A CAL Project Report

on

Face Recognition Based Attendance System

to be submitted in partial fulfilling of the requirements for the course on

Artificial Intelligence– CSE 3013
(F2)

by

Aditya Kumar 17BCE0065
Rajat Sharma 18BCB0065

Fall Semester 2020-2021

TABLE OF CONTENTS

1. ABSTRACT

Face Detection and Recognition (FR) technologies have made many improvements in the changing and growing world. Smart Attendance using Real-Time Face Recognition is a real-world solution which comes with day to day activities of handling student attendance. Face recognition-based attendance system is a technique of recognizing the students face for taking attendance by using facial recognition based on web cam video and other data technology. In my face-recognition based attendance system project, a computer system will be able to find and recognize human faces fast and precisely in front of web cam that are being captured through my system web cam. Numerous ML algorithms and techniques have been developed and implemented for improving the performance of face recognition but the concept to be implemented here is Deep Learning. It helps in conversion of the frames of the video into images so that the face of the student can be easily recognized for their attendance so that the attendance database can be easily reflected automatically.

1. INTRODUCTION

In the current situation of COVID-19, where the biggest problem is social distancing, this Face-Recognition Based Attendance System is an excellent idea that will help the companies to go in their normal flow and with this system we can efficiently take attendance also in Schools and colleges.

Deep Learning is one among the interesting Artificial Intelligence technology that enables the machine to train itself by providing some datasets as input and provides an appropriate output during testing by applying different learning algorithms. Nowadays Attendance is considered as an important factor for both the student as well as the teacher of an educational organization. With the advancement of the deep learning technology the machine automatically detects the attendance performance of the students and maintains a record of those collected data.

The project of creating a web application, Face Recognition Based Attendance System, for the management of attendance system of the company during this COVID-19 pandemic. Earlier this system, managing of attendance used to be manual from start to end, and records of the requests were kept in unsafe place in an Excel file by the main person, can also be known as admin. This Face Recognition Based Attendance System

was designed to ease the work of admin, make the attendance process safer for the company employees and decrease the risk of managing records of requests. Automated attendance system reduced a lot of time for execution of this process. In this system, attendance can be taken in both the times, first when the user enters to work, and secondly when he exits the office, and with the status IN and OUT respectively.

In general, the attendance system of the student can be maintained in two different forms namely:

- **Manual Attendance System (MAS)**
- **Automated Attendance System (AAS).**

The two common Human Face Recognition techniques are:

- **Feature-based approach**
- **Brightness-based approach.**

The Feature-based approach also known as local face recognition system, used in pointing the key features of the face like eyes, ears, nose, mouth, edges, etc., whereas the brightness-based approach also termed as the global face recognition system, used in recognizing all the parts of the image.

In this project, I have used Feature-based approach.

2.1 Tools and software used-

Front-end of the website- Bootstrap, HTML, CSS.

Text-editor- Brackets, Visual Studio Code.

For Web Based Application- Flask.

Back-end/Database – MySQL.

For Face-Recognition- Computer Vision, dlib, face-recognition packages.

2.2 Dataset used-

1. Training Images-

In a folder named as Training Images, I will be storing all the images of different users to actually train my Face-Recognition model.

2. For Face detection process-

I will be using Haar feature-based cascade classifier.

A Haar Cascade is basically a classifier which is used to detect particular objects from the source. **The haarcascade_frontalface_default.xml** is a haar cascade designed by OpenCV to detect the frontal face.

2.3 Possible Input and Output of the Project

Input-

Input Information of User- ID, Name. If the user is verified then take input images.

Output-

Attendance will be updated if the face matches with Images Database successfully.

2.4 Ultimate Utility value of the project to the society and/or Industry-

The project of creating a web application, Face Recognition Based Attendance System, for the management of attendance system of the company during this COVID-19 pandemic. Earlier this system, managing of attendance used to be manual from start to end, and records of the requests were kept in unsafe place in an Excel file by the main person, can also be known as admin. This Face Recognition Based Attendance System was designed to ease the work of admin, make the attendance process safer for the company employees and decrease the risk of managing records of requests. Automated attendance system reduced a lot of time for execution of this process. In this system, attendance can be taken in both the times, first when the user enters to work, and secondly when he exits the office, and with the status IN and OUT respectively.

3. Technical Skill Set gained in the process of project.

Following list mentions the technical knowledge I gained:

1. Machine Learning Fundamentals.
2. Building interactive ML Web-based applications using Streamlit.
3. Data Pre-processing

4. MySQL with XAMPP server
5. Frontend-side of web development
6. Face detection and Face Recognition Techniques
7. Making the web application from http to https (that is from unsecured to secured)
8. Computer Vision
9. Putting up the web application over the server
10. Natural Language Processing
11. Flask Web Framework.

4. SURVEY AND ANALYSIS

4.1 Face Recognition Based Attendance Management System Using Raspberry Pi.

In the present-day participation framework is viewed as manual. It requires a significant measure of time commonly for educators and understudies. There is still likelihood for proxies in the class when participation is manual. The primary commitment of the work is to build up a system the face recognition system that utilizes the substance of understudies as the feed input. To make it accessible for each stage they have picked the Raspberry Pi and a webcam linked to it. A webcam is connected with the Raspberry Pi module. Face pictures of the understudies are recorded utilizing the web cam. Distinctive face extraction steps are actualized utilizing the OpenCV stage utilizing the Qt maker IDE.

