A Project Report on

IMAGE-BASED ATTENDANCE SYSTEM

Submitted to partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE & ENGINEERING

By

GARIGA DIVYA	(17911A05D8)
RUDRARAJU AKHIL	(17911A05H1)
VENKATA VARUN KUMAR VEMULA	(17911A05H7)
YADAVA KRISHNA SHASHANK	(17911A05H8)

Under the Esteemed Guidance of

Dr. B. VijayaKumar

Professor

Department of Computer Science & Engineering

VIDYA JYOTHI INSTITUTE OF TECHNOLOGY

(An Autonomous Institution)

(Approved by AICTE, New Delhi & Affiliated to JNTUH, Hyderabad) Aziz Nagar Gate.
C.B. Post, Hyderabad-500075

2020-2021

A Project Report on

IMAGE-BASED ATTENDANCE SYSTEM

Submitted to partial fulfillment of the requirements for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE & ENGINEERING

By

GARIGA DIVYA	(17911A05D8)
RUDRARAJU AKHIL	(17911A05H1)
VENKATA VARUN KUMAR VEMULA	(17911A05H7)
YADAVA KRISHNA SHASHANK	(17911A05H8)

Under the Esteemed Guidance of

Dr. B. VijayaKumar

Professor

Department of Computer Science & Engineering

VIDYA JYOTHI INSTITUTE OF TECHNOLOGY

(An Autonomous Institution)

(Approved by AICTE, New Delhi & Affiliated to JNTUH, Hyderabad)

Aziz Nagar Gate, C.B. Post, Hyderabad-500075

2020-2021

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

This is to certify that the project report titled "**IMAGE-BASED ATTENDANCE SYSTEM**" is being submitted by **G.DIVYA (17911A05D8), R.AKHIL (17911A05H1), V.VARUN KUMAR VEMULA (17911A05H7), Y.KRISHNA SHASHANK (17911A05H8)** in partial fulfilment for the award of the Degree of Bachelor of Technology in **Computer Science & Engineering,** is a record of bonafide work carried out by us under my guidance and supervision.

These results embodied in this project report have not been submitted to any other University or Institute for the award of any degree of diploma.

Internal Guide
Dr. B. VijayaKumar
Professor

Head of Department
Dr. B. VijayaKumar
Professor

Head of the Department
Computer Science and Engineering
VJIT, Hyderabad-50075.

External Examiner

DECLARATION

We, G.DIVYA, R. AKHIL, V. VARUN KUMAR VEMULA, Y.KRISHNA SHASHANK hereby declare that the project entitled, **"IMAGE-BASED ATTENDANCE SYSTEM"** submitted for the degree of Bachelor of Technology in Computer Science and Engineering is original and has been done by us and this work is not copied and submitted anywhere for the award of any degree.

Project Associates:

Date:

Place: Hyderabad

G.DIVYA	**(17911A05D8)**
R. AKHIL	**(17911A05H1)**
V.VARUN KUMAR VEMULA	**(17911A05H7)**
Y.KRISHNA SHASHANK	**(17911A05H8)**

ACKNOWLEDGEMENT

We wish to express our sincere gratitude to the project guide, **Dr.B.Vijayakumar**, Professor, Vidya Jyothi Institute of Technology, Hyderabad for his timely cooperation and valuable suggestions while carrying out this work. It is his kindness that made us learn more from him.

We are grateful to **Dr. B. Vijayakumar**, Professor and HOD, department of CSE, for his help and support during our academic year.

We whole-heartedly convey our gratitude to Principal **Dr. A. Padmaja** for her constructive encouragement.

We would like to take this opportunity to express our gratitude to our Director **Dr. E. Sai Baba Reddy** for providing necessary infrastructure to complete this project.

We would thank our parents and all the faculty members who have contributed to our progress through the course to come to this stage.

Project Associates:

G.DIVYA	**(17911A05D8)**
R. AKHIL	**(17911A05H1)**
V.VARUN KUMAR VEMULA	**(17911A05H7)**
Y.KRISHNA SHASHANK	**(17911A05H8)**

ABSTRACT

The aim of this project is to mark attendance of the students attending classes, this application captures the student's face at frequent intervals of time set by the mentor, this captured face is compared with the student photo in the database and it produces an output containing the attendance of the students present during that class.

The existing system has fingerprint or facial biometrics or traditional roll call of the students which is time consuming, repetitive and especially in the case of calling out student's names there is a chance of proxy attendance hence the new system uses image processing which makes the attendance marking automatic, genuine and doesn't require any dedicated time just to note the presence of the students.

Contents

List Of Figures

CHAPTER-1

INTRODUCTION

In today's networked world, the need to maintain the security of information or physical property is becoming both increasingly important and increasingly difficult. From time to time, we hear about the crimes of credit card fraud, computer breaking's by hackers, or security breaches in a company or government building. In most of these crimes, the criminals were taking advantage of a fundamental flaw in the conventional access control systems: the systems do not grant access by "who we are", but by "what we have", such as ID cards, keys, passwords, PIN numbers, or mother's maiden name. None of these means are really define us. Recently, technology became available to allow verification of "true" individual identity. This technology is based in a field called "biometrics".

Biometric access control are automated methods of verifying or recognizing the identity of a living person on the basis of some physiological characteristics, such as fingerprints or facial features, or some aspects of the person's behavior, like his/her handwriting style or keystroke patterns. Since biometric systems identify a person by biological characteristics, they are difficult to forge. Face recognition is one of the few biometric methods that possess the merits of both high accuracy and low intrusiveness. It has the accuracy of a physiological approach without being intrusive. For this reason, since the early 70's (Kelly, 1970), face recognition has drawn the attention of researchers in fields from security, psychology, and image processing, to computer vision.

The technology aims in imparting a tremendous knowledge oriented technical innovation these days. Deep Learning is one among the interesting domain 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.

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

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

Manual Student Attendance Management system is a process where a teacher concerned with the particular subject need to call the students name and mark the attendance manually. Manual attendance may be considered as a time-consuming process or sometimes it happens for the teacher to miss someone or students may answer multiple times on the absence of their friends. So, the problem arises when we think about the traditional process of taking attendance in the classroom. To solve all these issues we go with Automatic Attendance System (AAS)

Automated Attendance System (AAS) is a process to automatically estimate the presence or the absence of the student in the classroom by using face recognition technology. It is also possible to recognize whether the student is sleeping or awake during the lecture and it can also be implemented in the exam sessions to ensure the presence of the student. The presence of the students can be determined by capturing their faces on to a high-definition monitor video streaming service, so it becomes highly reliable for the machine to understand the presence of all the students in the classroom. 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.

CHAPTER-2

LITERATURE SURVEY

A Counterpart Approach to Attendance and Feedback System using Machine Learning Techniques:

In this paper, the idea of two technologies namely Student Attendance and Feedback system has been implemented with a machine learning approach. This system automatically detects the student performance and maintains the student's records like attendance and their feedback on the subjects like Science, English, etc. Therefore, the attendance of the student can be made available by recognizing the face. On recognizing, the attendance details and details about the marks of the student is obtained as feedback.

Automated Attendance System Using Face Recognition:

Automated Attendance System using Face Recognition proposes that the system is based on face detection and recognition algorithms, which is used to automatically detects the student face when he/she enters the class and the system is capable to marks the attendance by recognizing him. Viola-Jones Algorithm has been used for face detection which detect human face using cascade classifier and PCA algorithm for feature selection and SVM for classification. When it is compared to traditional attendance marking this system saves the time and also helps to monitor the students

Student Attendance System Using Iris Detection: In this proposed system the student is requested to standing front of the camera to detect and recognize the iris, for the system to mark attendance for the student. Some algorithms like Gray Scale Conversion, Six Segment Rectangular Filter, Skin Pixel Detection is being used to detect the iris. It helps in preventing the proxy issues and it maintains the attendance of the student in an effective manner, but in one of the time-consuming process for a student or a staff to wait until the completion of the previous members.

Face Recognition-based Lecture Attendance System: This paper proposes that the system takes the attendance automatically recognition obtained by continuous

observation. Continuous observation helps in estimating and improving the performance of the attendance. To obtain the attendance, positions and face images of the students present in the class room are captured. Through continuous observation and recording the system estimates seating position and location of each student for attendance marking. The work is focused on the method to obtain the different weights of each focused seat according to its location. The effectiveness of the picture is also being discussed to enable the faster recognition of the image.

Face recognition is one of the few biometric methods that possess the merits of both high accuracy and low intrusiveness. It has the accuracy of a physiological approach without being intrusive. Over past 30 years, many researchers have proposed different face recognition techniques, motivated by the increased number of real world applications requiring the recognition of human faces. There are several problems that make automatic face recognition a very difficult task. However, the face image of a person inputs to the database that is usually acquired under different conditions. The important of automatic face recognition is much be cope with numerous variations of images of the same face due to changes in the following parameters such as

- Pose
- Illumination
- Expression
- Motion
- Facial hair
- Glasses
- Background of image.

Face recognition technology is well advance that can applied for many commercial applications such as personal identification, security system, image- film processing, psychology, computer interaction, entertainment system, smart card, law enforcement, surveillance and so on. Face recognition can be done in both a still image and video

sequence which has its origin in still-image face recognition. Different approaches of face recognition for still images can be categorized into three main groups.

1. **Holistic approach:** In holistic approach or global feature, the whole face region is taken into account as input data into face detection system. Examples of holistic methods are eigenfaces (most widely used method for face recognition), probabilistic eigenfaces, fisher faces, support vector machines, nearest feature lines (NFL) and independent-component analysis approaches. They are all based on principal component-analysis. (PCA) techniques that can be used to simplify a dataset into lower dimension while retaining the characteristics of dataset.

2. **Feature-based approach:** - In feature-based approaches or local feature that is the features on face such as nose, and then eyes are segmented and then used as input data for structural classifier Wavelets, especially Gabor wavelets, play a building block role for facial representation in these graph matching methods. A typical local feature representation consists of wavelet coefficients for different scales and rotations based on fixed wavelet bases. These locally estimated wavelet coefficients are robust to illumination change, translation, distortion, rotation, and scaling. Recognition of a new image takes place by transforming the image into the grid of jets, and matching all stored model graphs to the image. Conformation of the DLA is done by establishing and dynamically modifying links between vertices in the model domain.

3. **Hybrid approach:** The idea of this method comes from how human vision system perceives both holistic and local feature. The key factors that influence the performance of hybrid approach include how to determine which features should be combined and how to combine, so as to preserve their advantages and avert their disadvantages at the same time. These problems have close relationship with the multiple classifier system (MCS) and ensemble learning in the field of machine learning. Unfortunately, even in these fields, these problems remain unsolved. In spite of this, numerous efforts made in these fields indeed provide us some insights into solving these problems, and these lessons can be used as guidelines in designing a hybrid face recognition system.

hybrid approach that use both holistic and local information for recognition may be an effective way to reduce the complexity of classifiers and improve their generalization capability.

