SMART ATTENDANCE USING FACIAL RECOGNITION

M. LUBNA YASMEEN, ASSISTANT PROFESSOR
MALGIREDDY HRISHITH RAJ REDDY, BIEREDDY MRIDGALA MALIGIREDDY PRANAV REDDY, PEGADA CHANDANA

Sreyas Institutue of Engineering and Technology.

1. INTRODUCTION

1.1 MOTIVATION

Face Recognition Attendance Systems are becoming increasingly popular as they are the most feasible option currently available for organizations to make employee attendance contactless. Other types of biometric identification systems, such as fingerprints, capture an individual's identity through touch, while facial recognition is a touchless method of identifying employees and visitors. Touchless systems are an effective preventive measure in the current scenario as they facilitate the safe and efficient movement of people in and out. A face recognition attendance system identifies and verifies a person automatically and marks attendance based on their facial recognition.
Small and large businesses alike are taking note of the face recognition attendance system. It is no surprise that such systems are increasingly used in offices and business complexes due to a plethora of benefits, both for employers and employees.

1.2 PROBLEM STATEMENT

Traditional student attendance marking technique is often facing a lot of trouble. The face recognition student attendance system emphasizes its simplicity by eliminating classical student attendance marking technique such as calling student names or checking respective identification cards. There are not only disturbing the teaching process but also causes distraction for students during exam sessions. Apart from calling names, attendance sheet is passed around the classroom during the lecture sessions. The lecture class especially the class with a large number of students might find it difficult to have the attendance sheet being passed around the class. Thus, face recognition student attendance system is proposed in order to replace the manual signing of the presence of students which are burdensome and causes students get distracted in order to sign for their attendance.

Furthermore, the face recognition based automated student attendance system able to overcome the problem of fraudulent approach and lecturers does not have to count the number of students several times to ensure the presence of the students.
The paper proposed by Zhao, W et al. (2003) has listed the difficulties of facial identification. One of the difficulties of facial identification is the identification between known and unknown images. In addition, paper proposed by Pooja G.R et al. (2010) found out that the training process for face recognition student attendance system is slow and time-consuming. In addition, the paper proposed by Priyanka Wagh et al. (2015) mentioned that different lighting and head poses are often the problems that could degrade the performance of face recognition based student attendance system.

Hence, there is a need to develop a real time operating student attendance system which means the identification process must be done within defined time constraints to prevent omission. The extracted features from facial images which represent the identity of the students have to be consistent towards a change in background, illumination, pose and expression. High accuracy and fast computation time will be the evaluation points of the performance.

1.3 PROJECT OBJECTIVE

The objective of this project is to develop face recognition based automated student attendance system. Expected achievements in order to fulfill the objectives are:

- To detect the face segment from the video frame.
- To extract the useful features from the face detected.
- To classify the features in order to recognize the face detected.
- To record the attendance of the identified student.

Figure 1.1 Block Diagram

2. LITERATURE SURVEY

2.1 STUDENT ATTENDANCE SYSTEM

Arun Katara et al. (2017) mentioned disadvantages of RFID (Radio Frequency Identification) card system, fingerprint system and iris recognition system. RFID card system is implemented due to its simplicity. However, the user tends to help their friends to check in as long as they have their friend's ID card. The fingerprint system is indeed effective but not efficient because it takes time for the verification process so the user has to line up and perform the verification one by one. However for face recognition, the human face is always exposed and contain less information compared to iris. Iris recognition system which contains more detail might invade the privacy of the user. Voice recognition is available, but it is less accurate compared to other methods. Hence, face recognition system is suggested to be implemented in the student attendance system.

Table 2.1.1 Advantages & Disadvantages of Different Biometric System (Arun Katara et al., 2017)

System type	**Advantages**	**Disadvantages**
RFID card system	Simple	Fraudulent usage
Fingerprint system	Accurate	Time-consuming
Voice recognition system	-	Less accurate compared to others
Iris recognition system	Accurate	Privacy Invasion

2.2 FACE DETECTION

Difference between face detection and face recognition are often misunderstood. Face detection is to determine only the face segment or face region from image, whereas face recognition is to identify the owner of the facial image. S.Aanjanadevi et al. (2017) and Wei-Lun Chao (2007) presented a few factors which cause face detection and face recognition to encounter difficulties. These factors consist of background, illumination, pose, expression, occlusion, rotation, scaling and translation. The definition of each factor is tabulated in Table 2.2.1

Table 2.2.1 Factors Causing Face Detection Difficulties (S.Aanjanadevi et al., 2017)

Background	Variation of background and environment around people in the image which affect the efficiency of face recognition.
Illumination	Illumination is the variation caused by various lighting environments which degrade the facial feature detection.
Pose	Pose variation means different angle of the acquired the facial image which cause distortion to recognition process, especially for Eigen face and Fisher face recognition method.
Expression	Different facial expressions are used to express feelings and emotions. The expression variation causes spatial relation change and the facial-feature shape change.
Occlusion	Occlusion means part of the human face is unobserved. This will diminish the performance of face recognition algorithms due to deficiency information.
Rotation, scaling and translation	Transformation of images which might cause distortion of the original information about the images.

There are a few face detection methods that the previous researchers have worked on.

However, most of them used frontal upright facial images which consist of only one face. The face region is fully exposed without obstacles and free from the spectacles.

Akshara Jadhav et al. (2017) and by P. Arun Mozhi Devan et al. (2017) suggested Viola-Jones algorithm for face detection for student attendance system. They concluded that out of methods such as face geometry- based methods, Feature Invariant methods and Machine learning based methods, Viola-Jones algorithm is not only fast and robust, but gives high detection rate and perform better in different lighting condition.

Rahul V. Patil and S. B. Bangar (2017) also agreed that Viola-Jones algorithm gives better performance in different lighting condition. In addition, in the paper by Mrunmayee Shirodkar et al. (2015), they mentioned that Viola-Jones algorithm is able to eliminate the issues of illumination as well as scaling and rotation. In addition, Naveed Khan Balcoh (2012) proposed that Viola-Jones algorithm is the most efficient among all algorithms for instance the AdaBoost algorithm, the FloatBoost algorithm, Neural Networks, the S-AdaBoost algorithm, Support Vector Machines (SVM) and the Bayes classifier.

Varsha Gupta and Dipesh Sharma (2014) studied Local Binary Pattern (LBP), Adaboost algorithm, local successive mean quantization transform (SMQT) Features, sparse network of winnows (SNOW) Classifier Method and Neural Network-based face detection methods in addition to Viola-Jones algorithm. They concluded that Viola-Jones algorithm has the highest speed and highest accuracy among all the methods. Other methods for instance Local Binary Pattern and SMQT Features have simple computation and able to deal with illumination problem, their overall performance is weaker than Viola-Jones algorithm for face detection. The advantages and disadvantages of the methods is studied and tabulated in Table 2.2.2.

Table 2.2.2 Advantages & Disadvantages of Face Detection Methods (Varsha Gupta and Dipesh Sharma, 2014)

Face detection method	Advantages	Disadvantages
Viola jones algorithm	High detection speed High accuracy.	Long training time. Limited head pose. Not able to detect dark faces.
Local Binary pattern	Simple computation. High tolerance against the monotonic illumination changes.	Only used for binary and grey images. Overall performance is inaccurate compared to Viola-Jones algorithm.
AdaBoost algorithm (part of Viola jones algorithm)	Need not to have any prior knowledge about face structure.	The result highly depends on the training data and affected by weak classifiers.
SMQT Features and SNOW Classifier Method	Capable to deal with lighting problem in object detection. Efficient in computation.	The region contain very similar to grey value regions will be misidentified as face.
Neural-Network	High accuracy only if large size of image were trained.	Detection process is slow and computation is complex. Overall performance is weaker than Viola-Jones algorithm.

2.2.1 Viola-Jones Algorithm

Viola-Jones algorithm which was introduced by P. Viola, M. J. Jones (2001) is the most popular algorithm to localize the face segment from static images or video frame. Basically the concept of Viola-Jones algorithm consists of four parts. The first part is known as Haar feature, second part is where integral image is created, followed by implementation of Adaboost on the third part and lastly cascading process.