4.2 Face Recognition with Voice Assistance for the Visually Challenged

Outwardly weakened individuals face a ton of difficulties in everyday life. Having seen the challenges looked by them, their essential goal is to encourage certainty and to enable them to lead a day to day existence liberated from dangers identified with their wellbeing and prosperity. The absence of capacity to distinguish known people without hear-able or physical connection signals radically restricts the outwardly tested in their social associations and represents a danger to their security. In the course of recent years numerous model models have been created to help this populace with the undertaking of face acknowledgment. This application will diminish the intrinsic trouble for acknowledgment of an individual. It will introduce a facial acknowledgment application with an instinctive UI that empowers the oblivious to perceive individuals and cooperate socially. The deliberately planned interface lets the outwardly tested to have the option to access and use it with no prerequisite for obvious prompts as the clients are familiar by a voice right hand to explore through the application. The

whole form is intended to run proficiently on a Raspberry Pi 3 model B module utilizing the Android Things stage. The Open CV library has been utilized for the location and acknowledgment of individuals in this undertaking. This empowers the extension for the product to be run on a large number of gadgets, for example, camera installed glasses to caution clients of their environmental factors and distinguish individuals to interface securely. Since everything in the application is done progressively with no necessity for earlier datasets to be hardcoded it definitely improves the adaptability of the product.

4.3 Smart Office Surveillance Robot using Face Recognition.

This paper presents a surveillance robot that performs face recognition using artificial intelligence (AI), controlled via an Android application for examining threats in its vicinity. The robot is capable of detecting humans in real time scenarios. The activity cycle of the robot is categorized in a number of tasks: Object Detection, Face Detection, Face Recognition, generation of alert signals to the user. The robot detects the human face based on Haar Cascade Algorithm using a camera module interfaced with Raspberry Pi. The system is exposed to numerous facial images of people from different angles and background conditions. This forms the database to train the system which aids in facial recognition. Subsequently, when the robot detects a human face, it initiates the algorithm for recognition of face which uses Local Binary Pattern (LBP) approach. According to images in the dataset, the robot determines whether the human detected is a threat or not. In case of a match with the dataset, the user receives a message through GSM module on his/her mobile phone along with the name of the person recognized. In case of a mismatch, a threat or an alert message is received by the user. The motion of the robot is controlled by the android application through Bluetooth.

4.4 Human Face Recognition Using Local Binary Pattern Algorithm - Real Time Validation.

A real time face recognition using LBP algorithm and image processing techniques are proposed. Face image is represented by utilizing information about shape and texture. In order to represent the face effectively, area of the face is split into minute sections, then histograms of Local Binary Pattern (LBP) are extorted which are then united into a single histogram. Secondly, the recognition is carried out on computed feature space using nearest neighbour classifier. The developed algorithm is validated in real time by developing a prototype model using Raspberry Pi single board computer and also in simulation mode using MATLAB

software. The above obtained results match with each other. On comparing both the results, recognition time taken by the prototype model is more than that of the simulation results because of hardware limitations. The real time experimental results demonstrated that the face recognition rate of LBP algorithm is 89%.

4.5 A Face Recognition Based Automatic Attendance Management System by Combining LBP and HOG Features.

In the growing technology era, educational institutes are particular about the regularity of the students. Because the academic performance and evaluation depends on the attendance of the student. However, the method of taking attendance of the students still remains the orthodox way i.e., calling the roll number or taking the signature of the students in a sheet of paper. The shortcomings of these methodologies are that they sluggish and are quite often influenced by duplicate data entries by the students. So, in this paper we present a novel methodology of taking student's attendance through face recognition technique. The facial features of the students are extracted via Local Binary Pattern (LBP) and Histogram of Oriented Gradients (HOG). Both LBP and HOG features are combined to create a new feature vector. A classification model is implemented using Support Vector Machine (SVM) classifier which predicts student based on comparison made between the features of the query image and the features of the images stored in the student database.

4.6 Analysis and Optimization of Parameters used in Training a Cascade Classifier.

Training a cascade classifier for object detection using Local Binary Pattern (LBP) and Histogram of Gradients (HOG) features is computationally exorbitant. If the parameters of training are not chosen appropriately, the training may take weeks to complete with the output of an inefficient classifier. The state-of-the-art face recognition applications demand accurate and reliable cascade classifiers. Open Computer Vision (OpenCV) organization provides libraries which accomplish the training task once all parameters are given as inputs. In this paper we analyse the parameters experimentally and concluded with an optimal range of values for each of these parameters. Testing of the generated classifiers with optimal parameters values is performed on a dataset of 4000 test images. The training of these classifiers with optimal parameters takes an average training time of 25000 sec and provides average true positive detection of 88%.

4.7 Fast training and selection of Haar features using statistics in boosting-based face detection.

Training a cascade-based face detector using boosting and Haar features is computationally expensive, often re- quiring weeks on single CPU machines. The bottleneck is at training and selecting Haar features for a single weak classifier, currently in minutes. Traditional techniques for training a weak classifier usually run in O(NT log N), with N examples (approximately 10,000), and T features (ap- proximately 40,000). We present a method to train a weak classifier in time O(Nd2 + T), where d is the number of pixels of the probed image sub-window (usually from 350 to 500), by using only the statistics of the weighted input data. Experimental results revealed a significantly reduced training time of a weak classifier to the order of seconds. In particular, this method suffers very minimal immerse in training time with very large increases in members of Haar features, enjoying a significant gain in accuracy, even with reduced training time.