Related Work

Hajar Filaliet. al. had compared four methods based on machine learning that allows a machine to evolve through a learning process, and to perform tasks that are quite difficult to fill by more conventional algorithmic mean (Haar-AdaBoost, LBP-AdaBoost, GF-SVM, GFNN).The first two methods "Haar-AdaBoost, LBP-AdaBoost" are based on the Boosting algorithm, which is used both for selection and for learning a strong classifier with a cascade classification. While the last two classification methods "GF-SVM, GF-NN" use the Gabor filter to extract the characteristics. From this study, we found that the detection time varies from one method to another. in terms of performance rate, the Haar-AdaBoost method remains the best of the four methods. So, we will be using HaarAdaBoost.

The authors have proposed a system to avoid drawbacks of traditional manual attendance system. This paper describes how real time face detection and recognition can prove useful for marking attendance of students. The paper illustrates an automated attendance system which consists of a camera, installed in the classroom for capturing images, followed by the detection of multiple faces. This system consists of multiple steps such as Face Database Creation of Students, HOG features, Face Detection and Eye Detection, SVM Classifier, Comparison/Recognition, Attendance marking. The algorithms like Viola-Jones and HOG features along with SVM classifier are used to acquire the desired results. The paper had certain drawbacks where the system may be sensitive to lighting. The proposed system may eliminate this drawback by using the algorithms which may not be sensitive to lighting and also by using advanced high-resolution cameras.

E. Varadharaja et. al proposed a system for automatic attendance system using face recognition. The system is divided into four parts. First one is the Background Subtraction in which background of image is subtracted and only face remains in image. Second part is face detection and cropping of images i.e. only faces are cropped and stored. Third step is recognizing images with the help of Eigenvalue method. In this

method Eigen vectors are calculated using formulae and to recognize images Euclidean distance is calculated between stored images and testing image. Then attendance is marked for matching student. This method requires simple hardware installation but face recognition is difficult. Eigen vector method used in this paper gives an accuracy of 60-70%. Hence instead of using eigen vector, proposed system will use Haar features for face detection which gives better result than eigen vector method.

Shireesha Chintalapati, M.V. Raghunadh et. al [8] had described the different techniques to implement the attendance monitoring system using face recognition. The process is divided into two main parts. First one is the face detection technique and the second one is face recognition technique. Face detection can be implemented using Viola-Jones face detection algorithm which includes four main components i.e., Haar-features, integral image, Adaboost algorithm, cascade feature. Face recognition can be implemented using LBP (local binary patterns). LBP helps to convert the image into machine understandable formats i.e., in binary format. Before face detection and recognition, the captured image should be converted into grey scale to simplify the calculation. Face detection technique first capture the image (student dataset) and detect faces from the images; the detected faces are stored for further reference. Face recognition technique capture the images from classroom and try to recognize it by comparing it with a earlier detected face.

In conclusion, a better attendance monitoring system should be developed based on its portability, accessibility and the accuracy of the collected attendance information.

CHAPTER-3

SYSTEM REQUIREMENT SPECIFICATIONS

3.1 Existing Systems:

3.1.1 Fingerprint Based recognition system

In the Fingerprint based existing attendance system, a portable fingerprint device needs to be configured with the students fingerprint earlier. Later either during the lecture hours or before, the student needs to record the fingerprint on the configured device to ensure their attendance for the day. The problem with this approach is that during the lecture time it may distract the attention of the students.

3.1.2 RFID (Radio Frequency Identification) Based recognition system

In the RFID based existing system, the student needs to carry a Radio Frequency Identity Card with them and place the ID on the card reader to record their presence for the day. The system is capable of to connect to RS232 and record the attendance to the saved database. There are possibilities for the fraudulent access may occur. Some are students may make use of other students ID to ensure their presence when the particular student is absent or they even try to misuse it sometimes in Manual attendance marking for students

3.2 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, even the seating and the posture of the student need to be recognized. 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.

3.3 System Requirements

3.3.1 Software Requirements

- Python IDE: python 3.9.x and above: PyCharm
- Setup tools and pip to be installed for 3.9.x and above
- Python Modules: face-recognition, face_detection, flask,
- OS: Windows
- Visual Studio

3.3.2 Hardware Requirements

- RAM: 4GB and Higher
- Processor: Intel i3 and above
- Hard Disk: 500GB: Minimum
- Camera

3.4 Requirements Definition

3.4.1 Functional Requirements

- User must be able to manage student records.
- User must be able to see details of attendance of students for specified date.
- Only authorized user must be able to use the system.
- User must be able to print records.
- System must be attached to webcam and face recognition should be smooth.
- The administrator or the person who will be given the access to the system must login into system before using it.

- The information must be entered properly
- The information must be managed properly.

3.4.2 Non-Functional Requirements

- The GUI of the system will be user friendly.
- The data that will be showed to the users will be made sure that it is correct and is available for the time being.
- The system will be flexible to changes.
- The system will be extensible for changes and to the latest technologies.
- Efficiency and effectiveness of the system will be made sure.
- The performance of the system will be made sure.
- It will be made sure that the result of search must be according to the specifications entered by the user.
- It will be ensured sure that user must get the correct information for their search.
- User must get right and related information about the attendance record they are trying to view.
- The data to be edited will be made sure that the correct data is being edited.
- The data to be deleted will be confirmed first.
- The response time of the system will be effective.

CHAPTER-4

SOFTWARE AND HARDWARE ENVIRONMENT

4.1 Python IDE

4.1.1 Introduction

Python is a universally useful deciphered, intuitive, object-situated, and high-level programming language. Python was made during the year 1985-1990 by Guido van Rossum. As like Perl, Python source code can be accessible in the GNU General Public License (GPL). Python programming language utilizes little English catchphrases regularly where as other programming languages utilizes hard syntax, and it has less grammatical development when contrasted with different languages. Python is known to be straightforward uncommon programming language.

Python is known to be simple and easy to understandable rarely available programming language.

- **Interpreted** – Python is easy to learn and has couple of watchwords that has straightforward structure, and furthermore have unmistakably characterized linguistic structure, which permits it. Python is prepared by the translator at the runtime just, so we don't have to accumulate our program before executing it. This model is like that of the PERL and PHP the understudies to select the language rapidly.

- **Interactive** – Compilation can be done at Python prompt and can be interacted with the interpreter directly to write our programs.

- **Object-Oriented** – Python provides support for Object-Oriented programming or technique that can encapsulates the code within objects.

- **Python is a Beginner's Language** – Python is a once in a while accessible extraordinary programming language for the fledgling level software engineers and it is additionally a language that supports for the advancement of numerous wide scopes of utilizations from little content handling to WWW programs and furthermore to diversions.

Following are the Python's features –

- Python is not difficult to learn and has couple of catchphrases that has straightforward structure, and furthermore have plainly characterized grammar, which enables the understudies to choose the language rapidly.

- Python's source code is very easy to read and more clearly defined and easily understandable.

- **Easy-to-maintain** – The source code of python is very easy-to-maintaining.

- **A well-defined standard library** – Most of the Python's libraries are portable and are cross-platform which are compatible on many platforms like UNIX, Windows, and Macintosh.

- **Extendable** – Low-level features can be added to the Python interpreter. These modules make the programmers to use them or modify these modules to be more effective.

- **Interactive Mode** – Python provides easy interaction mode that allows interactive testing and removing errors.

- **Portable** – Python software is able to run on variety of different platforms that makes us to use the same type of interface on different platforms.

- **Databases** – Python programming gives interfaces to all the real databases.

- **GUI Programming** – Python gives underpins numerous GUI applications that are made and imported to a considerable lot of the framework calls, libraries and furthermore windows frameworks, for example, Windows, Macintosh, and furthermore the Window frameworks of Unix.

- **Scalable** - Python gives a very much characterized structure and furthermore offers help for huge projects when contrasted with shell scripting.

4.1.2 Functions in Python

It is possible, and extremely valuable, to characterize our very own functions in Python. As a rule, in the event that you have to complete a figuring just once, at that point utilize the interpreter.

But, when you or others have to use out a particular sort of function many times ordinarily, at that point characterize a function of your own.

You can use works in programming to package a lot of guidelines that you need to utilize over and over or that, in light of their own contained nature, are better independent in a sub-program and called when required. That implies that their capacity is a bit of code written to do a predetermined assignment.

To do a particular assignment, the capacity may or probably may won't require numerous data sources. At the point when the assignment is carried out, the capacity can or cannot return at least one quality.

The three types of functions in python:

help (), min(), print()

4.1.3 Variables

The term variables in programming language refer to the memory locations that are used to store values. So, when we create a variable, the interpreter allocates some memory to the variable based on the datatype of the variable and it also decides which type of data to be stored in the memory reserved for that variable. So, by using different datatypes we can store different data which are useful for our programming. For example, we can store different data such as integers, characters, strings and also decimal numbers based on our requirement.

4.1.4 Standard Data Types

When writing programs, we need to work with different types of data. Python provides provision to work with different types of data such as integers, decimal numbers, characters, strings and many more. Python provides various datatypes that

help us to use different type of data. Based on the type of data the interpreter automatically allocates memory for the data.

The following are the various standard datatypes in python:

- Numeric
- String
- List
- Tuple
- Dictionary

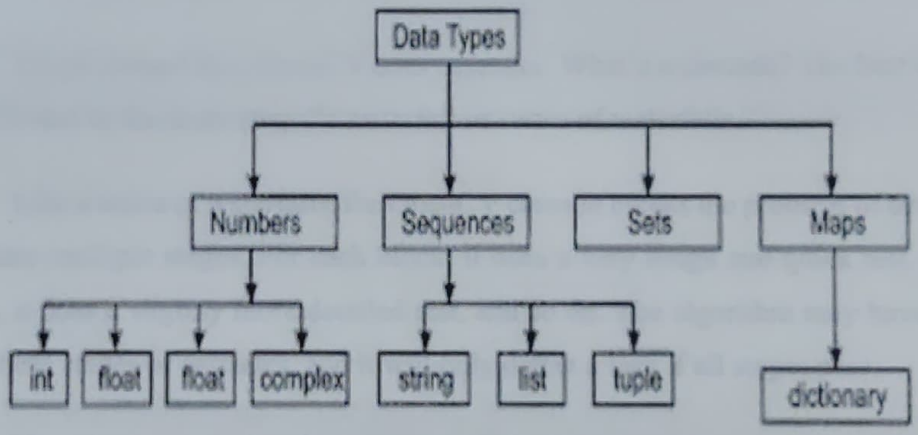


Figure 4.1.4 : Python Data Types

4.2 Libraries used

4.2.1 Open CV

Computer vision is a process by which we can understand the images and videos how they are stored and how we can manipulate and retrieve data from them. Computer Vision is the base or mostly used for Artificial Intelligence. Computer-Vision is playing a major role in self-driving cars, robotics as well as in photo correction apps.

OpenCV is the most popular library for computer vision. Originally written in C/C++, it now provides bindings for Python.

OpenCV uses machine learning algorithms to search for faces within a picture. Because faces are so complicated, there isn't one simple test that will tell you if it found a face or not. Instead, there are thousands of small patterns and features that must be matched. The algorithms break the task of identifying the face into thousands of smaller, bite-sized tasks, each of which is easy to solve. These tasks are also called classifiers.