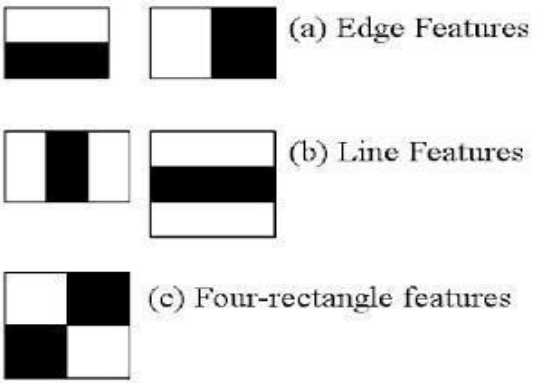


Figure 2.2.1.1 Haar Feature (Docs.opencv.org, 2018)

Viola-Jones algorithm analyses a given image using Haar features consisting of multiple rectangles (Mekha Joseph et al., 2016). Figure 2.1 shows several types of Haar features. The features perform as window function mapping onto the image. A single value result, which representing each feature can be computed by subtracting the sum of the white rectangle(s) from the sum of the black rectangle(s) (Mekha Joseph et al., 2016). The illustration is shown in Figure 2.2.2.2

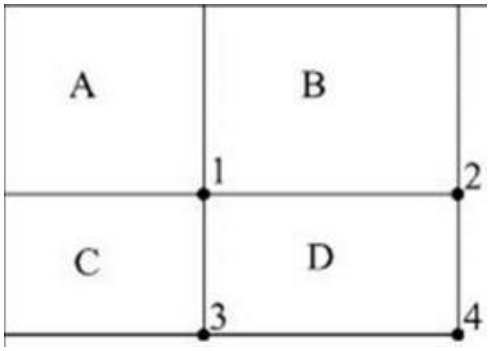


Figure 2.2.2.2 Integral of Image (Srushti Girhe et al., 2015)

The value of integrating image in a specific location is the sum of pixels on the left and the top of the respective location. In order to illustrate clearly, the value of the integral image at location 1 is the sum of the pixels in rectangle A. The values of integral image at the rest of the locations are cumulative. For instance, the value at location 2 is summation of A and B, (A + B), at location 3 is summation of A and C, (A + C), and at location 4 is summation of all the

regions, (A + B + C + D) (Srushti Girhe et al., 2015). Therefore, the sum within the D region can be computed with only addition and subtraction of diagonal at location 4 + 1 − (2 + 3) to eliminate rectangles A, B and C.

Burak Ozen (2017) and Chris McCormick (2013), they have mentioned that Adaboost which is also known as 'Adaptive Boosting' is a famous boosting technique in which multiple "weak classifiers" are combined into a single "strong classifier". The training set is selected for each new classifier according to the results of the previous classifier and determines how much weight should be given to each classifier in order to make it significant.

However, false detection may occur and it was required to remove manually based on human vision. Figure 2.2.2.3 shows an example of false face detection (circle with red).

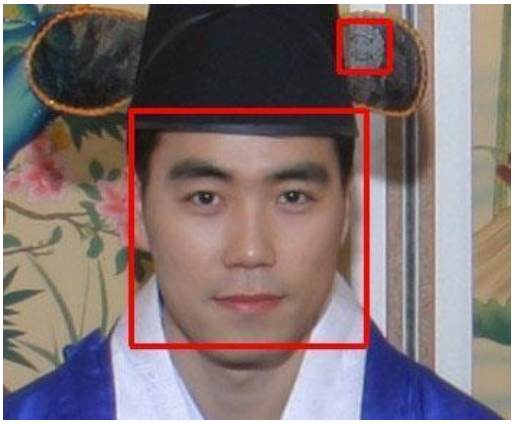


Figure 2.2.2.3 False Face Detection (Kihwan Kim, 2011)

2.3 PRE-PROCESSING

Subhi Singh et al. (2015) suggested cropping of detected face and colour image was converted to grayscale for pre-processing. They also proposed affine transform to be applied to align the facial image based on coordinates in middle of the eyes and scaling of image to be performed. Arun Katara et al (2017), Akshara Jadhav et.al (2017), Shireesha Chintalapati, and M.V. Raghunadh (2013), all of the 3 papers have proposed histogram equalization to be applied to facial image, and scaling of images was performed for pre-processing.

Pre-processing enhances the performance of the system. It plays an essential role to improve the accuracy of face recognition. Scaling is one of the important pre- processing steps to manipulate the size of the image. Scaling down of an image increases the processing speed by reducing the system computations since the number of pixels are reduced. The size and pixels of the image carry spatial information. Gonzalez, R. C. and Woods (2008) mentioned spatial information is a measure of the smallest discernible detail in an image. Hence, spatial information has to be manipulated carefully to avoid distortion of images to prevent checkerboard effect. The size should be same for all the images for normalization and standardization purposes. Subhi Singh et al (2015) proposed PCA (Principal Component Analysis) to extract features from facial images, same length and width of image is preferred, thus images were scaled to 120 × 120 pixels.

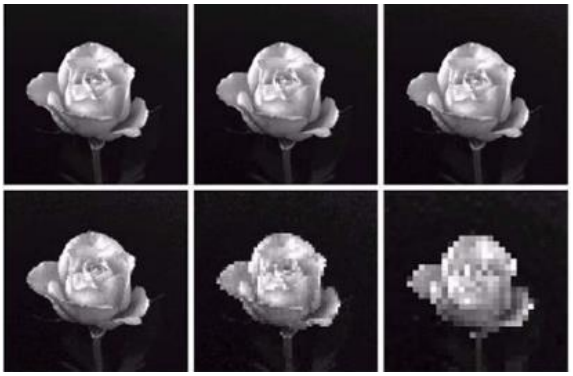


Figure 2.3.1 Images Show Checkerboard Effect Significantly Increasing from Left to Right (Gonzalez, R. C., & Woods, 2008)

Besides scaling of images, colour image is usually converted to grayscale image for pre-processing. Grayscale images are believed to be less sensitive to illumination condition and take less computational time. Grayscale image is 8 bit image which the pixel range from 0 to 255 whereas colour image is 24 bit image which pixel can have 16 77 7216 values. Hence, colour image requires more storage space and more computational power compared to grayscale images. (Kanan and Cottrell, 2012). If colour image is not necessary in computation, then it is considered as noise. In addition, pre-processing is important to enhance the contrast of images. In the paper of Pratiksha M. Patel (2016), he mentioned that Histogram equalization is one of the methods of pre-processing in order to improve the contrast of the image. It provides uniform distribution of intensities over the intensity level axis, which is able to reduce uneven illumination effect at the same time.

Figure 2.3.2 Facial Images Were Converted To Grayscale, Histogram Equalization Was Applied and Images Were Resized to 100x100 (Shireesha Chintalapati and M.V. Raghunadh, 2013)

There are a few methods to improve the contrast of images other than Histogram Equalization. Neethu M. Sasi and V. K. Jayasree (2013) studied Histogram Equalization and Contrast Limited Adaptive Histogram Equalization (CLAHE) in order to enhance myocardial perfusion images. Aliaa A. A. Youssif (2006) studied contrast enhancement together with illumination equalization methods to segment retinal vasculature. In addition, in paper by A., I. and E.Z., F. (2016) Image Contrast Enhancement Techniques and performance were studied. Unlike Histogram equalization, which operate on the data of the entire image, CLAHE operates on data of small regions throughout the image. Hence, the Contrast Limited Adaptive Histogram

Equalization is believed to outperform the conventional Histogram Equalization. Summary of the literature review for contrast improvement is tabulated in Table 2.3.3

Table 2.3.3 Summary of Contrast Improvement

Method	**Concept**	**Advantages**	**Disadvantages**
Histogram equalization	Contrast enhancement is performed by transforming the intensity values, resulting in uniformly distributed histogram.	ess sensitive to noise.	It depends on the global statistics of an image. It cause over enhancement for some part, while peripheral region need more enhancement.
Contrast Limited Adaptive Histogram Equalization (CLAHE)	Unlike, HE which works on entire image, it works on small data regions. Each tile's contrast is enhanced to ensure uniformly distributed histogram. Bilinear interpolation is then used to merge the neighbouring tiles.	prevent over enhancement as well as noise amplification.	More sensitive to noise compared to histogram equalization.

2.4 FEATURE EXTRACTION

The feature is a set of data that represents the information in an image. Extraction of facial feature is most essential for face recognition. However, selection of features could be an arduous task. Feature extraction algorithm has to be consistent and stable over a variety of changes in order to give high accuracy result.

There are a few feature extraction methods for face recognition. In the paper of Bhuvaneshwari et al. (2017), Abhishek Singh and Saurabh Kumar (2012) and Liton Chandra Paul and Abdulla Al Sumam (2012), they proposed PCA for the face recognition. D. Nithya (2015) also used PCA in face recognition based student attendance system. PCA is famous with its robust and high speed computation. Basically, PCA retains data variation and remove unnecessary existing correlations among the original features. PCA is basically a dimension reduction algorithm. It compresses each facial image which is represented by the matrix into single column vector. Furthermore, PCA removes average value from image to centralize the image data. The Principle Component of distribution of facial images is known as Eigen faces. Every single facial image from training set contributes to Eigen faces.

As a result, Eigen face encodes best variation among known facial images. Training images and test images are then projected onto Eigen face space to obtain projected training images and projected test image respectively. Euclidean distance is computed by comparing the distance between projected training images and projected test image to perform the recognition. PCA feature extraction process includes all trained facial images. Hence, the extracted feature contains correlation between facial images in the training set and the result of recognition of PCA highly depends on training set image.

