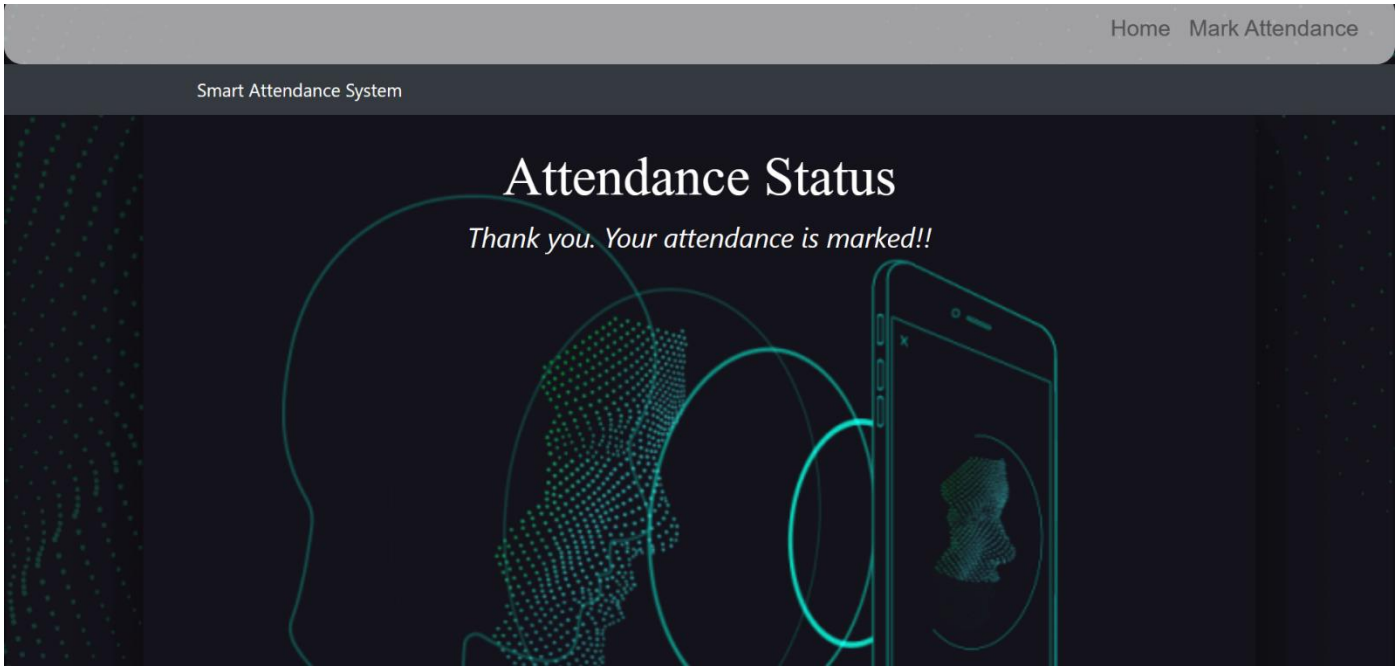
Face Recognition

Cropped facial images are fed into the face recognition algorithm and we get the results. The Eigen faces algorithm is applied to the image and compared with the database. We get the output as in figure after this process. If a person whose database is not present in the database, his image is simply ignored. However proper lighting to be maintained in order to prevent in any false detection.