For something like a face, you might have 6,000 or more classifiers, all of which must match for a face to be detected (within error limits, of course). But therein lies the problem: for face detection, the algorithm starts at the top left of a picture and moves down across small blocks of data, looking at each block, constantly asking, "Is this a face? … Is this a face? … Is this a face?" Since there are 6,000 or more tests per block, you might have millions of calculations to do, which will grind your computer to a halt.

To get around this, OpenCV uses cascades. What's a cascade? The best answer can be found in the dictionary: "a waterfall or series of waterfalls."

Like a series of waterfalls, the OpenCV cascade breaks the problem of detecting faces into multiple stages. For each block, it does a very rough and quick test. If that passes, it does a slightly more detailed test, and so on. The algorithm may have 30 to 50 of these stages or cascades, and it will only detect a face if all stages pass.

The advantage is that the majority of the picture will return a negative during the first few stages, which means the algorithm won't waste time testing all 6,000 features on it. Instead of taking hours, face detection can now be done in real time.

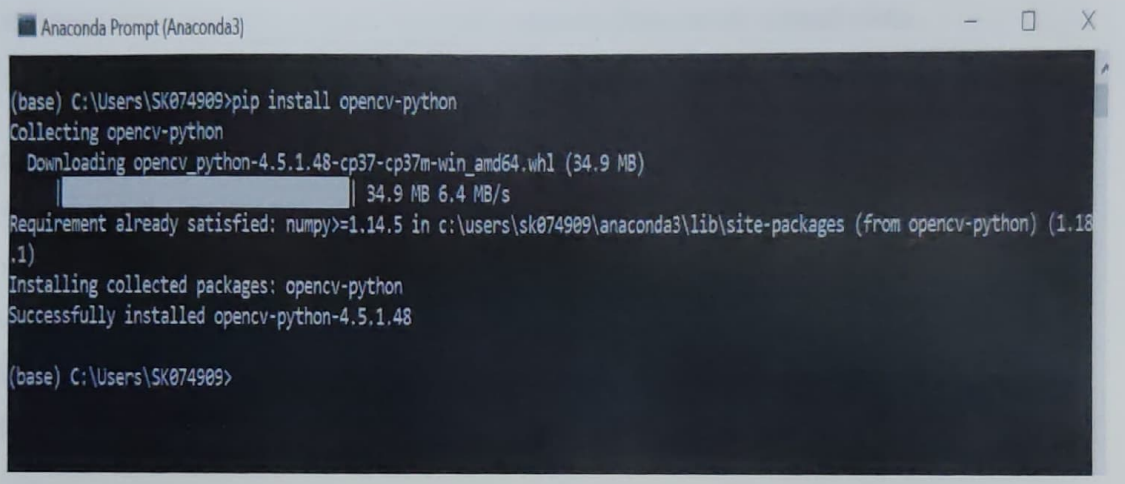
4.2.2 OpenCV Functionality

- Image/video I/O, processing, display (core, imgproc, highgui)
- Object/feature detection (objdetect, features2d, nonfree)
- Geometry-based monocular or stereo computer vision (calib3d, stitching, videostab)
- Computational photography (photo, video, superres)
- Machine learning & clustering (ml, flann)

4.2.3 Installation

Here we will be focusing on installing OpenCV for python only. We can install OpenCV using pip by using the following command in the command prompt.

pip install opencv-python



4.2.4 Applications of OpenCV

There are lots of applications which are solved using OpenCV, some of them are listed below

- Face recognition
- Number of people – count (foot traffic in a mall, etc.)
- Vehicle counting on highways along with their speeds
- Anomaly (defect) detection in the manufacturing process (the odd defective products)
- Video/image search and retrieval
- Object recognition
- Medical image analysis
- Movies – 3D structure from motion

4.3 Framework used

4.3.1 Flask

Flask is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries. ... Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies and several common framework related tools.

It is an API of Python that allows us to build up web-applications. It was developed by Armin Ronacher. Flask's framework is more explicit than Django's framework and is also easier to learn because it has less base code to implement a simple web-Application. A Web-Application Framework or Web Framework is the collection of modules and libraries that helps the developer to write applications without writing the low-level codes such as protocols, thread management, etc. Flask is based on WSGI (Web Server Gateway Interface) toolkit and Jinja2 template engine).

4.3.2 Features

- Development server and debugger
- Integrated support for unit testing
- RESTful request dispatching
- Uses Jinja templating
- Support for secure cookies (client-side sessions)
- 100% WSGI 1.0 compliant
- Unicode-based
- Extensive documentation
- Google App Engine compatibility
- Extensions available to enhance features desired

4.4 Modules Used

4.4.1 Face Detection

In computer vision, one essential problem we are trying to figure out is to automatically detect objects in an image without human intervention. Face detection can be thought of as such a problem where we detect human faces in an image. There may be slight differences in the faces of humans but overall, it is safe to say that there are certain features that are associated with all the human faces. There are vario us face detection algorithms but Viola-Jones Algorithm is one of the oldest methods that is also used today.

Face detection is usually the first step towards many face-related technologies, such as face recognition or verification. However, face detection can have very useful applications. The most successful application of face detection would probably be photo taking. When you take a photo of your friends, the face detection algorithm built into your digital camera detects where the faces are and adjusts the focus accordingly.

4.4.2 Face Recognition

Recognize and manipulate faces from Python or from the command line with the world's simplest face recognition library. Built using dib's state-of-the-art face recognition built with deep learning. The model has an accuracy of 99.38% on the Labeled Faces in the Wild benchmark. This also provides a simple face recognition command line tool that lets you do face recognition on a folder of images from the command line.

In order to perform face recognition with Python and OpenCV we need to install two additional libraries:

- dlib
- face- recognition

To install face-recognition, type in command prompt

pip install face-recognition

4.4.3 Applications

- Face Recognition is a well-researched problem and is widely used in both industry and in academia. As an example, a criminal in China was caught because a Face Recognition system in a mall detected his face and raised an alarm. Clearly, Face Recognition can be used to mitigate crime. There are many other interesting use cases of Face Recognition:

- Facial Authentication: Apple has brought in Face ID for Facial Authentication in iPhones. Some of the leading banks are trying to use Facial Authentication for lockers.

- Customer Service: Some of the banks in Malaysia have installed systems which use Face Recognition to detect valuable customers of the bank so that the bank can provide the personalized service. This way, banks are able to generate more revenues by retaining such customers and keeping them happy.

- Insurance Underwriting: Many insurance companies are using Face Recognition to match the face of the person with that provided in the photo ID proof. This way, the underwriting process becomes much faster.

4.5 Visual Studio using C++ Library

Microsoft Visual C++ (MSVC) refers to the C++, C, and assembly language development tools and libraries available as part of Visual Studio on Windows. These tools and libraries let you create Universal Windows Platform (UWP) apps, native Windows desktop and server applications, cross-platform libraries and apps that run on Windows, Linux, Android, and iOS, as well as managed apps and libraries that use the .NET Framework. You can use MSVC to write anything from simple console apps to the most sophisticated and complex apps for Windows desktop, from device drivers and operating system components to cross-platform games for mobile devices, and from the smallest IoT devices to multi-server high performance computing in the Azure cloud.

Visual Studio 2015, 2017 and 2019 can be installed side-by-side. You can use Visual Studio 2019 (compiler toolset v142) or Visual Studio 2017 (v141) to edit and build

programs using the toolset from Visual Studio 2017 (v141) and Visual Studio 2015 (v140).

4.6 Hardware

4.6.1 Camera

A webcam is a compact digital camera you can hook up to your computer to broadcast video images in real time (as they happen). Just like a digital camera, it captures light through a small lens at the front using a tiny grid of microscopic light-detectors built into an image-sensing microchip (either a charge-coupled device (CCD) or, more likely these days, a CMOS image sensor). As we'll see in a moment, the image sensor and its circuitry convert the picture in front of the camera into digital format—a string of zeros and ones that a computer knows how to handle. Unlike a digital camera, a webcam has no built-in memory chip or flash memory card: it doesn't need to "remember" pictures because it's designed to capture and transmit them immediately to a computer. That's why webcams have USB cables coming out of the back. The USB cable supplies power to the webcam from the computer and takes the digital information captured by the webcam's image sensor back to the computer—from where it travels on to the Internet. Some cams work wirelessly and don't need to be connected to a computer: typically, they use Wi-Fi to transmit their pictures to your Internet router, which can then make them available to other machines on your home network or, using the Internet, to anyone, anywhere in the world

4.6.2 Processor

A central processing unit, also called a central processor, main processor or just processor, is the electronic circuitry that executes instructions comprising a computer program. The CPU performs basic arithmetic, logic, controlling, and input/output operations specified by the instructions in the program

Ex

- Intel i3, Intel i5, Intel i7
- AMD Ryzen 3, Ryzen 5, Ryzen 7

The basic elements of a processor include:

1. The arithmetic logic unit (ALU), which carries out arithmetic and logic operations on the operands in instructions.

2. The floating point unit (FPU), also known as a math coprocessor or numeric coprocessor, a specialized coprocessor that manipulates numbers more quickly than the basic microprocessor circuitry can.

3. Registers, which hold instructions and other data. Registers supply operands to the ALU and store the results of operations.

4. L1 and L2cache memory. Their inclusion in the CPU saves time compared to having to get data from random access memory (RAM).

4.6.3 RAM

RAM (Random Access Memory) is the hardware in a computing device where the operating system (OS), application programs and data in current use are kept so they can be quickly reached by the device's processor. RAM is the main memory in a computer. It is much faster to read from and write to than other kinds of storage, such as a hard disk drive (HDD), solid-state drive (SSD) or optical drive.

RAM allows your computer to perform many of its everyday tasks, such as loading applications, browsing the internet, editing a spreadsheet, or experiencing the latest game. Memory also allows you to switch quickly among these tasks, remembering where you are in one task when you switch to another task. As a rule, the more memory you have, the better.

Random Access Memory is volatile. That means data is retained in RAM as long as the computer is on, but it is lost when the computer is turned off. When the computer is rebooted, the OS and other files are reloaded into RAM, usually from an HDD or SSD.

CHAPTER-5

SYSTEM DESIGN

5.1 System Architecture

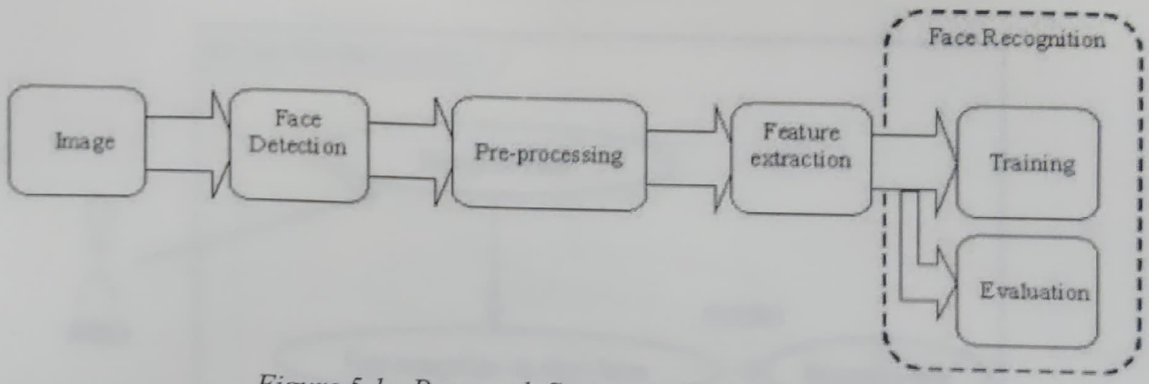


Figure 5.1 : Proposed System Architecture

5.2 UML Diagrams

UML diagram is a diagram that is designed based on Unified Modelling language with the aim to visually represent the system with roles, actors, anchors etc to understand and maintain the system easily. By using this we can better understand flaws or errors in the system so that we can maintain or alter the system properly.