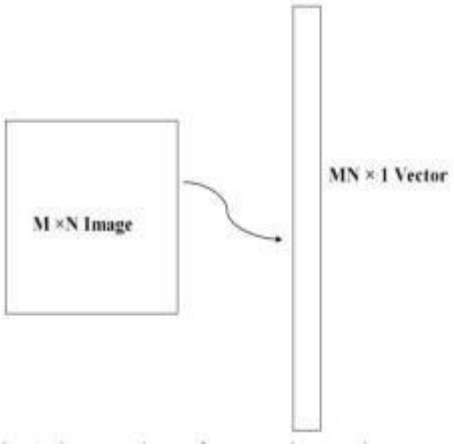


Figure 2.4.1 PCA Dimension Reduction (Liton Chandra Paul and Abdulla Al Sumam, 2012)

LDA (Linear discriminant analysis) also known as Fisher face is another popular algorithm for face recognition. In the paper by Suman Kumar Bhattacharyya and Kumar Rahul (2013), LDA was proposed for face recognition. LDA extract features by grouping images of the same class and separate images of different classes. LDA is able to perform well even with different facial expressions, illumination and pose due to its class separation characteristic. Same class is defined by facial images of the same individual, but with different facial expressions, varying

lighting or pose, whereas facial images of person with different identity are categorized as different classes. Same class images yield within-class scatter matrix meanwhile different class images yield between-class scatter matrix. LDA manage to maximize the ratio of the determinant of the between-class scatter matrix over the determinant of the within class scatter matrix. LDA is believed to have lower error rates compared to PCA only if more samples per class are trained and small size of different class.

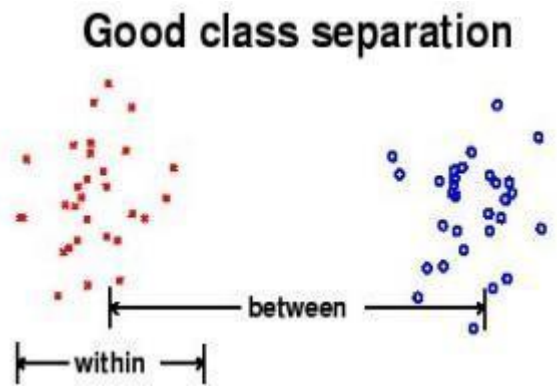


Figure 2.4.2 Class Separation in LDA (Suman Kumar Bhattacharyya and Kumar Rahul)

The original LBP (Local Binary Patterns) operator was introduced by the paper of Timo Ojala et al. (2002). In the paper by Md. Abdur Rahim et al. (2013), they proposed LBP to extract both texture details and contour to represent facial images. LBP divides each facial image into smaller regions and histogram of each region is extracted. The histograms of every region are concatenated into a single feature vector. This feature vector is the representation of the facial image and Chi square statistic is used to measure similarities between facial images. The smallest window size of each region is 3 by 3. It is computed by thresholding each pixel in a window where middle pixel is the threshold value. The neighborhood larger than threshold value is assigned to 1 whereas the neighborhood lower than threshold value is assigned to 0. Then the resulting binary pixels will form a byte value representing center pixel.

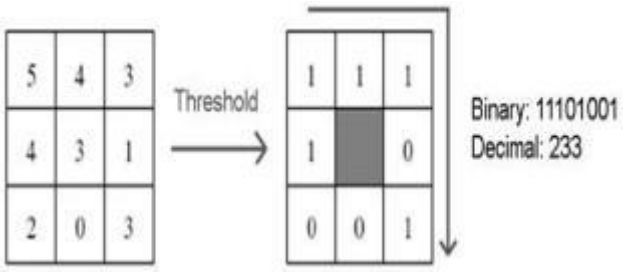


Figure 2.4.3 LBP Operator (Md. Abdur Rahim et.al, 2013)LBP has a few advantages which make it popular to be implemented. It has high tolerance against the monotonic illumination changes and it is able to deal with variety of facial expressions, image rotation and aging of persons. These overwhelming characteristics cause LBP to be prevalent in real-time applications.

Neural network is initially used only in face detection. It is then further studied to be implemented in face recognition. In the paper by Manisha M. Kasar et al. (2016), Artificial

Neural Network (ANN) was studied for face recognition. ANN consists of the network of artificial neurons known as "nodes". The nodes act as human brain in order to make recognition and classification. These nodes are interconnected and values are assigned to determine the strength of their connections. High value indicates strong connection. Neurons were categorized into three types of nodes or layers which are input nodes, hidden nodes, and output nodes. Input nodes are given weight based on its impact. Hidden nodes consist of some mathematical function and thresholding function to perform prediction or probabilities that determine and block unnecessary inputs and result is yield in output nodes. Hidden nodes can be more than one layer. Multiple inputs generate one output at the output node.

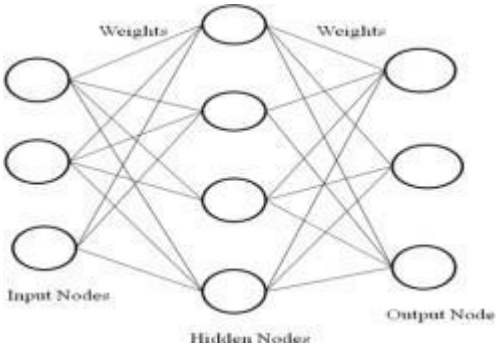


Figure 2.4.4 Artificial Neural Network (ANN) (Manisha M. Kasar et al., 2016)

Convolutional Neural Network (CNN) is another neural network algorithm for face recognition. Similar to ANN, CNN consists of the input layer, hidden layer and output layer. Hidden layers of a CNN consists of multiple layers which are convolutional layers, pooling layers, fully connected layers and normalization layers. However, a thousand or millions of facial images have to be trained for CNN to work accurately and it takes long time to train, for instance Deep-face which is introduced by Facebook.

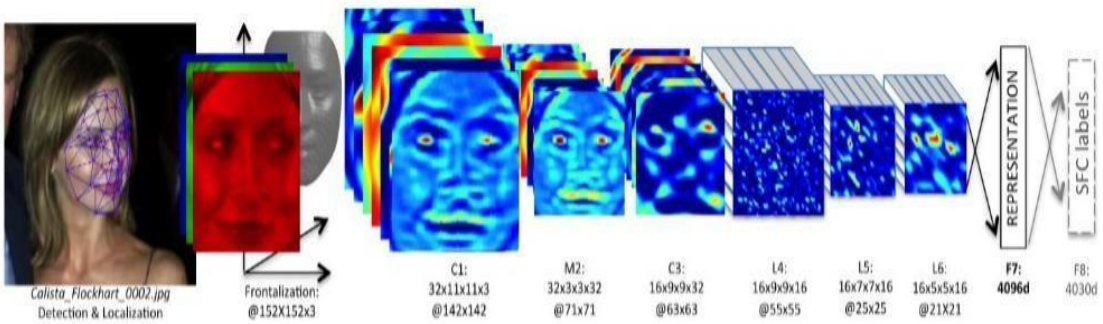


Figure 2.4.5 Deepface Architecture by Facebook (Yaniv Taigman et al, 2014)

2.4.1 Types of Feature Extraction

Divyarajsinh N. Parmar and Brijesh B. Mehta (2013) face recognition system can be categorized into a few Holistic-based methods, Feature-based methods and Hybrid methods. Holistic-based methods are also known as appearance-based methods, which mean entire information about a face patch is involved and used to perform some transformation to obtain a complex representation for recognition. Example of Holistic-based methods are PCA(Principal Component Analysis) and LDA(Linear dependent Analysis).On the other hand, feature-based methods directly extract detail from specific points especially facial features such as eyes, noses, and lips whereas other information which is considered as redundant will be discarded. Example of feature-based method is LBP (Local Binary Pattern). These methods mentioned are usually combined to exist as Hybrid method, for example Holistic-based method combine with Feature-based in order to increase efficiency.

3.SYSTEM REQUIREMENTS SPECIFICATION

This chapter gives an overview of the software and hardware components required for our project.

3.1 Technical Requirements

i. Hardware Requirements

- A standalone computer (i3 5th Gen, 8gb ram or higher)
- High-quality wireless camera to capture images
- Secondary memory to store all the images and database

ii. Software requirements

- PyCharm professional 2017.2.4 or higher
- Python 3.5 or more
- Windows 8 or higher
- Latest version of all libraries

I. Functional Requirements

System functional requirement describes activities and services that must provide.

- A user must be able to manage student records.
- An only authorized user must be able to use the system.
- A system must be attached to wireless camera and face recognition should besmooth.
- The administrator or the person who will be given the access to the system mustlogin into the system before using it.
- The information must be entered and managed properly

II. Non-Functional Requirements

Non-functional Requirements are characteristics or attributes of the system that can judgeits operation. The following points clarify them:

a. Accuracy and Precision: the system should perform its process with accuracy and precision to avoid problems.
b. Flexibility: the system should be easy to modify, any wrong should be correct.
c. Security: the system should be secure and saving student's privacy.
d. Usability: the system should be easy to deal with and simple to understand.
e. Maintainability: the maintenance group should be able to cope up with anyproblem when occurs suddenly.
f. Speed and Responsiveness: Execution of operations should be fast.