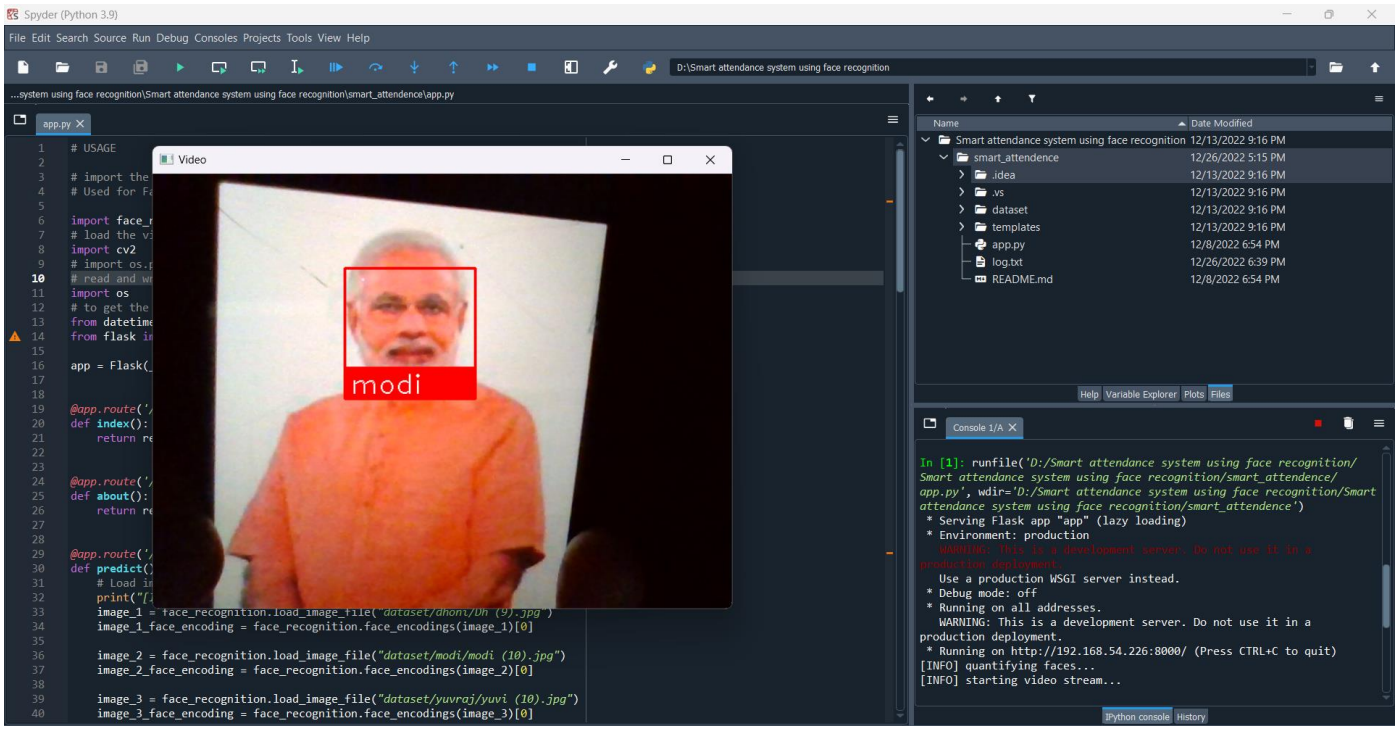
Home.html :



Upload.html :



Output:



ADVANTAGES & DISADVANTAGES

ADVANTAGES

Automated system

Time and Cost-saving

Better Security

Touchless feature

Easy to install

DISADVANTAGES

Breach of privacy

Biased performance

No so reliable

APPLICATION

In Since the usage of intelligent software is more and more present,facial recognition is also becoming a part of our day to day routine and there is no reason why it should not be used for the automatization of talking evidence of attendance. This represents our small contribution to modernizing the education system.

Example of facial recognition in real life:

PREVENT RETAIL CRIME – It lowers crime rates,by identifying know criminals when they enter retail establishments, Their photographs are kept in a database of criminals and management is notified when a notorious person enters .

CONCLUSION

In this system we have implemented an attendance system for a lecture, section or laboratory by which lecturer or teaching assistant can record students' attendance. It saves time and effort, especially if it is a lecture with huge number of students. Automated Attendance System has been envisioned for the purpose of reducing the drawbacks in the traditional (manual) system. This attendance system demonstrates the use of image processing techniquesin classroom. This system can not only merely help in the attendance system, but also improve the goodwill of an institution.

FUTURE SCOPE

There are currently no regulations in the United States expressly covering the biometric data of a person. Facial recognition devices are already being tested or implemented for airport protection, and it is reported that their faceprint has now been produced by more than half the United States populace. Information may be collected and processed by a facial recognition program, and a person does not even recognize it. Then, a hacker might reach the details, and the knowledge of a person would propagate without even realizing it. Government entities or marketers may use this data to monitor individuals too. Worse still, a false positive may include a person for a crime they are not.