4.8 Intelligent biometric detection system for disabled people.

This paper presents an intelligent system for disabled people to have a full control of computers using biometric detection so that they can still operate computers like normal people do. An idea behind this project is a fact that the emergence of difficulties for disabled people to interact with computers. For instance, paralyzed people are unable to control computer mouse. Hence this project was conducted. This project only uses a webcam as an input which captures the user's head and then tracks their head movement and iris in order to control the mouse actions. It is done by using image processing techniques such as Component Localization, Haar Cascade classifier and Hough transform.

4.9 A Design of Iris Recognition System at a Distance.

Iris recognition is a powerful biometric for personal identification, but it is difficult to acquire good-quality iris images in real time. For making iris recognition more convenient to use, we design an iris recognition system at a distance about 3 meters. There are many key issues to design such a system, including iris image acquisition, human-machine-interface and image processing. In this paper, we respectively introduce how we deal with these problems and accomplish the engineering design. Experiments show that our system is convenient to use at the distance of 3 meters and the recognition rate is not worse than the state-of-the-art close-range systems.

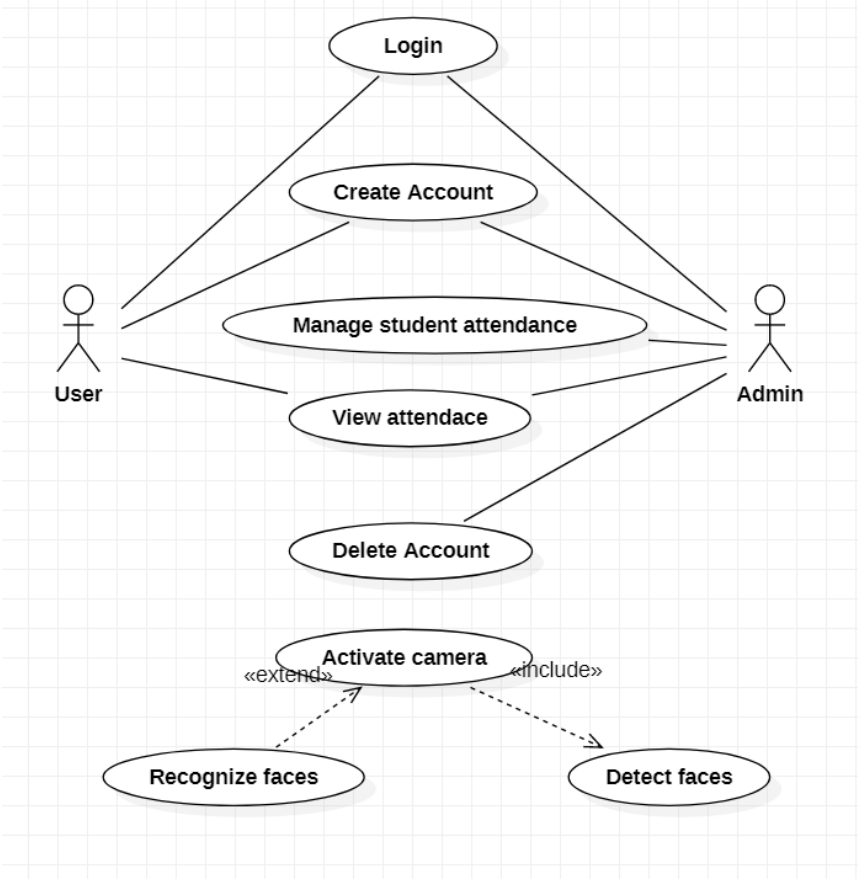
4.10 Face Recognition using Modified Local Directional Pattern Image.

Generally, binary pattern transforms have been used in the field of the face recognition and facial expression, since they are robust to illumination. Thus, this paper proposes an

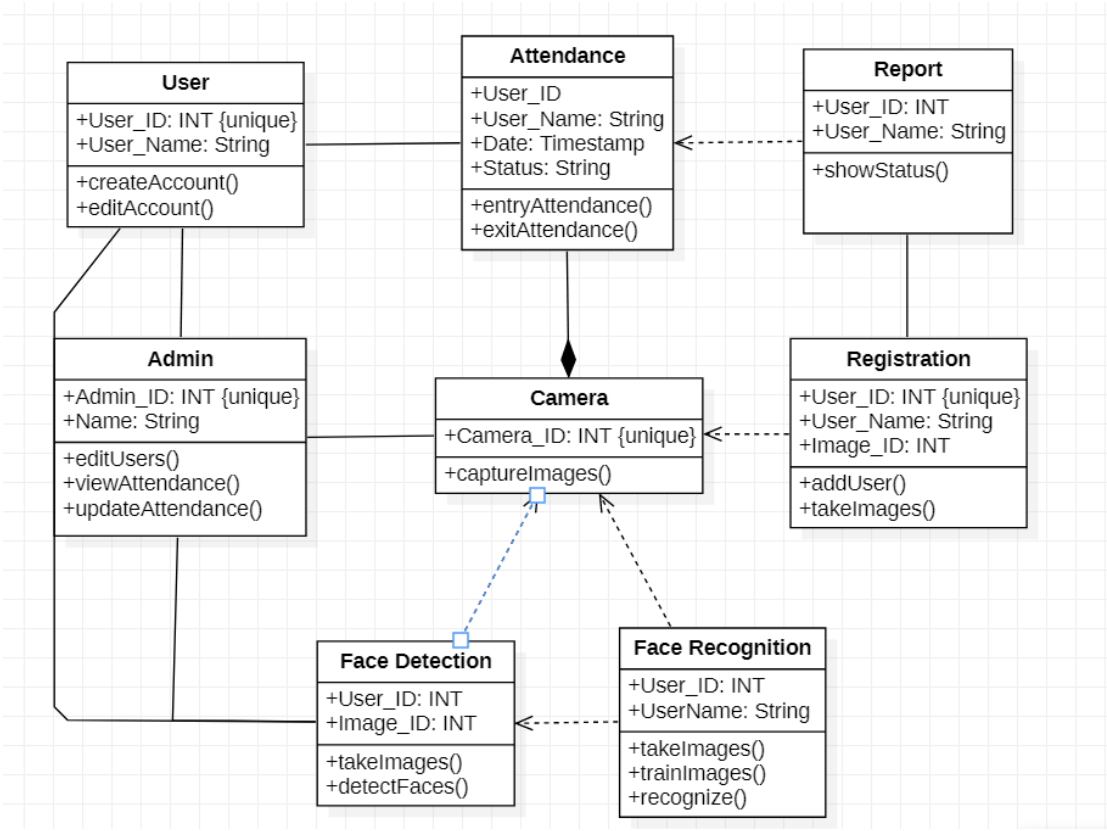
illumination-robust face recognition system combining an MLDP, which improves the texture component of the LDP, and a 2D-PCA algorithm. Unlike that binary pattern transforms such as LBP and LDP were used to extract histogram features, the proposed method directly uses the MLDP image for feature extraction by 2D-PCA. The performance evaluation of proposed method was carried out using various algorithms such as PCA, 2D-PCA and Gabor wavelets-based LBP on Yale B and CMU-PIE databases which were constructed under varying lighting condition. From the experimental results, we confirmed that the proposed method showed the best recognition accuracy.