Non–Functional Requirements are as follow:

- The GUI of the system will be user-friendly.
- The data that will be shown to the users will be made sure that it is correct and isavailable for the time being. The system will be flexible to changes.
- The system will be extended 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.

III. Student Requirements

- A student needs to enter the proper details while registering him/her.
- He/ She needs to sit properly and capture 10-15 images of himself/herself indifferent directions and expressions.
- At the time of taking attendance, students need to sit properly facing the camera.

4.SYSTEM DESIGN

4.1 System Design

In this phase, the system and software design documents are prepared as per the requirement specification document. This helps define overall system architecture.

There are two kinds of design documents developed in this phase:High-Level Design (HLD)

- Brief description and name of each module
- An outline about the functionality of every module
- Interface relationship and dependencies between modules
- Database tables identified along with their key elements
    - Complete architecture diagrams along with technology detailsLow-Level Design(LLD)
- Functional logic of the modules
- Database tables, which include type and size
- Complete detail of the interface
- Addresses all types of dependency issues
- Listing of error messages
- Complete input and outputs for every module

4.2 UML Design:

Unified Modeling Language (UML) is a general purpose modeling language. The main aim of UML is to define a standard way to visualize the way a system has been designed. It is quite similar to blueprints used in other fields of engineering.
UML is not a programming language; it is rather a visual language. We use UML diagrams to portraythe behavior and structure of a system, UML helps software engineers, businessmen and system architects with modeling, design and analysis. The Object Management Group (OMG) adopted UnifiedModeling Language as a standard in 1997. It's been managed by OMG ever since. International Organization for Standardization (ISO) published UML as an approved standard in 2005. UML has been revised over the years and is reviewed periodically.

Do we really need UML?

- Complex applications need collaboration and planning from multiple teams and hence require a clear and concise way to communicate amongst them.
- Businessmen do not understand code. So UML becomes essential to communicate with non-programmer's essential requirements, functionalities and processes of the system.
- A lot of time is saved down the line when teams are able to visualize processes, user

interactions and static structure of the system.

- UML is linked with object-oriented design and analysis. UML makes the use of elements and forms associations between them to form diagrams. Diagrams in UML can be broadly classified as:

The Primary goals in the design of the UML are as follows:

- Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
- Provide extendibility and specialization mechanisms to extend the core concepts.
- Be independent of particular programming languages and development process.
- Provide a formal basis for understanding the modeling language.
- Encourage the growth of OO tools market.
- Support higher level development concepts such as collaborations, frameworks, patterns and components.
- Integrate best practices.

Types of UML Diagrams:

Structural Diagrams:

Capture static aspects or structure of a system. Structural Diagrams include: Component Diagrams,Object Diagrams, Class Diagrams and Deployment Diagrams.

Behavior Diagrams:

Capture dynamic aspects or behavior of the system. Behavior diagrams include: Use Case Diagrams,State Diagrams, Activity Diagrams and Interaction Diagrams.

The image below shows the hierarchy of diagrams according to UML

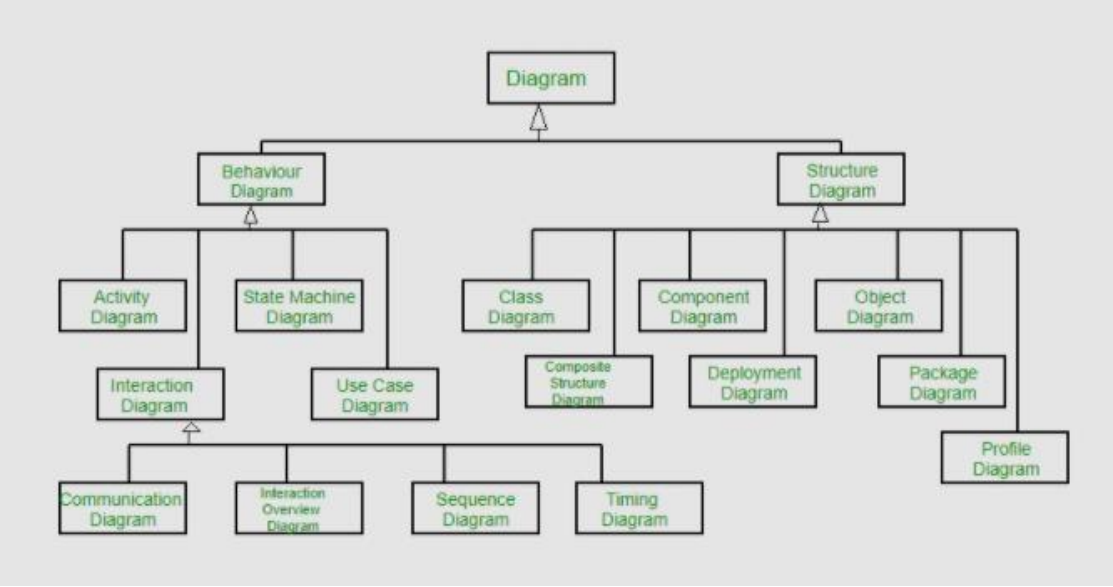


Figure-4.2.1 UML Hierarchy diagrams

4.2.1 CLASS DIAGRAM:

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of staticstructure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.

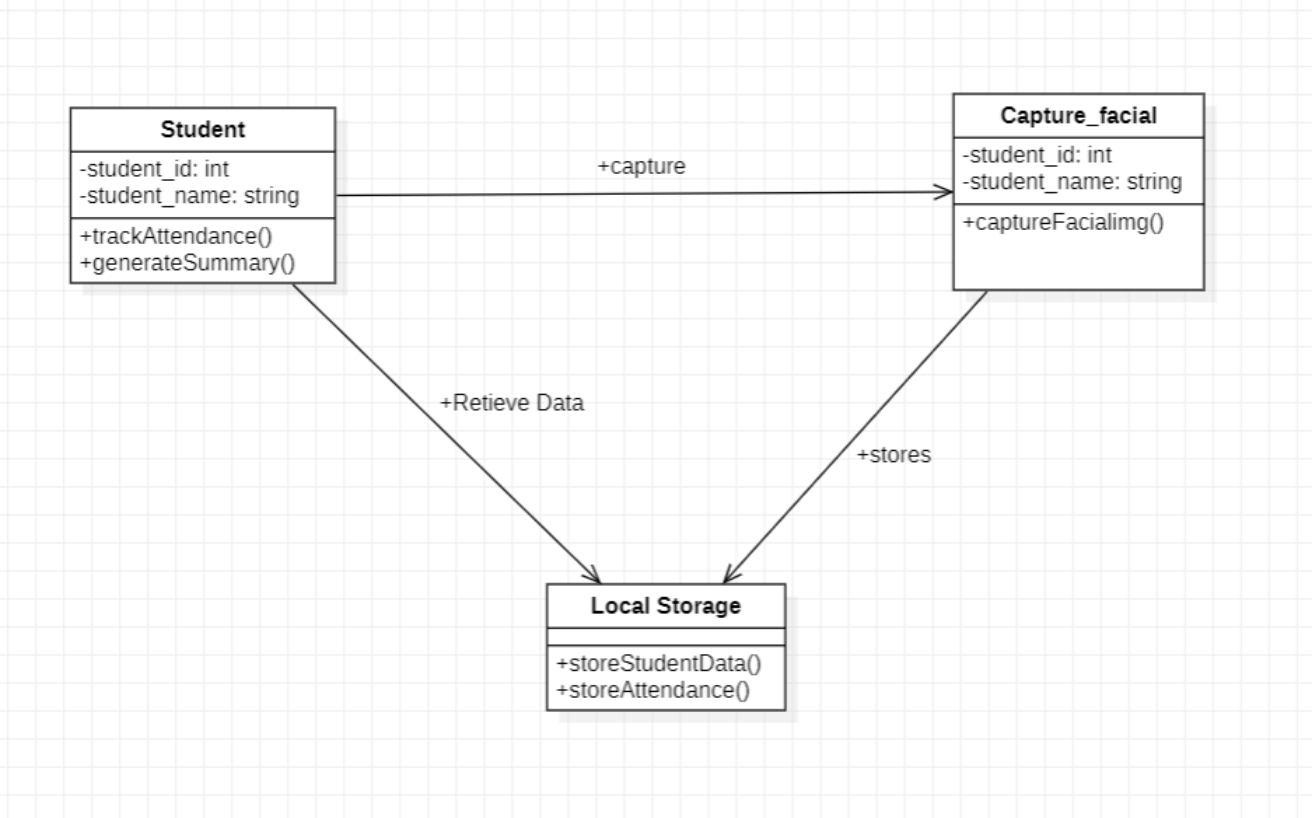


Figure-4.2.1.1 Class Diagram

4.2.2 USE CASE DIAGRAM:

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram definedby and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases.

The main purpose of a use case diagram is to show what systemfunctions are performed for which actor. Roles of the actors in the system can be depicted.