Different types of UML diagrams include:

- Class diagram
- Activity diagram
- Use case diagram
- Sequence diagram
- Component diagram and many more.

5.2.1 Use Case Diagram

Use case diagram also represented as behavior diagrams. These diagrams are used to explain the set of actions that system need to be performed in accordance with the external use.

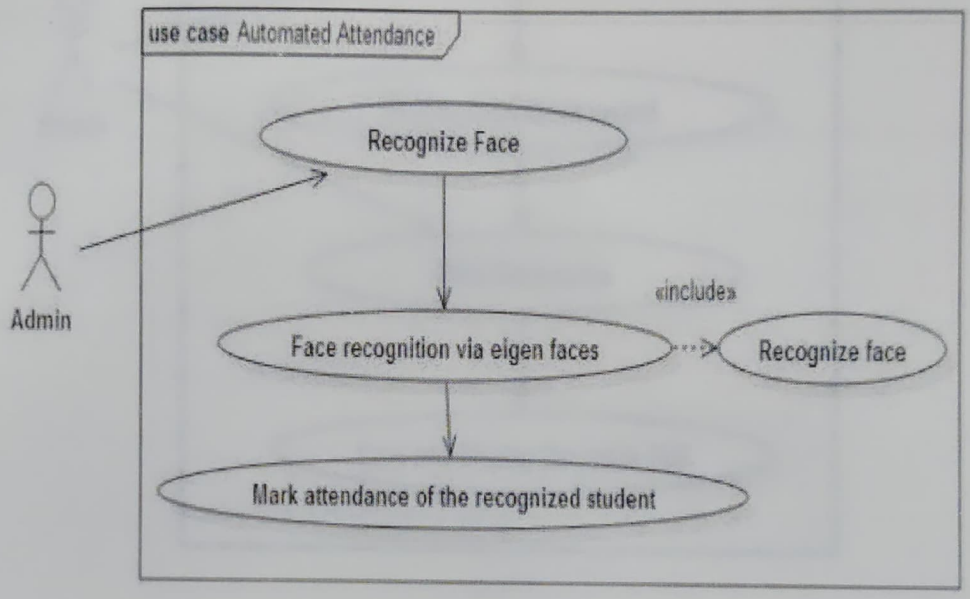


Figure 5.2.1.1: Use case diagram for automated attendance

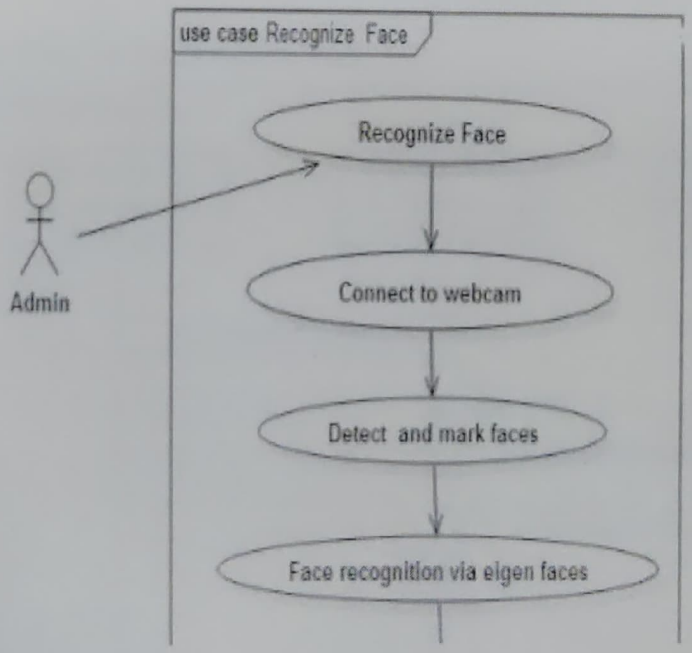


Figure 5.2.1.2: Use case diagram for face recognize

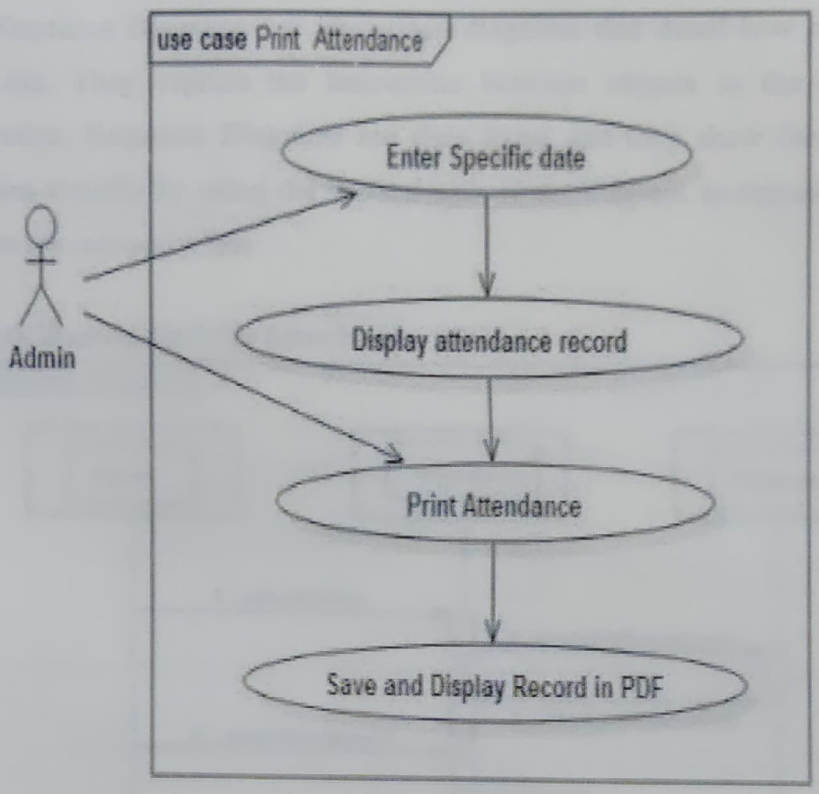


Figure 5.2.1.3: Use case diagram for marking attendance.

5.2.2 Sequence diagram

Sequence Diagrams are interaction diagrams that detail how operations are carried out. They capture the interaction between objects in the context of a collaboration. Sequence Diagrams are time focus and they show the order of the interaction visually by using the vertical axis of the diagram to represent time what messages are sent and when.