5. DESIGN OF DIAGRAMS

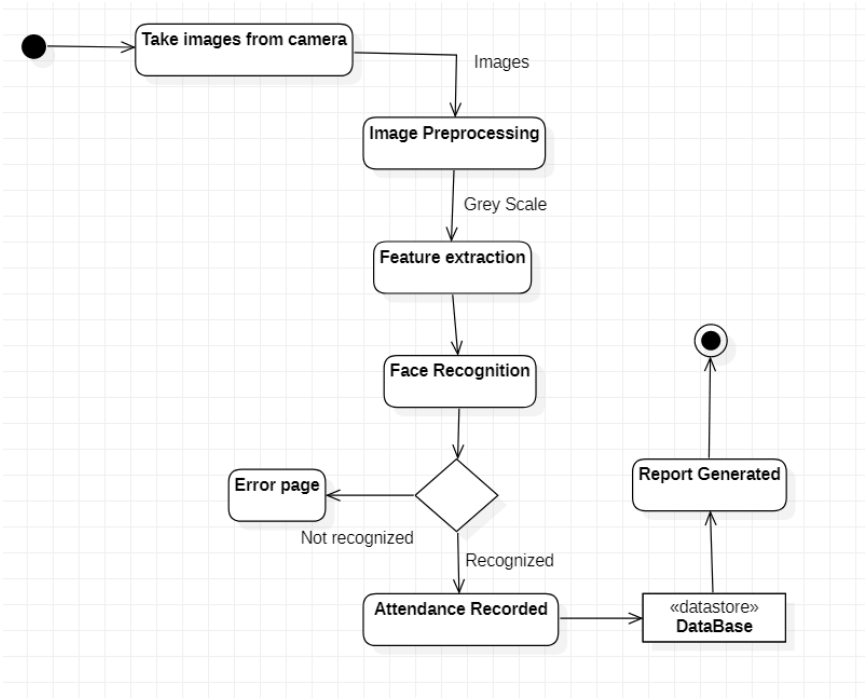
a. Use case Diagram-



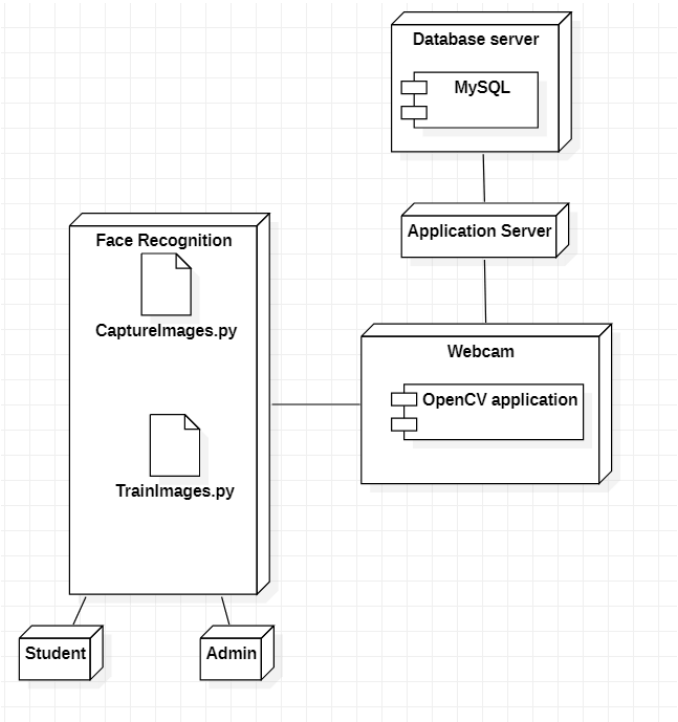
4.2 Class Diagram



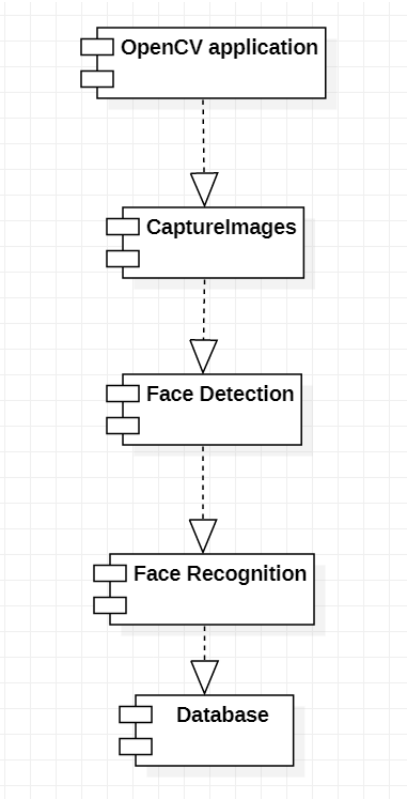
4.3Activity Diagram



4.4Deployment Diagram-

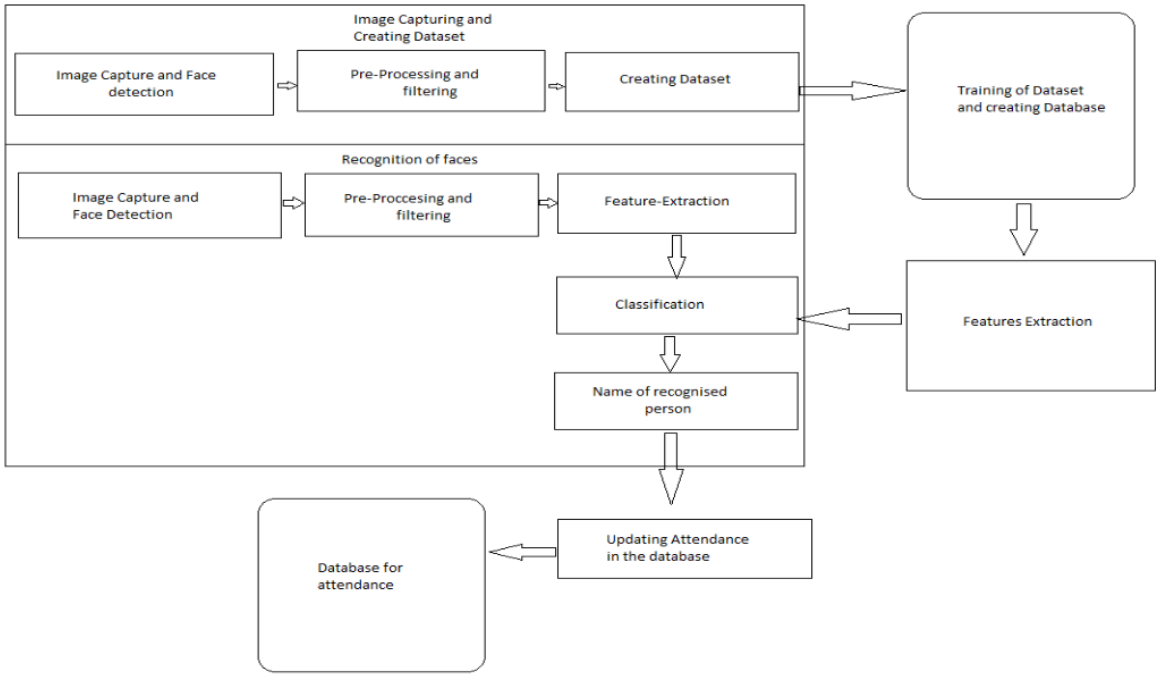


4.5Component Diagram-



5.DEVELOPMENT OF MODEL

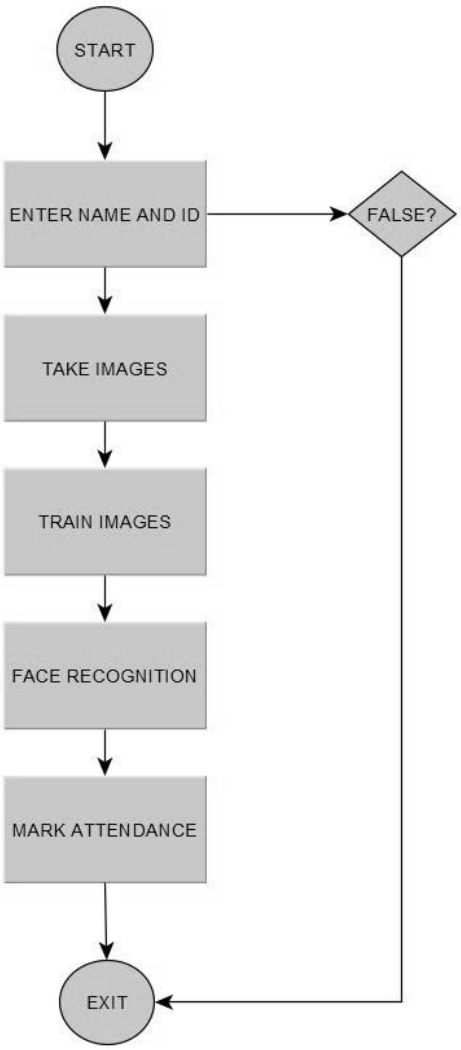
5.1Architecture of the System



- The user will create his login account, with input as name and ID.
- After creating the account successfully, the user can login into the website.
- After logging into the web application successfully, the user will enter the name and ID again, to verify the account.
- Once the verification is done successfully, 60 images will be taken from his webcam and stored into the image folder.
- After taking the images successfully, the model will be trained using the neural network and after training the model, it will be dumped into a pickle file.
- Once the model is trained and stored for the user, attendance will be taken by recognizing the face of the user.
- When the user takes images from his webcam, the encoding of the input images is done and stored into a list. The recorded encoding is matched with the stored encoding to verify the user.
- Click on ENTRY button, to mark the attendance with status “IN”, and click on EXIT button to mark attendance with status “OUT”.