Figure-4.2.2.1 Use Case Diagram

4.2.3 ACTIVITY DIAGRAM:

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

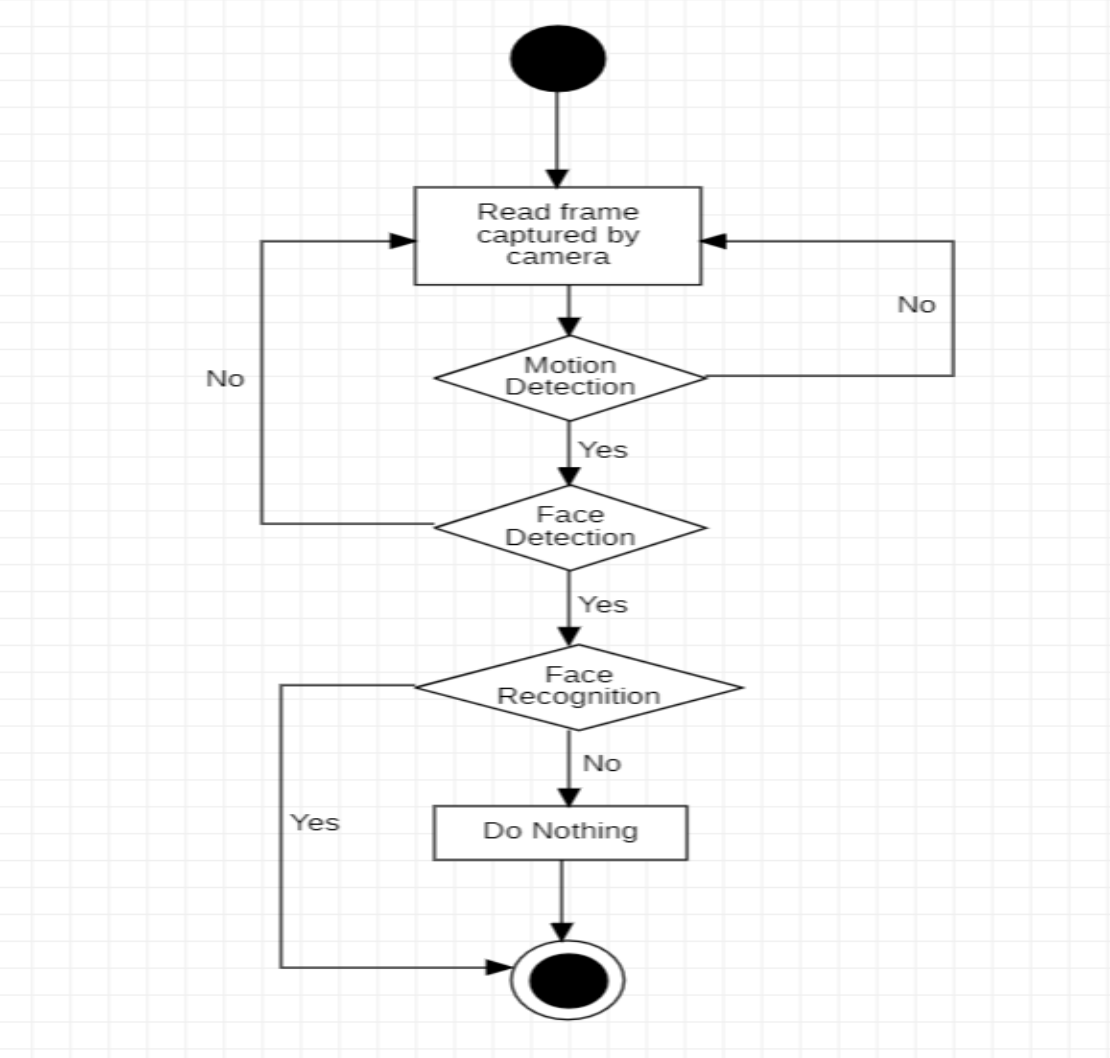


Figure-4.2.3.1 Activity Diagram

4.2.4 SEQUENCE DIAGRAM:

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that showshow processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