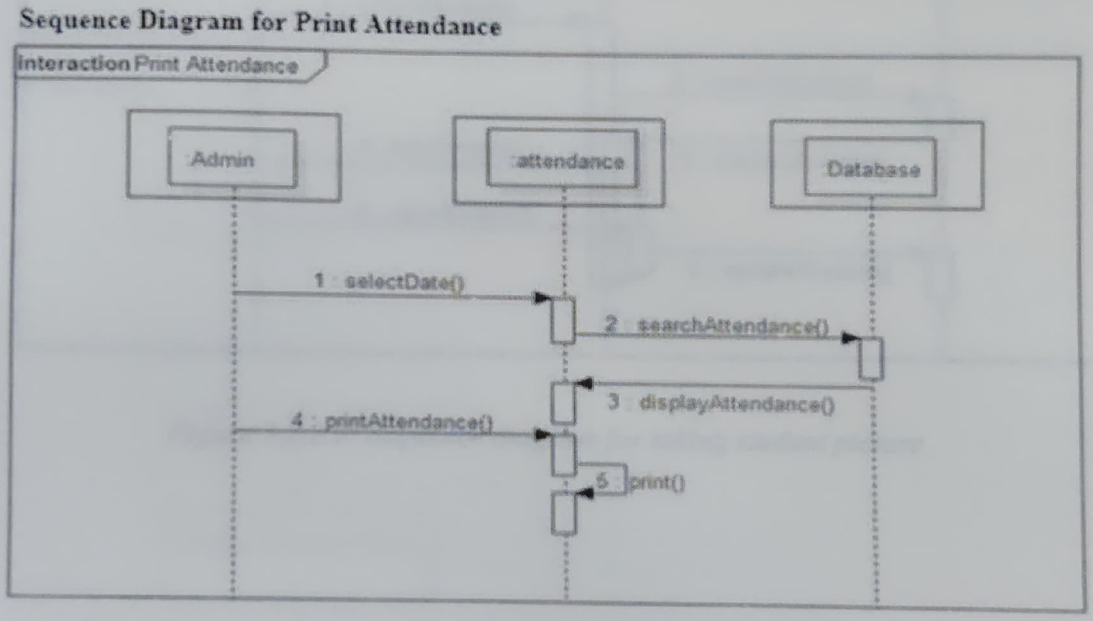


Figure 5.2.2.1: Sequence diagram for printing attendance

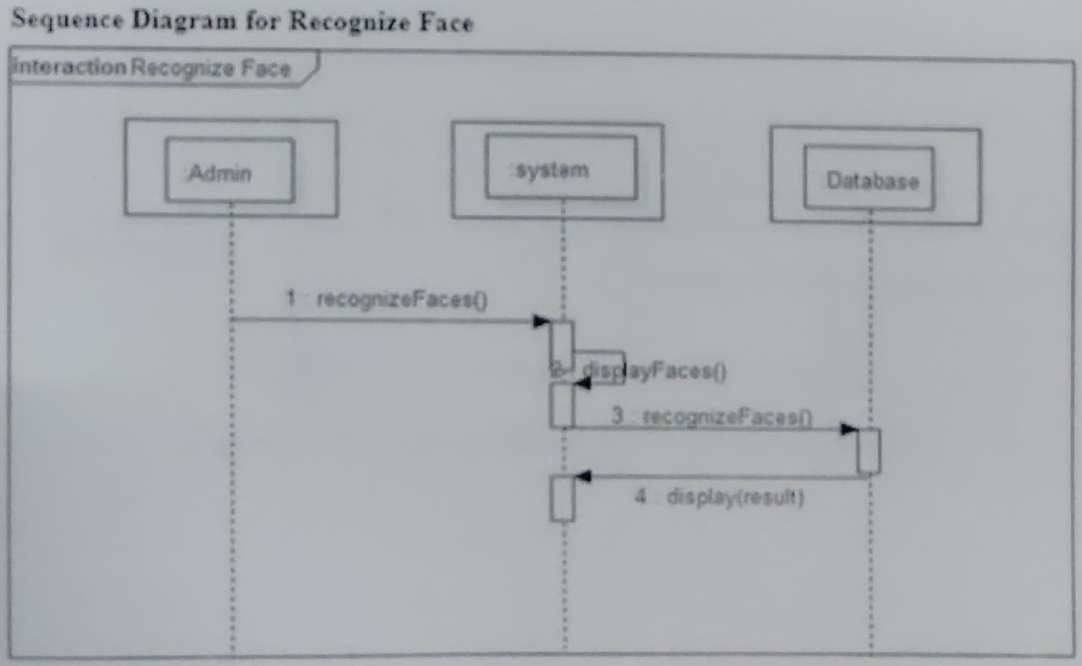


Figure 5.2.2.2: Sequence diagram for face recognize

Sequence Diagram for Take Student Picture

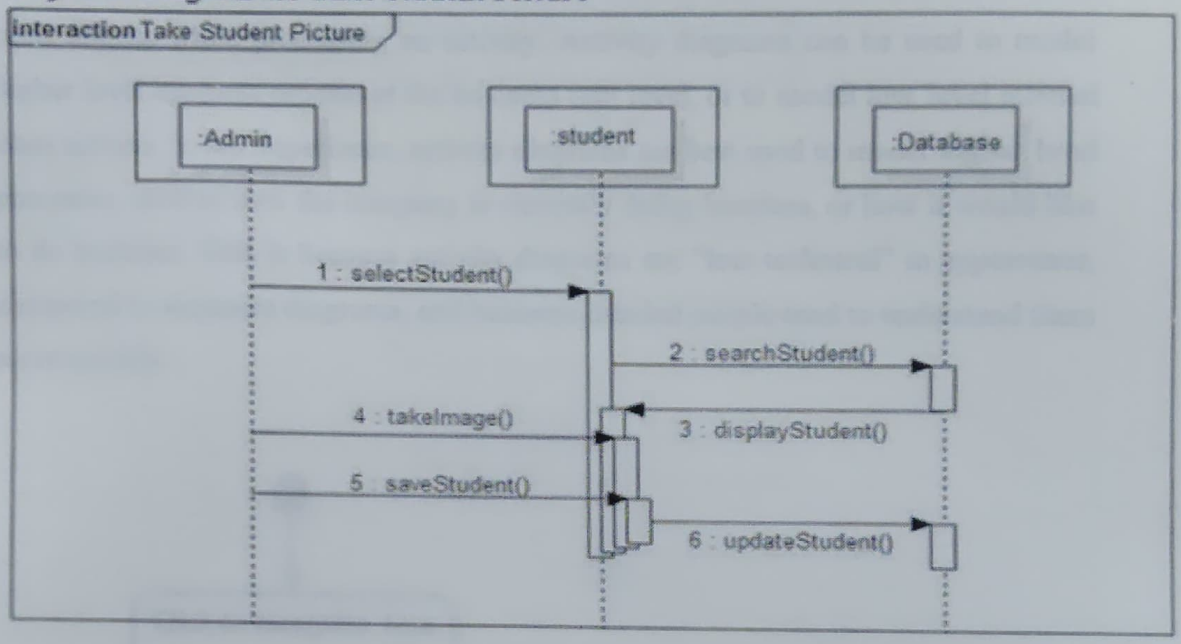


Figure 5.2.2.3: Sequence diagram for taking student picture

5.2.3 Activity Diagram

Activity diagrams show the procedural flow of control between two or more class objects while processing an activity. Activity diagrams can be used to model higher level business process at the business unit level, or to model low level internal class actions. In my experience, activity diagrams are best used to model higher level processes, such as how the company is currently doing business, or how it would like to do business. This is because activity diagrams are "less technical" in appearance, compared to sequence diagrams, and business minded people tend to understand them more quickly.

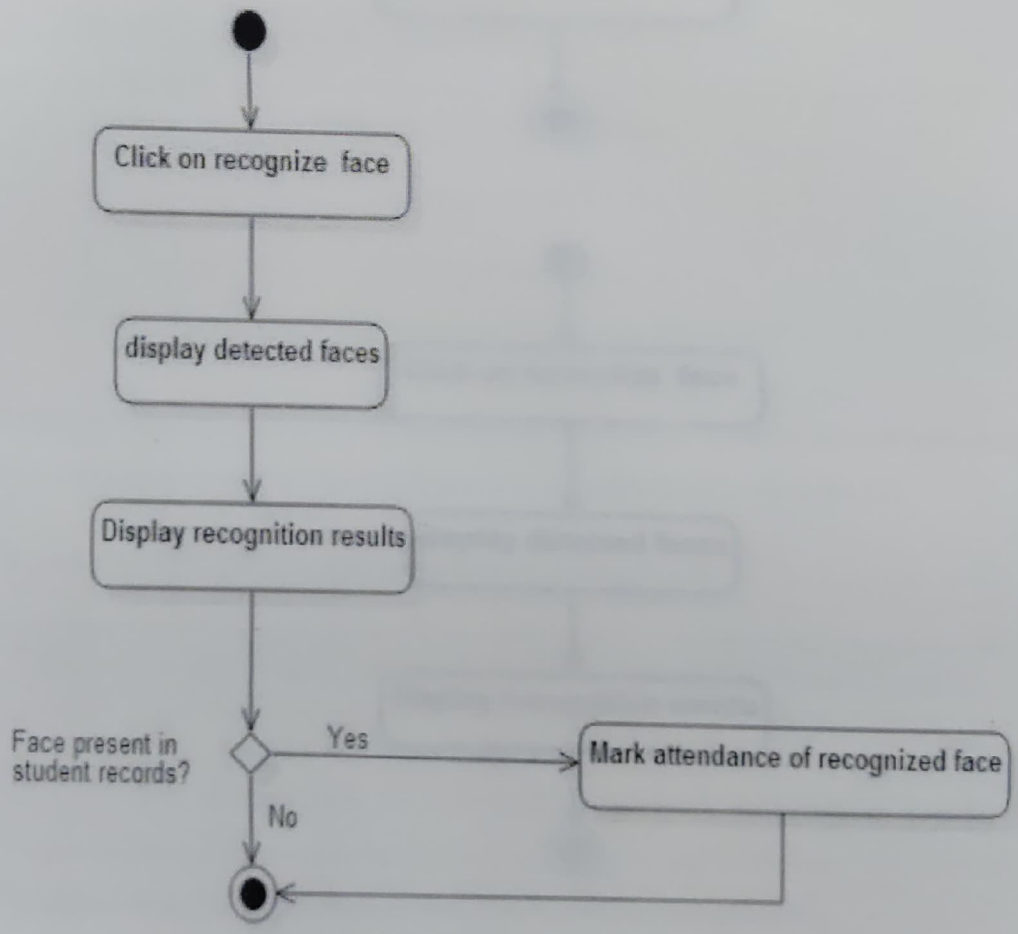


Figure 5.2.3.1: Activity diagram for marking attendance

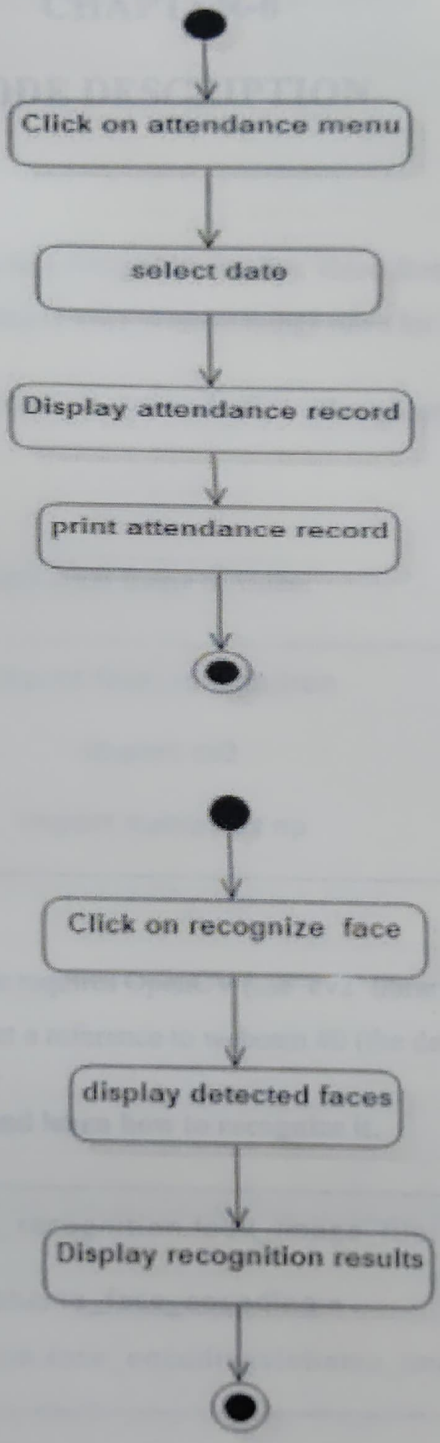


Figure 5.2.3.2: Activity diagram for automated attendance

CHAPTER-6

CODE DESCRIPTION

6.1 Implementation

This is a demo of running face recognition on live video from the webcam which includes some basic performance tweaks to make things run a lot faster:

- Process each video frame at 1/4 resolution (though still display it at full resolution).

- Only detect faces in every other frame of video.

```
import face_recognition

import cv2

import numpy as np
```

PLEASE NOTE: This example requires OpenCV (the `cv2` library) to be installed only to read from your webcam. Get a reference to webcam #0 (the default one).

1. **Load a sample picture and learn how to recognize it.**

```
obama_image = face_recognition.load_image_file("obama.jpg")

obama_face_encoding =
face_recognition.face_encodings(obama_image)[0]
```

2. **Load a second sample picture and learn how to recognize it.**

```
biden_image = face_recognition.load_image_file("biden.jpg")

biden_face_encoding =
face_recognition.face_encodings(biden_image)[0]
```

3. **Create arrays of known face encodings and their names**

```
known_face_encodings = [
obama_face_encoding,
biden_face_encoding]
known_face_names = [
"Barack Obama",
"Joe Biden" ]
```

4. **Initialize some variables**

```
face_locations = []
face_encodings = []
face_names = []
process_this_frame = True
```

5. **Grab a single frame of video**

```
ret, frame = video_capture.read()
```

6. **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)
```

7. **Convert the image from BGR color (which OpenCV uses) to RGB color (which face_recognition uses)**

```
rgb_small_frame = small_frame[:, :, ::-1]
```

8. **Only process every other frame of video to save time**

```
if process_this_frame:
```

9. **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 = []
for face_encoding in face_encodings:
```

10. **See if the face is a match for the known face(s)**

```
matches = face_recognition.compare_faces(known_face_encodings,
face_encoding)
name = "Unknown"
```

11. **If a match was found in known_face_encodings, just use the first one.**

a. if True in matches:

```
first_match_index = matches.index(True)
name = known_face_names[first_match_index]
```

b. Or instead, use the known face with the smallest distance to the new face

```
face_distances =
face_recognition.face_distance(known_face_encodings,
face_encoding)
best_match_index = np.argmin(face_distances)
```

```
if matches[best_match_index]:
name = known_face_names[best_match_index]
face_names.append(name)
process_this_frame = not process_this_frame
```

12. Display the results

```
for (top, right, bottom, left), name in zip(face_locations,
face_names):
```

13. Scale back up face locations since the frame we detected in was scaled to 1/4 size

```
top *= 4
right *= 4
bottom *= 4
left *= 4
```