- The attendance of every user is stored into the MySQL database.
- We can verify whether the user was absent or present on a particular date, from the report page.
- Now the user can logout from the system.

5.2 Process flow of the System



6. System components

Following are the system components which are necessary to make system complete and make the application run smoothly and without any errors. The mentioned components are required so that all the services of the application can be utilized.

- Microsoft Server 2016 OS
    - This is the operating system, on which the application was deployed and accessed by all other users.
- XAMPP
    - Server Application so that the web application can work with the required configurations and can be accessed by the authorized users.
- Microsoft SQL Server 2019
    - This server was used to design the database and store the data in the database
    - This SQL server was maintained as it was provided large amount of secured storage of data.
- MSSQL drivers
    - These drivers were required to make XAMPP work with Microsoft SQL Database Server
    - Using these drivers, we can connect application with database.
    - Driver names:
        - Microsoft ODBC Driver 17
        - Microsoft Drivers 5.6 for PHP named: php_pdo_sqlsrv_71ts.dll and php _sqlsrv_71ts.dll
    - The above-mentioned drivers are to be installed in location php/ext directory.
- Flask
- Supported Web browsers:

- Chrome (version 72 and above)
- Mozilla Firefox (version 60 and above)
- The system doesn't support Microsoft Edge as it didn't support few features of CSS and HTML

- LDAP server
  - LDAP server is required to authenticate person using this application
  - This application was only made accessible within the company domain, that means no other person outside the company can access this application

6.1 Modules and Libraries

Many modules and libraries were used by me in development of this web application. Those libraries helped me in designing of the portal with the advanced commands and in- built functions.

6.1.1 Face Detection.

Face detection is a type of application classified under "computer vision" technology. It is the process in which algorithms are developed and trained to properly locate faces or objects (in object detection, a related system), in images.

Haar Cascade Classifier, this uses machine learning techniques to get a high degree of accuracy from what is called "training data". This uses "integral image" concepts to compute the "features" detected. Haar Cascades use the Adaboost learning algorithm which selects a small number of important features from a large set to give an efficient result of classifiers.

Haar Cascades use machine learning techniques in which a function is trained from a lot of positive and negative images. This process in the algorithm is feature extraction.

```
import cv2 #computer vision

cam=cv2.VideoCapture(0) #camera on

harcascadePath = "haarcascade_frontalface_default.xml" #detect face
```

detector=cv2.CascadeClassifier(harcascadePath) #detected face

6.1.2 Training Model

Encoding process is done in the image dataset, and stored in encoding variable. This encoding is done also by using the face_recognition module and after successful training the model is dumped into the pickle file.

```
for(i,imgpath) in enumerate(imagepaths):
        name=imgpath.split(os.path.sep)[1]
        img=cv2.imread(imgpath)
        rgb=cv2.cvtColor(img,cv2.COLOR_BGR2RGB)
        boxes=face_recognition.face_locations(rgb,model="hog")
       encoding=face_recognition.face_encodings(rgb,boxes)
```

6.1.3 Face Recognition

The model consists of a fully connected layer with 128 hidden units followed by an L2 normalization layer on top of the convolutional base.

The two top layers are the ones responsible for creating 128-dimensional embeddings from images.

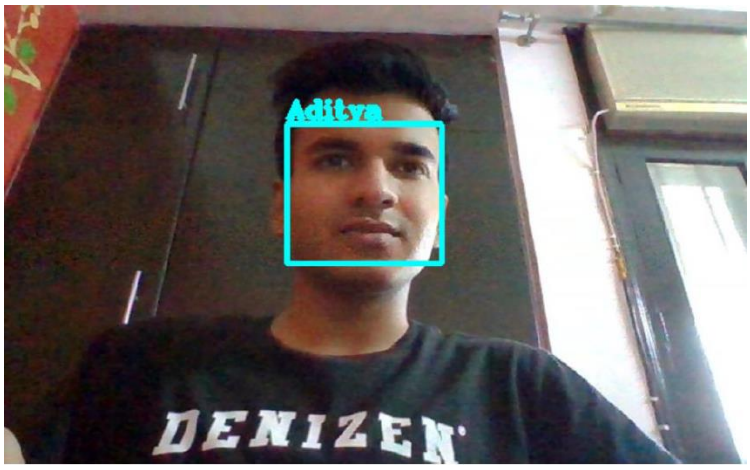


Figure 6: Recognized face

```
while 1:
```

```
ret,img=cap.read() #INPUT VALUES FROM CAM
rgb=cv2.cvtColor(img,cv2.COLOR_BGR2RGB)
face=face_recognition.face_locations(rgb,model="hog") #FINDING FACE
#encoding face
encodings=face_recognition.face_encodings(rgb,face)
finalName=[]
ts = time.time()
date = datetime.datetime.fromtimestamp(ts).strftime('%Y-%m-%d')
timeStamp = datetime.datetime.fromtimestamp(ts).strftime('%H:%M:%S')
Hour,Minute,Second=timeStamp.split(":")
status="IN"
row=[Ids,names,date,status,timeStamp]
date = datetime.datetime.fromtimestamp(ts).strftime('%Y-%m-%d')
for enc in encodings: #MATCHES ENCODING FROM DATASET
    matches=face_recognition.compare_faces(np.array(enc),np.array(data["encoding"]))
    name="Unknown"
    if True in matches:
        matchedID=[i for(i,b) in enumerate(matches) if b]
        for i in matchedID:
            name=data["name"][i]
        finalName.append(name)
```

6.1.4 dlib library

dlib library provides HOG+SVM face detector and provides also a pretrained CNN facial detector that can be ran using GPU or using CPU.