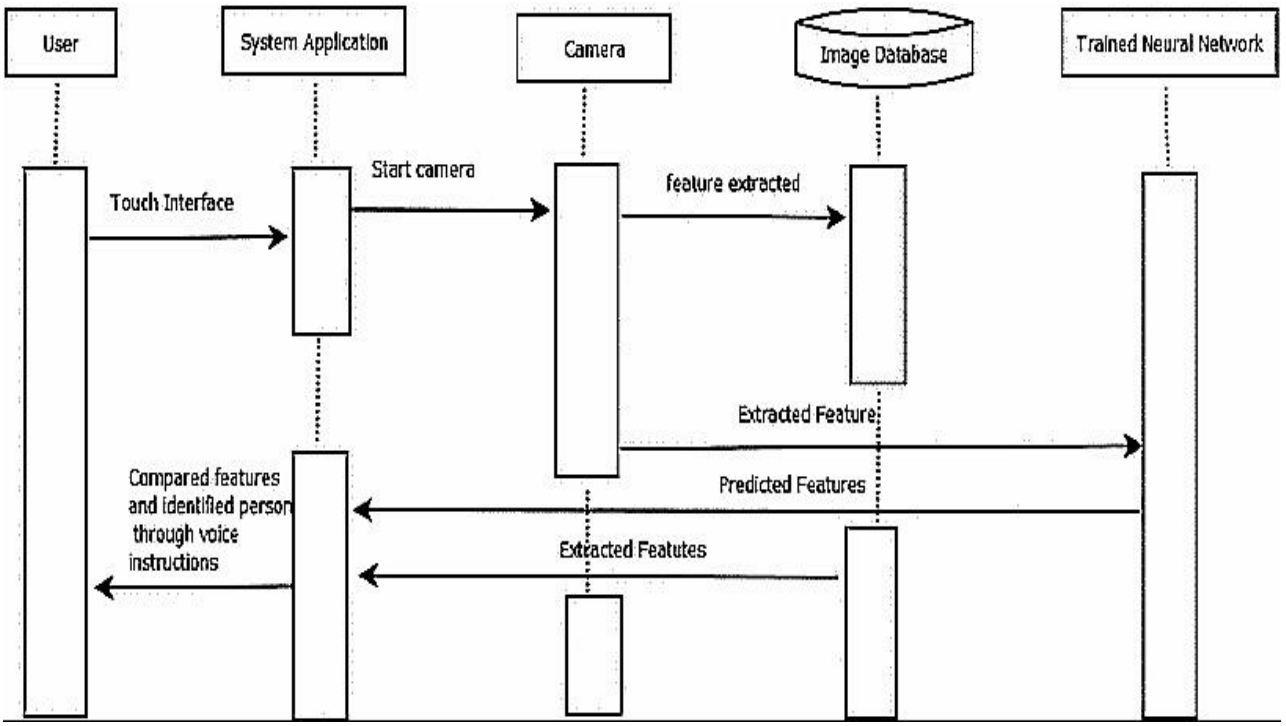


Figure-4.2.4.1 Sequence Diagram

4.3 SYSTEM ARCHITECTURE:

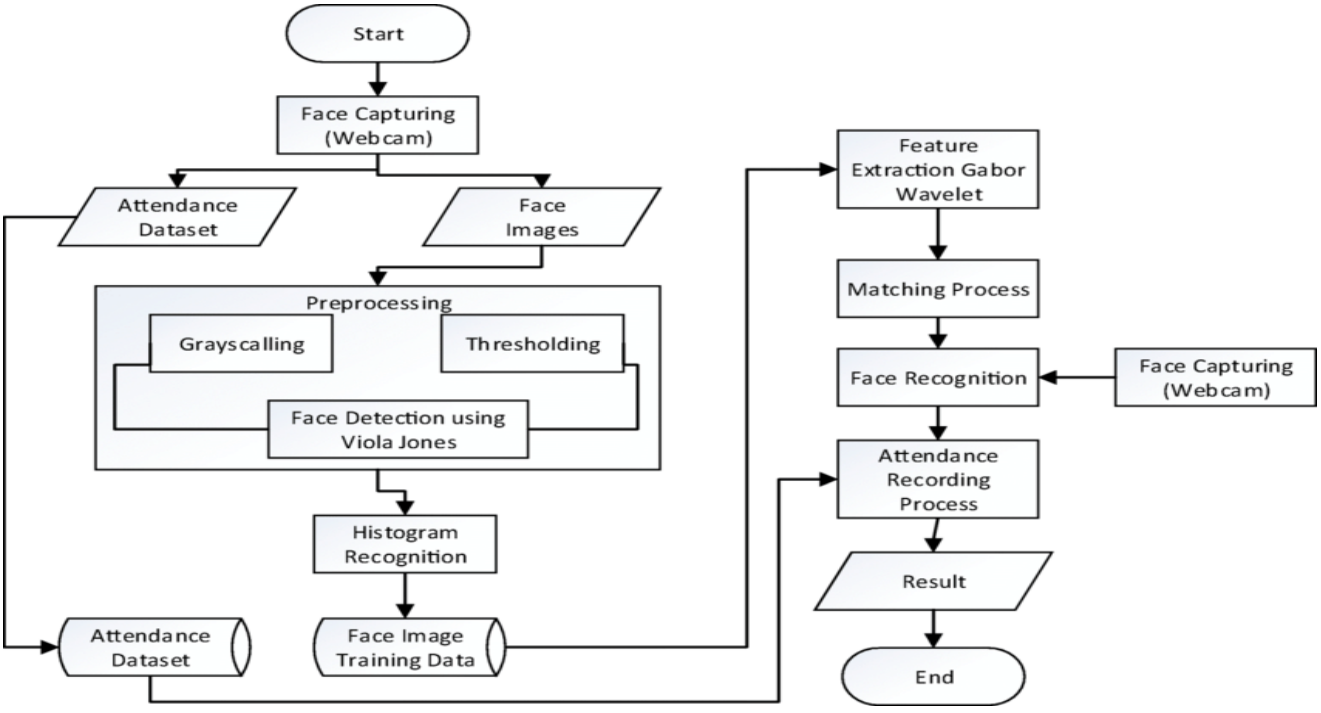


Figure-4.3.1 System Architecture

4.4 DATA FLOW DIAGRAM

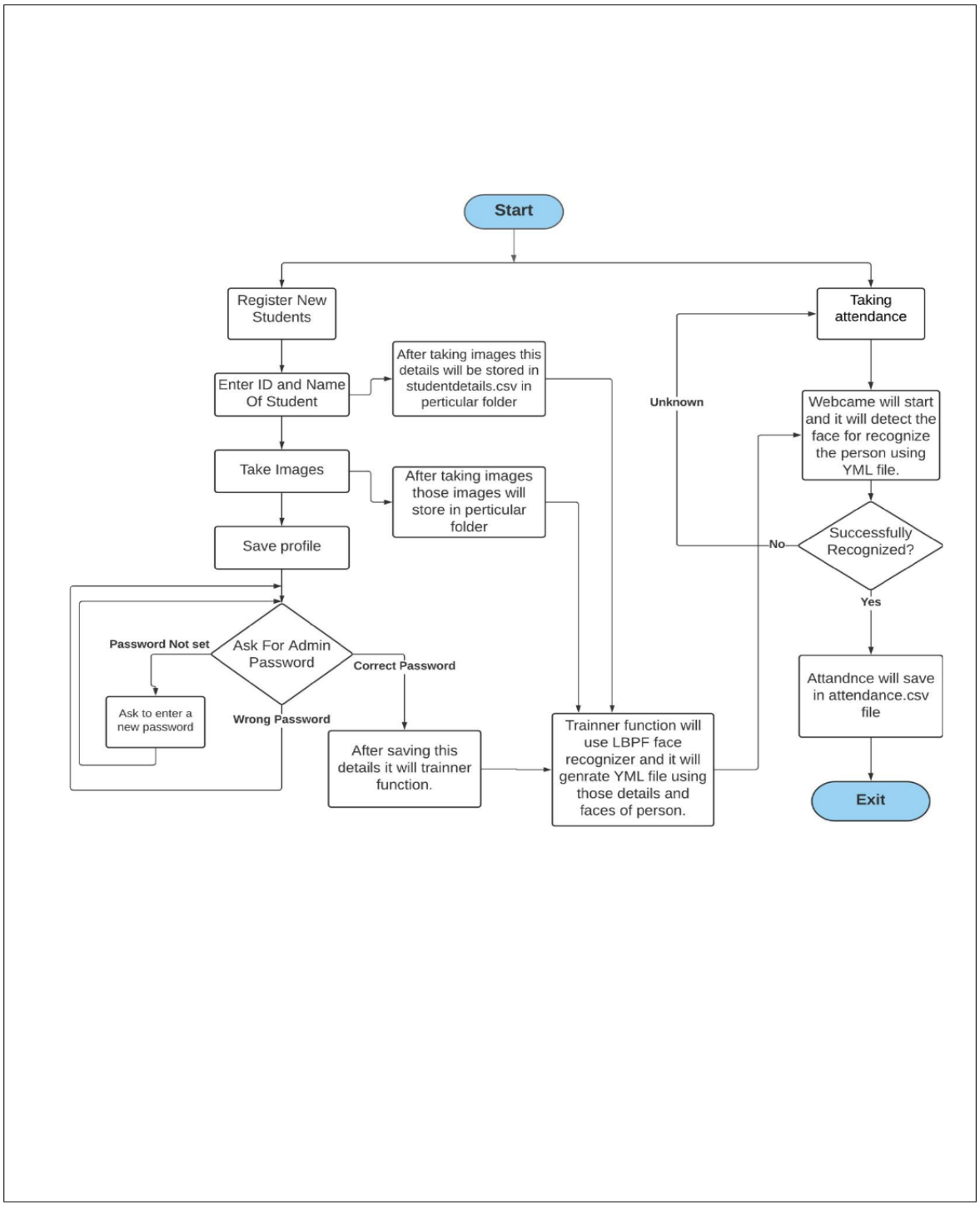


Figure 4.4.1 Data Flow Diagram

4.5 TECHNOLOGY DESCRIPTION

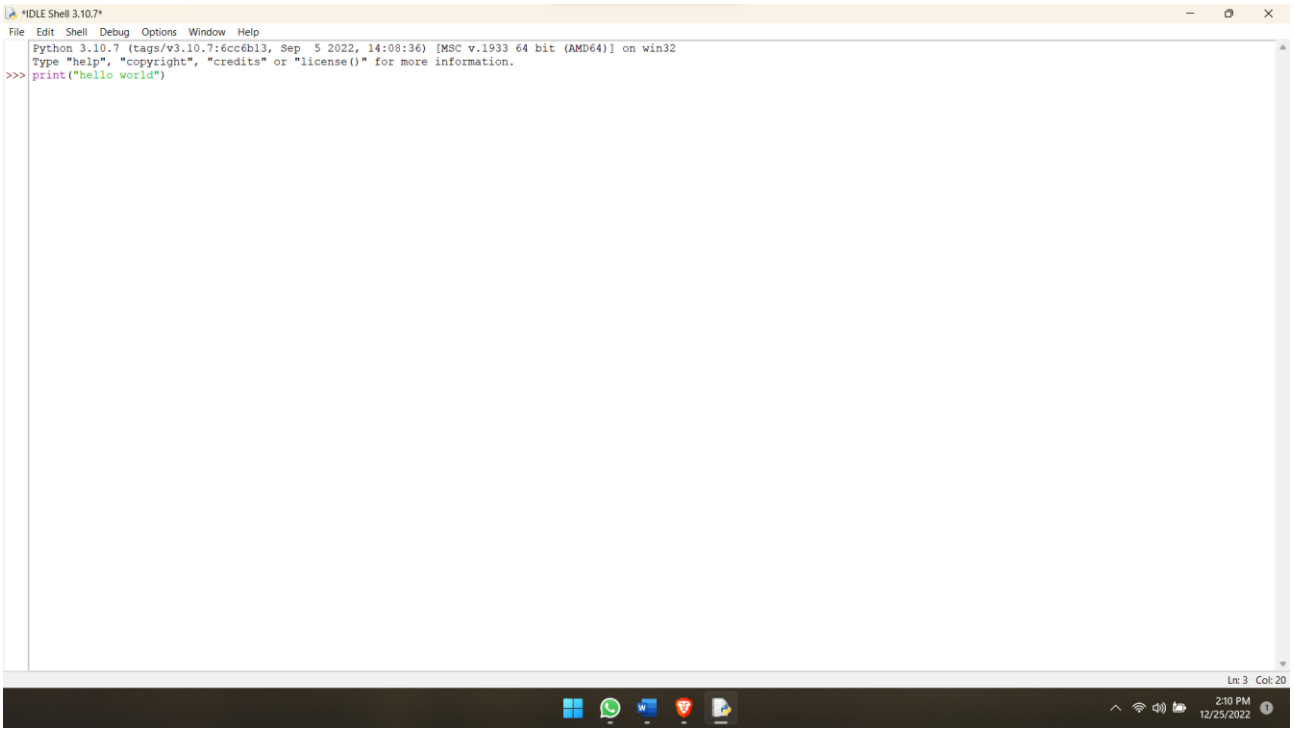
The Python programming language is an Open Source, cross-platform, high level, dynamic, interpreted language. The Python 'philosophy' emphasizes readability, clarity and simplicity, whilst maximizing the power and expressiveness available to the programmer. The ultimate compliment to a Python programmer isnot that his code is clever, but that it is elegant. For these reasons Python is an excellent 'first language',while still being a powerful tool in the hands of the seasoned and cynical programmer. Python is a very flexible language. It is widely used for many different purposes. Typical uses include:

- Web application programming with frameworks like Zope, Django and Turbogears
- System administration tasks via simple scripts
- Desktop applications using GUI toolkits like Tkinter or wxPython (and recently Windows Forms and IronPython)
- Creating windows applications, using the Pywin32 extension for full windows integration and possibly Py2exe to create standalone programs
- Scientific research using packages like Scipy and Matplotlib

4.5.1 IDLE:

IDLE is Python’s Integrated Development and Learning Environment.
IDLE has the following features:

- coded in 100% pure Python, using the tkinter GUI toolkit
- cross-platform: works mostly the same on Windows, Unix, and macOS
- Python shell window (interactive interpreter) with colorizing of code input, output, and error messages
- multi-window text editor with multiple undo, Python colorizing, smart indent, call tips, auto completion, and other features
- search within any window, replace within editor windows, and search through multiple files (grep)
- debugger with persistent breakpoints, stepping, and viewing of global and local namespaces
- configuration, browsers, and other dialogs



4.5.1.1 Python IDLE 3.10

5. IMPLEMENTATION

5.1 Libraries:

Numpy:

NumPy is a Python package which stands for 'Numerical Python'. It is the core library for scientific computing, which contains a powerful n-dimensional array object, provide tools for integrating C, C++etc. It is also useful in linear algebra, random number capability etc. NumPy array can also be used asan efficient multi-dimensional container for generic data. Now, let me tell you what exactly is a pythonnumpy array.