14 Draw a box around the face

```
cv2.rectangle(frame, (left, top), (right, bottom), (0, 0, 255), 2)
```

15. Draw a label with a name below the face

```
cv2.rectangle(frame, (left, bottom - 35), (right, bottom), (0, 0, 255),
cv2.FILLED)
font = cv2.FONT_HERSHEY_DUPLEX
cv2.putText(frame, name, (left + 6, bottom - 6), font, 1.0, (255, 255,
255), 1)
```

16. Display the resulting image

```
cv2.imshow('Video', frame)
```

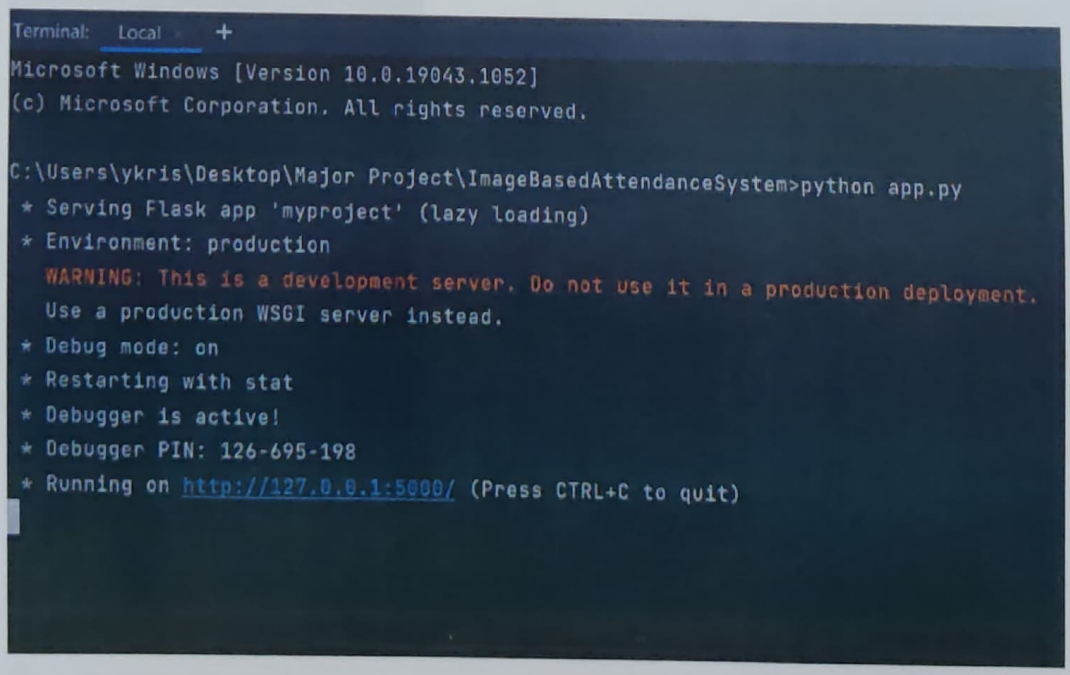
17. Hit 'q' on the keyboard to quit!

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

18. Release handle to the webcam

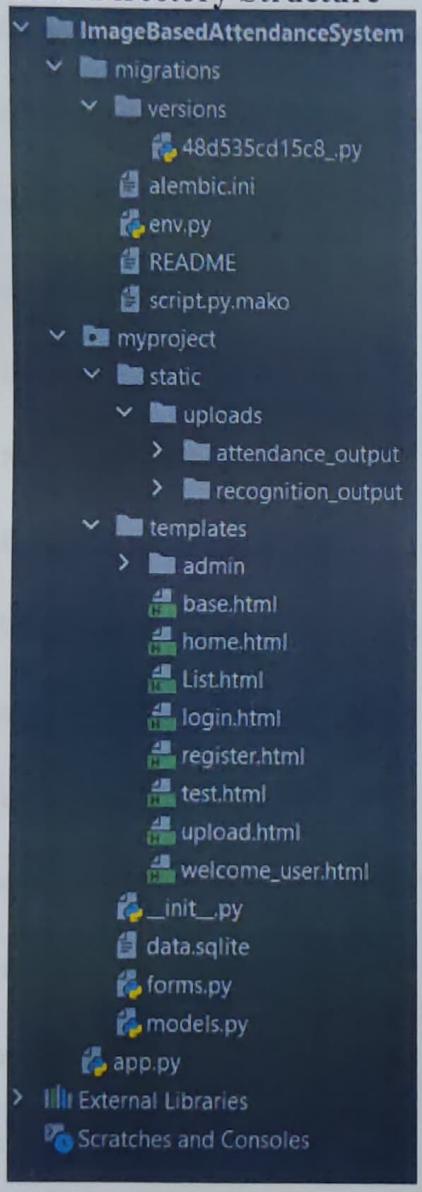
```
video_capture.release()
cv2.destroyAllWindows()
```

6.2 Terminal Screen



6.3 Flask App Building

6.3.1 Directory Structure



6.3.2 app.py

```
import pickle

from datetime import datetime

import face_recognition

from flask_admin import Admin, BaseView

from myproject import app, db
```

```
from flask import render_template, redirect, request, url_for, flash, abort

from flask_login import login_user, login_required, logout_user

from myproject.models import User, Attendance

from myproject.forms import LoginForm, RegistrationForm

import datetime

import openpyxl

from os import path

from flask import Flask

from flask import send_file

UPLOAD_FOLDER = r'myproject\static\uploads'

app1 = Flask(__name__)

app1.secret_key = "secret key"

app1.config['UPLOAD_FOLDER'] = UPLOAD_FOLDER

user_img_encoding = []

subject = ''

current_user = ''

start = ''

end = ''

@app.route('/')

def home():

return render_template('home.html')
```

```
@app.route('/excel', methods=["GET"])
def excel():
if request.method == "GET":
print(request.args.get('period'))
subject_id = request.args.get('period')
end_time = request.args.get('end_time')
print(subject_id + " in excel")
l = end_time.rsplit(':')
end_hour = int(l[0])
end_min = int(l[1])
now = datetime.datetime.now()
today_end = now.replace(hour=end_hour, minute=end_min, second=0,
microsecond=0)
now = datetime.datetime.now()
if now > today_end:
print(subject_id)
print(end_time)
if (path.exists(r"myproject\static\uploads\attendance_output\\" + subject_id + "-
output.xlsx")):
return send_file(r"static\uploads\attendance_output\\" + subject_id + "-output.xlsx",
as_attachment=True)
else:
```

```
if (path.exists(r"myproject\static\uploads\recognition_output\\" + subject_id)):

f = open(r"myproject\static\uploads\recognition_output\\" + subject_id, "r")

print("file opened for reading")

hash_map = {}

while True:

line = f.readline()

if not line:

break

l = line.strip().split(' ')

key = l[0]

val = l[2]

if val == "True":

if key in hash_map.keys():

val = hash_map[key]

val += 1

hash_map[key] = val

else:

hash_map[key] = 1

print(hash_map)

workbook = openpyxl.Workbook()

sheet = workbook.active
```

```
row = 1
for key, values in hash_map.items():
# Put the key in the first column for each key in the dictionary
sheet.cell(row=row, column=1, value=key)
column = 2
sheet.cell(row=row, column=column, value=values)
row += 1
workbook.save(
filename=r"myproject\static\uploads\attendance_output\\" + subject_id + "-
output.xlsx")
return send_file(r"static\uploads\attendance_output\\" + subject_id + "-output.xlsx",
as_attachment=True)
# else:
# print("Data loss")
return '''<html> <head></head><body><h1>Please download after class
ends</h1></body></html>'''
@app.route('/login', methods=['GET', 'POST'])
def login():
form = LoginForm()
if form.validate_on_submit():
# Grab the user from our User Models table
global current_user
```

```
current_user = User.query.filter_by(email=form.email.data).first()

print(User.query.all())

print('username', current_user)

user = current_user

current_user = user.username

global user_img_encoding

s = User.query.get(user.id)

user_img_encoding = pickle.loads(s.image)

# https://stackoverflow.com/questions/2209755/python-operation-vs-is-not

if user.check_password(form.password.data) and user is not None:

# Log in the user

login_user(user)

flash('Logged in successfully.')

# print("as soon as login", user_img_encoding)

# If a user was trying to visit a page that requires a login

# flask saves that URL as 'next'.

next = request.args.get('next')

if next == None or not next[0] == '/':

next = url_for('home')

return redirect(next)

return render_template('login.html', form=form)
```

```
@app.route('/welcome', methods=["GET"])
@login_required
def welcome_user():
if (request.method == "GET"):
now = datetime.datetime.now()
print("now", now)
start = request.args.get('start_time')
end = request.args.get('end_time')
print("start", start)
print("end", end)
if start != "" and end != "":
l = start.rsplit(':')
start_hour = int(l[0])
start_min = int(l[1])
print(l)
l = end.rsplit(':')
end_hour = int(l[0])
end_min = int(l[1])
print(l)
today_start = now.replace(hour=start_hour, minute=start_min, second=0,
microsecond=0)
```

```
today_end = now.replace(hour=end_hour, minute=end_min, second=0,
microsecond=0)

print("today_start", today_start)

print("today_end", today_end)

if today_start <= now < today_end:

print("came here")

return render_template('welcome_user.html')

else:

return '''

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Title</title>

</head>

<body>

<h1>The class has not yet started or it is over</h1>

<a href="/login"> Return</a>

</body>

</html>

'''

return '''

<html lang="en">
```

```
<head>
<meta charset="UTF-8">
<title>Title</title>
</head>
<body>
<h1>Class is not yet scheduled!</h1>
<a href="/login"> Return</a>
</body>
</html>
'''
@app.route('/classes')
@login_required
def todays_classes():
result = db.session.execute(
"select * from Attendance where class_date>=date('now') and
class_date<date('now','+1 days')")
return render_template('List.html', result=result)
@app.route('/logout')
@login_required
def logout():
logout_user()
flash('You logged out!')
```

```
return redirect(url_for('home'))

@app.route('/admin/submit', methods=['POST'])

def admin_side():

if (request.method == 'POST'):

print('time and date')

global subject

subject = str(request.form['subject'])

global start, end

start = request.form['start']

end = request.form['end']

class_date = str(request.form['date'])

print('class_date', class_date)

class_date = datetime.datetime.strptime(class_date, "%Y-%m-%d")

class_date = class_date.date()

x = datetime.datetime.now()

day = str(x.strftime("%d"))

month = str(x.strftime("%m"))

year = str(x.strftime("%Y"))

sub_id = subject + "-" + str(class_date)

record = Attendance(subjectid=sub_id, start_time=str(start), end_time=str(end),
class_date=class_date)

db.session.add(record)
```

```
db.session.commit()
f = open(r'myproject\static\uploads\recognition_output\\' + sub_id, "a")
return MyView().render('/admin/index.html')
@app.route('/register', methods=['GET', 'POST'])
def register():
form = RegistrationForm()
if form.validate_on_submit():
if request.method == 'POST':
filestream = request.files['image']
img = face_recognition.load_image_file(filestream)
img_encodings = face_recognition.face_encodings(img)
pickled_image = pickle.dumps(img_encodings)
print(pickle.loads(pickled_image))
user = User(email=form.email.data,
username=form.username.data,
password=form.password.data,
image=pickled_image)
db.session.add(user)
db.session.commit()
flash('Thanks for registering! Now you can login!')
return redirect(url_for('login'))
```

```
return render_template('register.html', form=form)

# @app.route('/foo')

# @login_required

# def foo():

# return render_template('upload.html')

class MyView(BaseView):

def __init__(self, *args, **kwargs):

self._default_view = True

super(MyView, self).__init__(*args, **kwargs)

self.admin = Admin()

@app.route('/handle', methods=['GET', 'POST'])

def handle():

image = None

if request.method == 'POST':

print('hello')

file = request.files['webcam'] # image is of type FileStorage

x = datetime.datetime.now()

day = str(x.strftime("%d"))

month = str(x.strftime("%m"))

year = str(x.strftime("%Y"))

global subject, current_user
```

```
print(subject)

if (subject != ''):

f = open(r'myproject\static\uploads\recognition_output\\' + subject + "-" + year + "-" +
month + "-" + day,

"a")

img = face_recognition.load_image_file(file)

snapshot_encodings = face_recognition.face_encodings(img)

print("current", snapshot_encodings)

print("db", user_img_encoding)

results = face_recognition.compare_faces(user_img_encoding,
snapshot_encodings[0])

f.writelines([current_user + " " + " %s\n" % item for item in results])

f.close()

print(subject)

print("user_img_encoding", len(user_img_encoding))

print(" snapshot_encodings", len(snapshot_encodings))

print("results:", results)

else:

flash('Admin has not yet started the class please wait')

return '''<html><head></head><body><h1>Admin has not yet started the class please
wait!</h1></body></html>'''

# file.save(os.path.join(app1.config['UPLOAD_FOLDER'],filename))

# filename,user_img_encoding
```

```
return '<h1>Hello</h1>'

if __name__ == '__main__':

app.run(debug=True)
```