6.1.5 Bootstrap

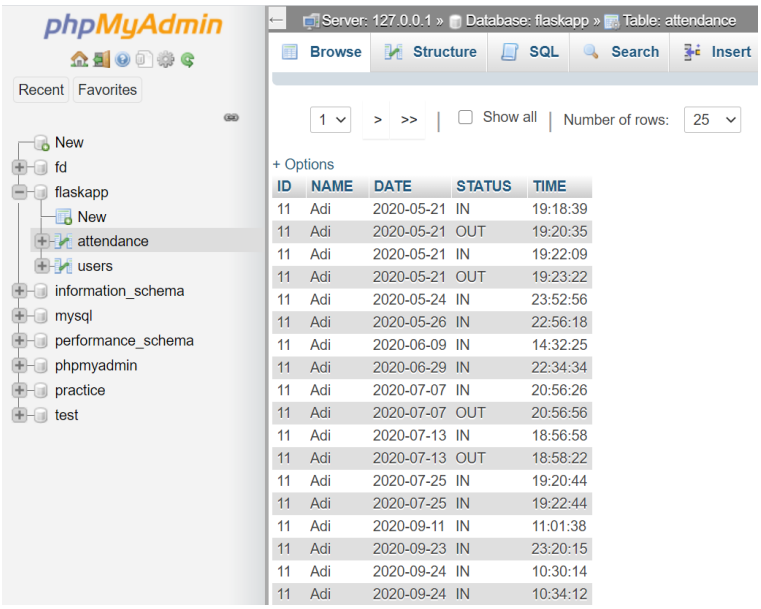
Bootstrap is a framework of HTML, JS and CSS which helps the developer in designing the front-end web page of the web application. This gives me a base on which I can create and develop multiple pages with consistency.

Bootstrap helped me design a dynamic and interactive portal. I used this toolkit for maintaining consistency in the design of different pages.

There are many predefined selectors in this toolkit. They help me to frame the page and even the animation within that page.

6.1.7 Database

6.1.7.1 Attendance



ID	NAME	DATE	STATUS	TIME
11	Adi	2020-05-21	IN	19:18:39
11	Adi	2020-05-21	OUT	19:20:35
11	Adi	2020-05-21	IN	19:22:09
11	Adi	2020-05-21	OUT	19:23:22
11	Adi	2020-05-24	IN	23:52:56
11	Adi	2020-05-26	IN	22:56:18
11	Adi	2020-06-09	IN	14:32:25
11	Adi	2020-06-29	IN	22:34:34
11	Adi	2020-07-07	IN	20:56:26
11	Adi	2020-07-07	OUT	20:56:56
11	Adi	2020-07-13	IN	18:56:58
11	Adi	2020-07-13	OUT	18:58:22
11	Adi	2020-07-25	IN	19:20:44
11	Adi	2020-07-25	IN	19:22:44
11	Adi	2020-09-11	IN	11:01:38
11	Adi	2020-09-23	IN	23:20:15
11	Adi	2020-09-24	IN	10:30:14
11	Adi	2020-09-24	IN	10:34:12

Figure 4: Attendance database.

6.1.7.2 Users

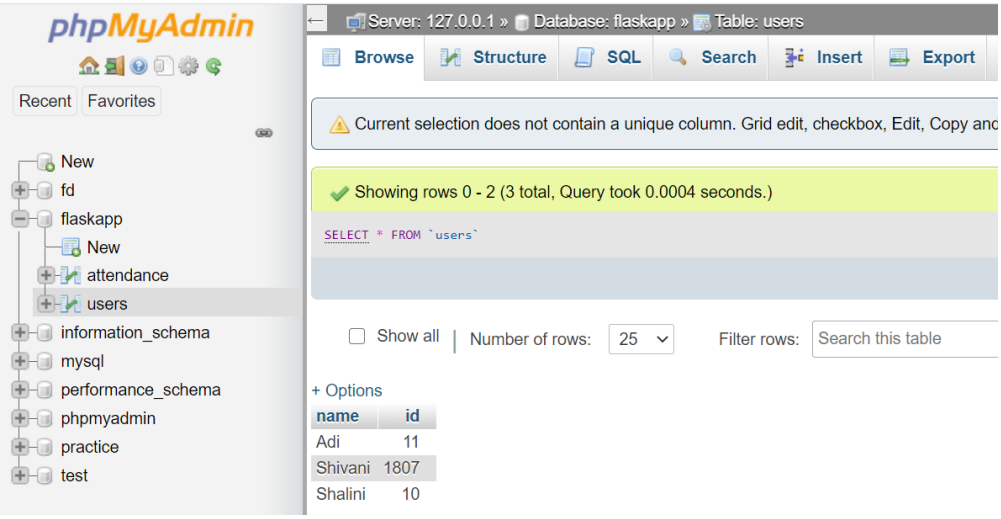


Figure 5: Users database

7.Software Testing methods

We can choose **White Box Testing** for the system.