Hundreds of companies have embraced face recognition. Integrating and installing is reasonably straightforward, but it has also provided users a feeling of utilizing a system that is more sophisticated and safer than passwords or PINs, thereby increasing user experience. Nonetheless, plenty is often unclear on the road to implementing what many deem the ideal biometric approach, causing several relatively severe blunders along the way.

BIBILOGRAPHY

1.www.google.com
2.www.wikipedia.org
3.https://ieeexplore.ieee.org

Appendix:

Source code

```
# USAGE

# import the necessary packages
# Used for Face Recognition

import face_recognition
# load the video and processthe video frame
import cv2
# import os.path
# read and write contents to CSV FIle
import os
# to get the time stamp
from datetime import datetime
from flask import Flask, request, render_template

app = Flask(__name__, template_folder="templates")


@app.route('/', methods=['GET'])
def index():
    return render_template('home.html')


@app.route('/home', methods=['GET'])
def about():
    return render_template('upload.html')
```

```
@app.route('/upload', methods=['GET', 'POST'])
def predict():
    # Load images.
    print("[INFO] quantifying faces...")
    image_1 = face_recognition.load_image_file("dataset/dhoni/Dh (9).jpg")
    image_1_face_encoding = face_recognition.face_encodings(image_1)[0]

    image_2 = face_recognition.load_image_file("dataset/modi/modi (10).jpg")
    image_2_face_encoding = face_recognition.face_encodings(image_2)[0]

    image_3 = face_recognition.load_image_file("dataset/yuvraj/yuvi (10).jpg")
    image_3_face_encoding = face_recognition.face_encodings(image_3)[0]

    image_4 = face_recognition.load_image_file("dataset/sharukh/sha (2).jpg")
    image_4_face_encoding = face_recognition.face_encodings(image_4)[0]

    # Create arrays of known face encodings and their names
    known_face_encodings = [

        image_1_face_encoding,
        image_2_face_encoding,
        image_3_face_encoding,
        image_4_face_encoding

    ]
    known_face_names = [

        "dhoni",
        "modi",
        "sharukh",
        "yuvraj"
    ]
```

```
    # Initializing  variables
    face_locations = []
    face_encodings = []
    face_names = []
    process_this_frame = True

    # Get a reference to webcam #0 (the default one)

    print("[INFO] starting video stream...")
    video_capture = cv2.VideoCapture(0)

    # loop over frames from the video file stream
    while True:
        # grab the frame from the threaded video stream
        ret, frame = video_capture.read()

        # Resize frame of video to 1/4 size for faster face recognition processing
        small_frame = cv2.resize(frame, (0, 0), fx=0.25, fy=0.25)

        # Convert the image from BGR color (which OpenCV uses) to
        # RGB color (which face_recognition uses)
        rgb_small_frame = small_frame[:, :, ::-1]

        # Only process every other frame of video to save time
        if process_this_frame:
            # Find all the faces and face encodings in the current frame of video
            face_locations = face_recognition.face_locations(rgb_small_frame)
            face_encodings = face_recognition.face_encodings(rgb_small_frame, face_locations)
```

```
        face_names = []
        name = ""
        for face_encoding in face_encodings:
            # See if the face is a match for the known face(s)
            matches = face_recognition.compare_faces(known_face_encodings, face_encoding)
            name = "Unknown"

            # If a match was found in known_face_encodings, just use the first one.
            if True in matches:
                first_match_index = matches.index(True)
                name = known_face_names[first_match_index]
                # Enter Tracking log
                logFile = open('log.txt', mode="a")
                # set the file pointer to end of the file
                pos = logFile.seek(0, os.SEEK_END)
                # If this is a empty log file then write the column headings
                if pos == 0:
                    logFile.write("Year,Month,Day,Time,Name,Attendance")
                # Set Date and Time
                ts = datetime.now()
                newDate = ts.strftime("%m-%d-%y")
                year = ts.strftime("%Y")
                month = ts.strftime("%m")
                day = ts.strftime("%d")

                time1 = ts.strftime("%H:%M:%S")
                info = "{},{},{},{},{},{}\n".format(year, month, day, time1, name, "Present")
                logFile.write(info)
                logFile.close()
```