6.3.3 HTML code

```
<!DOCTYPE html>

<html>

<head>

<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"
integrity="sha384-
Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6J
Xm" crossorigin="anonymous">

<script src="https://code.jquery.com/jquery-3.2.1.slim.min.js" integrity="sha384-
KJ3o2DKtIkvYIK3UENzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG
5KkN" crossorigin="anonymous"></script>

<script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js"
integrity="sha384-
ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q
" crossorigin="anonymous"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"
integrity="sha384-
JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVCmYl
" crossorigin="anonymous"></script>

<meta charset="utf-8">

<title></title>
```

```
</head>

<body>

<ul class="nav">

<li class="nav-item">

<a class="nav-link" href="{{ url_for('home') }}">Home</a>

<li class="nav-link"><a href="{{ url_for('todays_classes') }}">Today's
classes</a></li>

</li>

{% if current_user.is_authenticated %}

<li class="nav-link"><a href="{{ url_for('logout') }}">Log Out</a></li>

{% else %}

<li class="nav-link"><a href="{{ url_for('login') }}">Log In</a></li>

<li class="nav-link"><a href="{{ url_for('register') }}">Register</a></li>

{% endif %}

</ul>

{% block content %}

{% endblock %}

</body>

</html>
```

CHAPTER-7

TESTING

The main use of testing is to find out errors. Testing is the way toward attempting to find each possible flaw or shortcoming in a work item. It gives a way to deal with check the helpfulness of parts, sub-assemblies, social occasions just as a finished thing It is the path toward working on programming with the point of ensuring that the Software system satisfies its necessities and customer wants and does not b omb in an unacceptable manner. There are various sorts of test. Each test type keeps an eye on a specific testing need. Testing permits to expel the mistakes and improve the framework execution. There are numerous kinds of tests which enables us to improve our venture execution and to make it mistake free. What's more we likewise have tests which encourage us to check singular modules autonomously and furthermore to check complete framework together according to our convenience.

7.1 TYPES OF TESTS

7.1.1 Unit testing

Unit testing incorporates the arrangement of analyses that favour that within program basis is working properly, and that program information sources produce significant yields. It checks whether little segments are working appropriately or not. Every single decision branch and inside code stream should be endorsed. It is the attempting of individual programming units of the application .it is done after the completion of an individual unit before fuse. This is an auxiliary attempting, that relies upon learning of its improvement and is prominent. Unit tests perform fundamental tests at section level and test a specific business system, application, or possibly structure plan. Unit tests ensure that all of a thoughtful method for a business technique performs unequivocally to the recorded points of interest and contains obviously portrayed data sources and foreseen results.

A unit test encourages you to discover which part is broken in your application and fixes it quicker.

7.1.2 Integration testing

Integration tests are expected to test joined programming modules to choose whether they everything considered continue running as one program. Testing is an event driven and is dynamically stressed over the crucial after effect of screens or fields. Combination tests show that in spite of the way that the sections were autonomously satisfied, as showed up by successfully unit testing, the gathering of portions are correct and unsurprising. Combination testing is expressly away for revealing the issues that rise up out of the gathering of these portions.

Integration testing permits to discover blunders because of unexpected communication between the framework and the sub-framework segments. We test the product in order to test and to identify all the potential mistakes in our undertaking once we complete the source code and before conveying it to the clients.

The techniques for performing tests. These techniques provide guidance for testing:

- To test the internal logic of the software components.
- To test the input and output domains of a programs and to uncover the errors in program function, behavior and performance.

We can test the software by using two methods:

- White Box testing: In this the internal logic program is being checked by using different test case design techniques.
- Black Box testing: In this the software requirements are tested by using different test case design techniques.

Both the above-mentioned techniques help in finding out the maximum number of errors with minimal time and effort.

7.1.3 Performance Testing

This test is done to find the run-time performance of the software with the context of the integrated system. These tests can be carried out throughout the testing process. For example, the performance of individual module is accessed during white box testing under unit testing.

7.1.4 Verification and Validation

Testing procedure is a piece of subject alluding to checking and approval of our task. We have to find the framework determinations and we should attempt to meet the details of the client and to fulfil the client, for this reason, we need to check and approve the item and we have to ensure that everything is working appropriately. Check and approval are the two unique things. One is performed to guarantee that the product is working accurately and to implement a particular usefulness and the other is done to guarantee if the client prerequisites are appropriately met or not by the finished result.

Check is progressively similar to 'would we say we are building the item, right?' and approval is increasingly similar to 'would we say we are building the correct item?'.

CHAPTER-8

RESULT AND OUTPUT SCREENS

8.1 Results

The system is developed using Python Programming language for deploying an easy and a secure way of taking down attendance. The software first captures an image of all the authorized persons and stores the information into database. The system then stores the image by mapping it into a face coordinate structure. Next time whenever the registered person enters the online class the system recognizes the person and marks his attendance along with the time. The system is convenient and secure for the users and also saves time and efforts.