To install Python NumPy, go to your command prompt and type "pip install numpy". Once the installation is completed, go to your IDE (For example: PyCharm) and simply import it by typing: "import numpy as np".

Pandas:

Pandas are an open-source Python Library providing high-performance data manipulation and analysistool using its powerful data structures. The name Pandas is derived from the word Panel Data – an Econometrics from Multidimensional data.

In 2008, developer Wes McKinney started developing pandas when in need of high performance, flexible tool for analysis of data.

Prior to Pandas, Python was majorly used for data munging and preparation. It had very little

contribution towards data analysis. Pandas solved this problem. Using Pandas, we can accomplish fivetypical steps in the processing and analysis of data, regardless of the origin of data — load, prepare, manipulate, model, and analyze.
Python with Pandas is used in a wide range of fields including academic and commercial domains
including finance, economics, Statistics, analytics, etc.

Standard Python distribution doesn't come bundled with Pandas module. A lightweight alternative is to install NumPy using popular Python package installer, **pip.** pip install pandas

Open-CV:

OpenCV is a huge open-source library for computer vision, machine learning, and image processing. OpenCV supports a wide variety of programming languages like Python, C++, Java,etc. It can process images and videos to identify objects, faces, or even the handwriting of a human. When it is integrated with various libraries, such as Numpy which is a highly optimized library for numerical operations, then the number of weapons increases in your Arsenal i.e whatever operations one can do in Numpy can be combined with OpenCV.
This OpenCV tutorial will help you learn the Image-processing from Basics to Advance, like operations on Images, Videos using a huge set of Opencv-programs and projects.

Tkinter:

The tkinter package (“Tk interface”) is the standard Python interface to the Tcl/Tk GUI toolkit. Both Tk and tkinter are available on most Unix platforms, including macOS, as well as on Windows systems.
Running python -m tkinter from the command line should open a window demonstrating a simple Tk interface, letting you know that tkinter is properly installed on your system, and also showing what version of Tcl/Tk is installed, so you can read the Tcl/Tk documentation specific to that version.
Tkinter supports a range of Tcl/Tk versions, built either with or without thread support. The official Python binary release bundles Tcl/Tk 8.6 threaded. See the source code for the _tkinter module for more information about supported versions.
Tkinter is not a thin wrapper, but adds a fair amount of its own logic to make the experience more pythonic. This documentation will concentrate on these additions and changes, and refer to the official Tcl/Tk documentation for details that are unchanged.

PIL Library:

The Python Pillow library is a fork of an older library called PIL. PIL stands for Python Imaging Library, and it’s the original library that enabled Python to deal with images. PIL was discontinued in 2011 and only supports Python 2. To use its developers’ own description, Pillow is the friendly PIL fork that kept the library alive and includes support for Python 3.
There’s more than one module in Python to deal with images and perform image processing. If you want to deal with images directly by manipulating their pixels, then you can

use NumPy and SciPy. Other popular libraries for image processing are OpenCV, scikit-image, and Mahotas. Some of these libraries are faster and more powerful than Pillow.

Time Module: -

Python has a module named time to handle time related task. To use functions defined in the module, we need to import the module first.

Date Time Module:-

A date in python is not a date type of its own, but we can import a module named date time work with dates as a date objects.

5.2 ALGORITHMS:

Haar cascade Algorithm : -

It is a machine learning based approach where a cascade function is trained from a lot of positive and negative images (where positive images are those where the object to be detected is present, negative are those where it is not). It is then used to detect objects in other images. Luckily, OpenCV offers pre-trained Haar cascade algorithms, organized into categories (faces, eyes and so forth), depending on the images they have been trainedon.

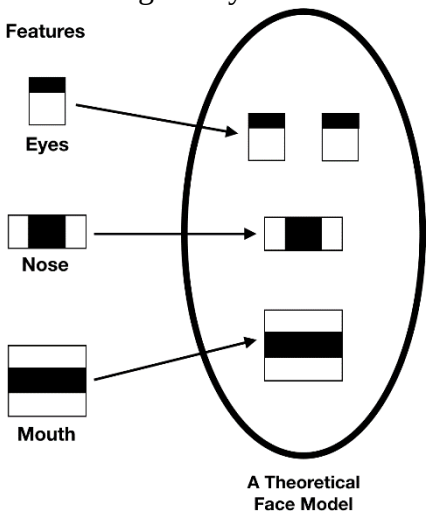


Fig 5.2.2.1 (Haar Features)

LBPH Algorithm : -

Local Binary Pattern (LBP) is a simple yet very efficient texture operator which labelsthe pixels of an image by thresholding the neighborhood of each pixel and considers the result as a binary number. It has further been determined that when LBP is combined with histograms of oriented gradients (HOG) descriptor, it improves the detection performance considerably on some datasets.

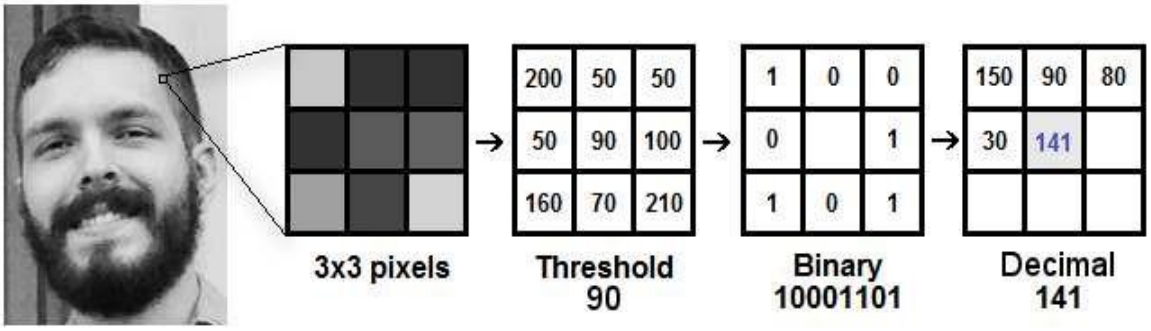


Fig 5.2.2.2 (LBPH Algorithm)

5.3 METHODOLOGY FLOW:

The approach performs face recognition-based student attendance system. The methodology flow begins with the capture of image by using simple and handy interface, followed by pre-processing of the captured facial images, then feature extraction from the facial images, subjective selection and lastly classification of the facial images to be recognized. Both LBP and PCA feature extraction methods are studied in detail and computed in this proposed approach in order to make comparisons. LBP is enhanced in this approach to reduce the illumination effect. An algorithm to combine enhanced LBP and PCA is also designed for subjective selection in order to increase the accuracy. The details of each stage will be discussed in the following sections.

Input Images:

Although our own database should be used to design real time face recognition student attendance system, the databases that are provided by the previous researchers are also used to design the system more effectively, efficiently and for evaluation purposes.

Yale face database is used as both training set and testing set to evaluate the performance. Yale face database contains one hundred and sixty-five grayscale images of fifteen individuals. There are eleven images per individual; each image of the individual is in different condition. The conditions included center-light, with glasses, happy, left-light, without glasses, normal, right-light, sad, sleepy, surprised and wink. These different

variations provided by the database is able to ensure the system to be operated consistently in variety of situations and conditions.

Figure 5.3.1 Sample Images in Yale Face Database (Cvc.cs.yale.edu, 1997)

For our own database, the images of students are captured by using laptop built in camera and mobile phone camera. Each student provided four images, two for training set and two for testing set. The images captured by using laptop built in camera are categorized as low quality images, whereas mobile phone camera captured images are categorized as high quality images. The high quality images consists of seventeen students while low quality images consists of twenty-six students. The recognition rate of low quality images and high quality images will be compared in Chapter 4 to draw a conclusion in term of performance between image sets of different quality.