Reasons to choose a specific testing method-

1. As the tester has knowledge of the source code, it becomes very easy to find out which type of data can help in testing the application effectively.
2. It helps in optimizing the code.
3. Extra lines of code can be removed which can bring in hidden defects.
4. Due to the tester's knowledge about the code, maximum coverage is attained during test scenario writing.

The system can also be tested by using **Grey Box Testing** with having a limited knowledge of the internal workings of an application.

- Since the access to source code is not available, the ability to go over the code and test coverage is limited.
- The tests can be redundant if the software designer has already run a test case.
- Testing every possible input stream is unrealistic because it would take an unreasonable amount of time; therefore, many program paths will go untested.
- Not suited for algorithm testing.

Software Testing table-

Test ID	Test Scenario	Test Case	Test Data	Expected Result	Actual Result	Pass/ Fail
T0001	Verify login page	Enter valid ID, Enter valid password	<valid ID> <valid password>	Login Success	Login Success	Pass
T0002	Verify login page	Enter invalid ID, Enter invalid password	<invalid ID> <invalid password>	Login Failed	Login Failed	Pass
T0003	Verify login page	Enter valid ID, Enter invalid password	<valid ID> <invalid password>	Login Failed	Login Failed	Pass
T0004	Verify login page	Enter invalid ID, Enter valid password	<invalid ID> <valid password>	Login Failed	Login Failed	Pass
T0005	Verify SignUp page	Enter char ID, Enter valid name	<invalid ID> <valid name>	SignUp Failed.	SignUp Failed	Pass
T0006	Verify SignUp page	Enter numerical ID, Enter numerical name	<valid ID> <invalid password>	SignUp Failed	SignUp Failed	Pass
T0007	Verify SignUp page	Enter numerical ID, char name	<valid ID> <valid name>	SignUp Success	SignUp Success	Pass
T0008	Verify Registration page.	Enter valid name, Enter valid ID	<valid ID> <valid name>	Student verified Success.	Student verified Success.	Pass

T0009	Verify Registration page.	Enter valid name, Enter invalid ID	<valid name> <invalid ID>	Student verification failed.	Student verification failed.	Pass
T0010	Verify Registration page.	Enter invalid name, Enter valid ID	<invalid name> <valid ID>	Student verification failed	Student verification failed	Pass
T0011	Verify Registration page.	Enter invalid name, Enter invalid ID	<invalid name> <invalid ID>	Student verification failed.	Student verified.	Pass
T0012	Verify attendance status	No image input	<no input>	Attendance status failed.	Attendance status failed.	Pass
T0013	Verify attendance status	Image not matched with the user.	<invalid image>	Attendance status failed.	Attendance status failed.	Pass
T0014	Verify attendance status	Correct image of user,	<valid image>	Attendance status success.	Attendance status success.	Pass

8. CONCLUSION

Thus, the aim of this project is to capture the video of the users, convert it into frames, relate it with the database to ensure their presence or absence, mark attendance to the particular student to maintain the record. The Automated Attendance System helps in increasing the accuracy and speed ultimately achieve the high-precision real-time attendance to meet the need for automatic attendance evaluation.

9. REFERENCES

1. Abdul Muqeet, Mohd. (2019). Face Recognition Based Attendance Management System Using Raspberry Pi. Journal of Information and Computational Science. 9. 86-90.

2. Chakravarthy, G. & Anupam, K. & Varma, P. & Teja, G. & Rodda, Sireesha. (2020). Face Recognition with Voice Assistance for the Visually Challenged. 10.1007/978-981-15-1084-7_68.

3. Ahuja, Karishma & K V, Anand Krishnan & Kiran, A. & Dalin, E.M. & Sagar, Shivaji. (2018). Smart Office Surveillance Robot using Face Recognition. International Journal of Mechanical and Production Engineering Research and Development. 8. 725-734. 10.24247/ijmperdjun201877.

4. Jagi, Shubha & Meenakshi, M.. (2020). Human Face Recognition Using Local Binary Pattern Algorithm - Real Time Validation. 10.1007/978-3-030-37218-7_28.

5. Patgar, Manjunath & Patil, Soumya & Shettar, Rajashekhar. (2019). A Face Recognition Based Automatic Attendance Management System by Combining LBP and HOG Features. 10.1007/978-3-030-03146-6_109.

6. Annamraju, Abhishek Kumar & Singh, Akash. (2015). Analysis and Optimization of Parameters used in Training a Cascade Classifier. Advances in Image and Video Processing (ISSN: 2054-7412). 3. 25-48. 10.14738/aivp.32.1152.

7. Pham, Minh-Tri & Cham, Tat-Jen. (2007). Fast training and selection of Haar features using statistics in boosting-based face detection. 1-7. 10.1109/ICCV.2007.4409038.

8. Bayasut, Bilal & Pramudya, Gede & Muda, Noor. (2011). Intelligent biometric detection system for disabled people. 346-350. 10.1109/HIS.2011.6122130.

9. Dong, Wenbo & Sun, Zhenan & Tan, Tieniu. (2009). A Design of Iris Recognition System at a Distance. 1 - 5. 10.1109/CCPR.2009.5344030.

10. Kim, Dong-Ju & Lee, Sang-Heon & Sohn, Myoung-Kyu. (2013). Face Recognition using Modified Local Directional Pattern Image. KIPS Transactions on Software and Data Engineering. 2. 205-208. 10.3745/KTSDE.2013.2.3.205.