8.2 Screens

8.2.1 User Portal

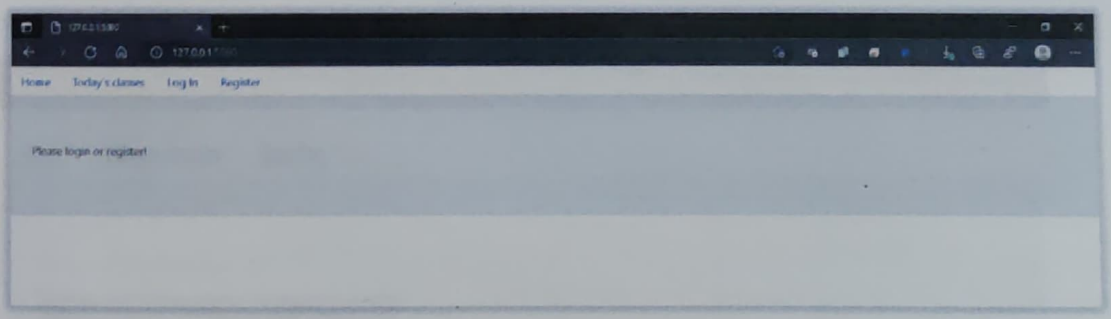


Figure 8.2.1.1: Home Page

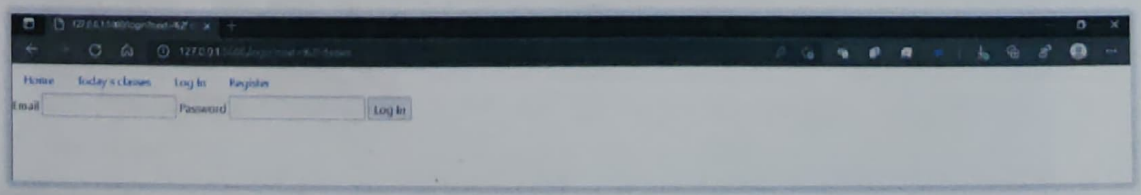


Figure 8.2.1.2: Login For Students

Figure 8.2.1.3: Registration for Students

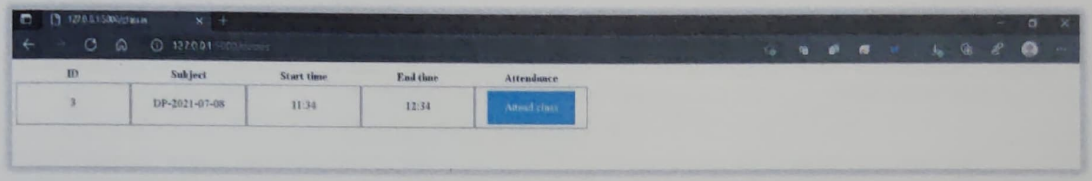


Figure 8.2.1.4: Classes to be attended by the students

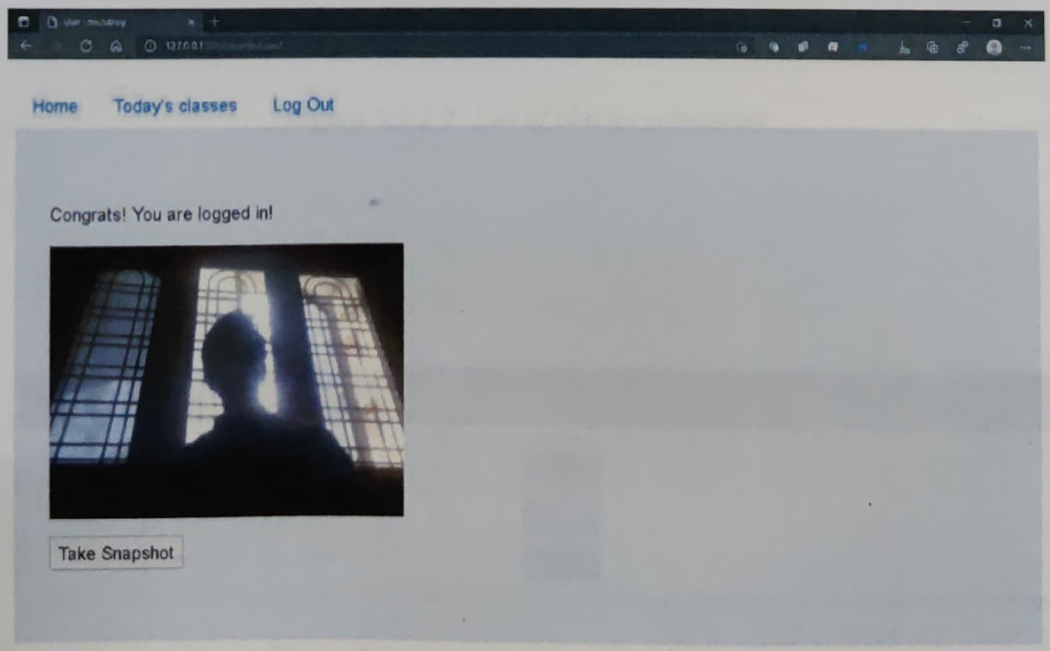


Figure 8.2.1.5: Attending Classes

8.2.2 Admin Portal

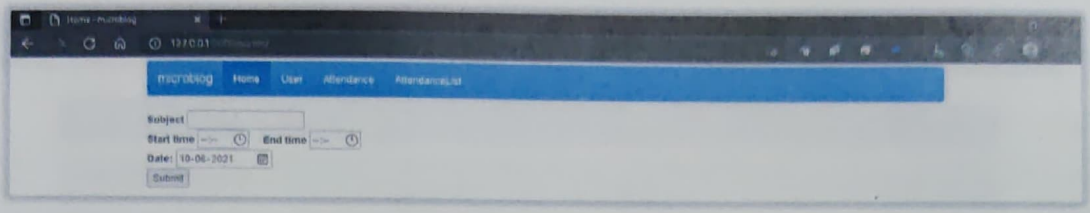


Figure 8.2.2.1: Classes to be scheduled by the admin

Figure 8.2.2.2: List of classes scheduled

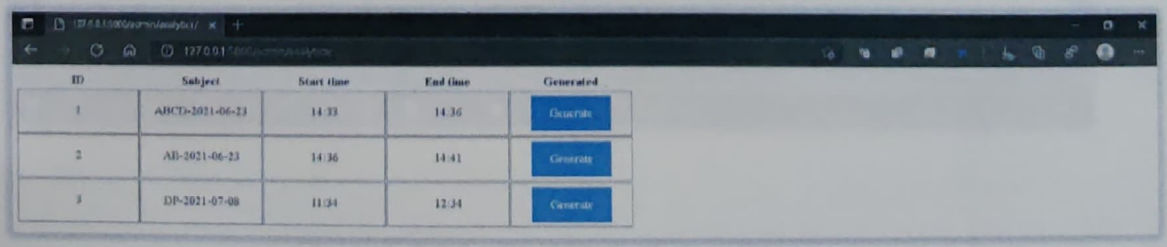


Figure 8.2.2.3: Attendance File Generation

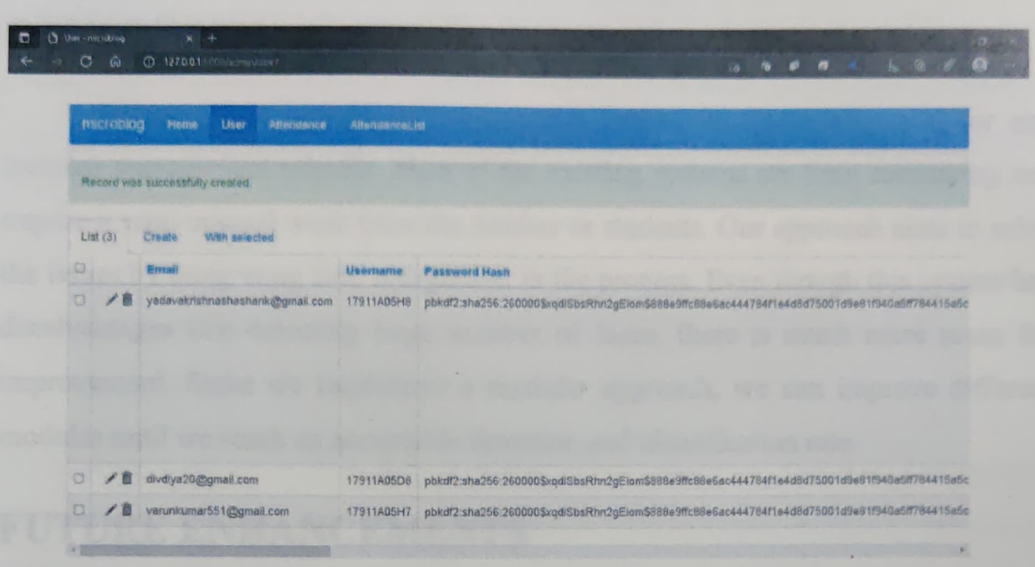


Figure 8.2.2.4: Viewing students in the database

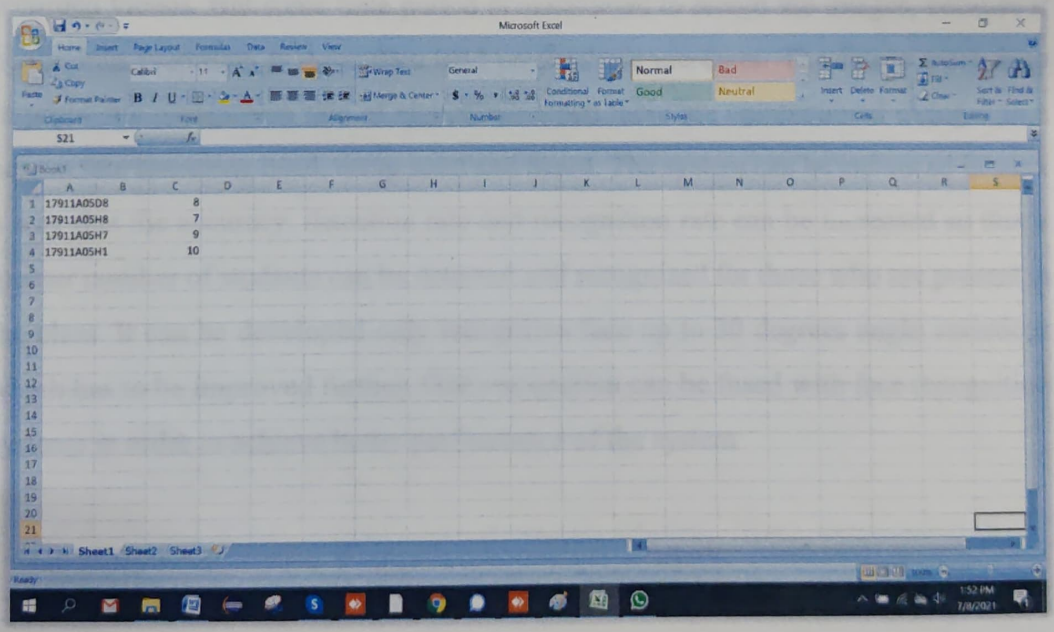


Figure 8.2.2.5: Attendance Output File

CHAPTER-9

CONCLUSION

An automatic attendance management system is a necessary tool for any learning management schools. Most of the existing systems are time consuming and require a semi manual work from the teacher or students. Our approach aims to solve the issues by integrating face recognition in the process. Even though this system has disadvantages like detecting large number of faces, there is much more room for improvement. Since we implement a modular approach, we can improve different modules until we reach an acceptable detection and identification rate.

FUTURE ENHANCEMENTS

Further extensions can be made, to achieve the real time detection of specific student in the surveillance premises. Instead of taking images, we can also work with recorded videos. But some time period is maintained to record the images, because if continuous recording is done then load on database increases. The future work is to improve the recognition rate of algorithms when there are unintentional changes in a person like tonsuring head, using scarf and beard. The system can be enhanced in such a way that the accuracy, detection rate and recognition rate can be increased so that a greater number of students can be detected and recognized for those who are present in the class. It can be developed only recognizes face up to 30 degrees angle variations which has to be improved further. Gait recognition can be fused with face recognition systems in order to achieve better performance of the system

REFERENCES

[1] N.Sudhakar Reddy, M.V.Sumanth, S.Suresh Babu, "A Counterpart Approach to Attendance and Feedback System using Machine Learning Techniques",Journal of Emerging Technologies and Innovative Research (JETIR), Volume 5, Issue 12, Dec 2018.

[2] Dan Wang, Rong Fu, Zuying Luo, "Classroom Attendance Auto-management Based on Deep Learning",Advances in Social Science, Education and Humanities Research, volume 123,ICESAME 2017.

[3] Akshara Jadhav, Akshay Jadhav, Tushar Ladhe, Krishna Yeolekar, "Automated Attendance System Using Face Recognition", International Research Journal of Engineering and Technology (IRJET), Volume 4, Issue 1, Jan 2017.

[4] B Prabhavathi, V Tanuja, V Madhu Viswanatham and M Rajashekhara Babu, "A smart technique for attendance system to recognize faces through parallelism", IOP Conf. Series: Materials Science and Engineering 263, 2017.

[5] Prajakta Lad, Sonali More, Simran Parkhe, Priyanka Nikam, Dipalee Chaudhari, " Student Attendance System Using Iris Detection", IJARIIE-ISSN(O)-2395-4396, Vol-3 Issue-2 2017.

[6] Samuel Lukas, Aditya Rama Mitra, Ririn Ikana Desanti, Dion Krisnadi, "Student Attendance System in Classroom Using Face Recognition Technique", Conference Paper DOI: 10.1109/ICTC.2016.7763360, Oct 2016.

[7] K.Senthamil Selvi, P.Chitrakala, A.Antony Jenitha, "Face Recognition Based Attendance Marking System", IJCSMC, Vol. 3, Issue. 2, February 2014.

[8] Yohei KAWAGUCHI, Tetsuo SHOJI, Weijane LIN, Koh KAKUSHO, Michihiko MINOH, "Face Recognition-based Lecture Attendance System", Oct 2014.

[9] Shireesha Chintalapati, M.V. Raghunadh, "Automated Attendance Management System Based On Face Recognition Algorithms", IEEE International Conference on Computational Intelligence and Computing Research, 2013.

[10] B. K. Mohamed and C. Raghu, "Fingerprint attendance system for classroom needs," India Conference (INDICON), Annual IEEE, pp. 433–438, 2012.

[11] L. Zhi-fang, Y. Zhi-sheng, A.K.Jain and W. Yun-qiong, 2003, "Face Detection And Facial Feature Extraction In Color Image", Proc. The Fifth International Conference on Computational Intelligence and Multimedia Applications (ICCIMA'03), pp.126-130, Xi'an,China.

[12] S. Kherchaoui and A. Houacine, 2010, "Face Detection Based On A Model Of The Skin Color With Constraints And Template Matching", Proc. 2010 International Conference onMachine and Web Intelligence, pp. 469 - 472, Algiers, Algeria.

[13] Hajar Filali Jamal Riffi Adnane Mohamed Mahraz Hamid Tairi, Multiple face detection based on machine learning, 978-1-5386-4396-9/18/$31.00 c 2018 IEEE.

[14] E.Varadharajan, R.Dharani, S.Jeevitha, B.Kavinmathi, S.Hemalatha,Automatic attendance management system using face detection, Online International Conference on Green Engineering and Technologies (IC-GET), 978-1-5090-4556-3/16/$31.00 ©2016 IEEE.

[15] Shireesha Chintalapati, M.V. Raghunadh, Automated Attendance Management System Based on Face Recognition Algorithms, 2013 IEEE International