Figure 5.3.2 Sample of High Quality Images

Figure 5.3.3 Sample of Low Quality Images

Limitations of the Images

The input image for the proposed approach has to be frontal, upright and only a single face. Although the system is designed to be able to recognize the student with glasses and without glasses, student should provide both facial images with and without glasses to be trained to increase the accuracy to be recognized without glasses. The training image and testing image should be captured by using the same device to avoid quality difference. The students have to register in order to be recognized. The enrolment can be done on the spot through the user-

friendly interface.
These conditions have to be satisfied to ensure that the proposed approach can perform well.

Face Detection

Viola-Jones object detection framework will be used to detect the face from the video camera recording frame. The working principle of Viola-Jones algorithm is mentioned in Chapter 2. The limitation of the Viola-Jones framework is that the facial image has to be a frontal upright image, the face of the individual must point towards the camera in a video frame.

Pre-Processing

Testing set and training set images are captured using a camera. There are unwanted noise and uneven lighting exists in the images. Therefore, several pre-processing steps are necessary before proceeding to feature extraction.
Pre-processing steps that would be carried out include scaling of image, median filtering, conversion of colour images to grayscale images and adaptive histogram equalization. The details of these steps would be discussed in the later sections.

Scaling of Image

Scaling of images is one of the frequent tasks in image processing. The size of the images has to be carefully manipulated to prevent loss of spatial information. (Gonzalez, R. C., & Woods, 2008), In order to perform face recognition, the size of the image has to be equalized. This has become crucial, especially in the feature extraction process, the test images and training images have to be in the same size and dimension to ensure the precise outcome. Thus, in this proposed approach test images and train images are standardize at size 250 × 250 pixels.

Median filtering is a robust noise reduction method. It is widely used in various applications due to its capability to remove unwanted noise as well as retaining useful detail in images. Since the colour images captured by using a camera are RGB images, median filtering is done on three different channels of the image. Figure 3.3 shows the image before and after noise removal by median filtering in three channels. If the input image is a grayscale image, then the median filtering can be performed directly without separating the channels.

Figure 5.3.4 Median Filtering Done on Three Channels

Figure 5.3.5 Median Filtering Done on a Single Channel Conversion to Grayscale Image

Camera captures color images, however the proposed contrast improvement method CLAHE can only be performed on grayscale images. After improving the contrast, the illumination effect of the images able to be reduced. LBP extracts the grayscale features from the contrast improved images as 8 bit texture descriptor (Ojala, T. et al., 2002).Therefore, color images have to be converted to grayscale images before proceeding to the later steps. By converting color images to grayscale images, the complexity of the computation can be reduced resulting in higher speed of computation (Kanan and Cottrell, 2012). Figure 3.4 shows the conversion of images to grayscale image.

Figure 5.3.6 Conversion of Image to Grayscale Image

Contrast Limited Adaptive Histogram Equalization

Histogram equalization or histogram stretching is a technique of image contrast enhancement. (Pratiksha M. Patel, 2016). The contrast improvement is usually performed on the grayscale images. Image contrast is improved by stretching the range of its pixel intensity values to span over the desired range of values, between 0 and 255 in grayscale. The reason that Contrast Limited Adaptive Histogram Equalization (CLAHE) is used instead of histogram equalization is because histogram equalization depends on the global statistics. Hence, it causes over enhancement of some parts of image while other parts are not enhanced properly. This distorts the features of the image. It is a serious issue because the features of the image have to be extracted for the face recognition. Thus, CLAHE which is depend on local statistic is used.

Figure 5.3.7 Contrast Improvement

Feature Extraction

Different facial images mean there are changes in textural or geometric information. In order to perform face recognition, these features have to be extracted from the facial images and classified appropriately. In this project, enhanced LBP and PCA are used for face recognition. The idea comes from nature of human visual perception which performs face recognition depending on the local statistic and global statistic features. Enhanced LBP extracts the local grayscale features by performing feature extraction on a small region throughout the entire image. On the other hand, PCA extracts the global grayscale features which means feature extraction is performed on the whole image.

Working Principle of Original LBP

LBP is basically a texture based descriptor which it encoded local primitive into binary string. (Timo Ojala et al., 2002). The original LBP operator works on a 3 × 3 mask size. 3 × 3 mask size contains 9 pixels. The center pixel will be used as a threshold to convert the neighboring pixels (the other 8 pixels) into binary digit.

If the neighboring pixel value is larger than the center pixel value, then it is assigned to 1, otherwise it is assigned to 0. After that, the neighborhoods pixel bits are concatenated to a binary code to form a byte value representing the center pixel. Figure 3.6 shows an example of LBP conversion.

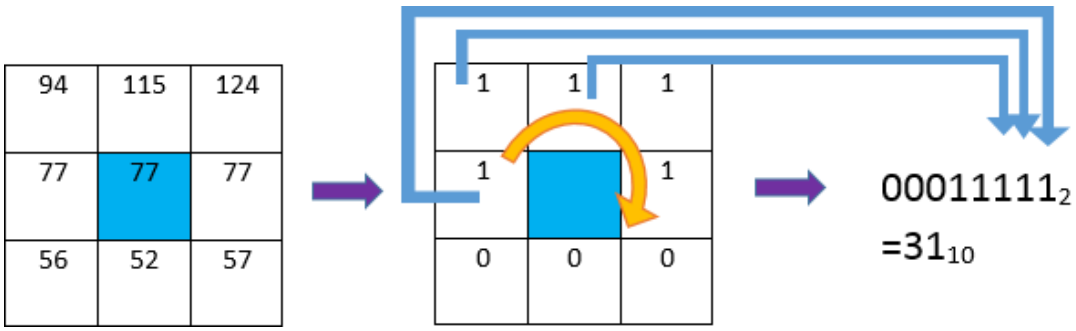


Figure 5.3.8 Example of LBP Conversion

5.4 EXECUTABLE CODE:

```
######################## IMPORTING ##############################
import tkinter as tk
from tkinter import ttk
from tkinter import messagebox as mess
import tkinter.simpledialog as tsd
import cv2,os
import csv
```

```
import numpy as np
from PIL import Image
import pandas as pd
import datetime
import time

########################## FUNCTIONS ###############################

def assure_path_exists(path):
    dir = os.path.dirname(path)
    if not os.path.exists(dir):
        os.makedirs(dir)

####################################################################

def tick():
    time_string = time.strftime('%H:%M:%S')
    clock.config(text=time_string)
    clock.after(200,tick)

####################################################################

def contact():
    mess._show(title='Contact us', message="Please contact us on :
'facialreccognition@gmail.com' ")

####################################################################

def check_haarcascadefile():
    exists = os.path.isfile("haarcascade_frontalface_default.xml")
    if exists:
        pass
    else:
        mess._show(title='Some file missing', message='Please contact us for help')
        window.destroy()

####################################################################

def save_pass():
    assure_path_exists("TrainingImageLabel/")
    exists1 = os.path.isfile("TrainingImageLabel\psd.txt")
    if exists1:
        tf = open("TrainingImageLabel\psd.txt", "r")
        key = tf.read()
```

```
    else:
        master.destroy()
        new_pas = tsd.askstring('Old Password not found', 'Please enter a new password below',
show='*')
        if new_pas == None:
            mess._show(title='No Password Entered', message='Password not set!! Please try
again')
        else:
            tf = open("TrainingImageLabel\psd.txt", "w")
            tf.write(new_pas)
            mess._show(title='Password Registered', message='New password was registered
successfully!!')
            return
    op = (old.get())
    newp= (new.get())
    nnewp = (nnew.get())
    if (op == key):
        if(newp == nnewp):
            txf = open("TrainingImageLabel\psd.txt", "w")
            txf.write(newp)
        else:
            mess._show(title='Error', message='Confirm new password again!!!')
            return
    else:
        mess._show(title='Wrong Password', message='Please enter correct old password.')
        return
    mess._show(title='Password Changed', message='Password changed successfully!!')
    master.destroy()

####################################################################

def change_pass():
    global master
    master = tk.Tk()
    master.geometry("400x160")
    master.resizable(False,False)
    master.title("Change Password")
    master.configure(background="white")
    lbl4 = tk.Label(master,text='    Enter Old Password',bg='white',font=('times', 12, ' bold '))
    lbl4.place(x=10,y=10)
    global old
    old=tk.Entry(master,width=25 ,fg="black",relief='solid',font=('times', 12, ' bold '),show='*')
    old.place(x=180,y=10)
    lbl5 = tk.Label(master, text='   Enter New Password', bg='white', font=('times', 12, ' bold '))
```

```
    lbl5.place(x=10, y=45)
    global new
    new = tk.Entry(master, width=25, fg="black",relief='solid', font=('times', 12, ' bold '),show='*')
    new.place(x=180, y=45)
    lbl6 = tk.Label(master, text='Confirm New Password', bg='white', font=('times', 12, ' bold '))
    lbl6.place(x=10, y=80)
    global nnew
    nnew = tk.Entry(master, width=25, fg="black", relief='solid',font=('times', 12, ' bold '),show='*')
    nnew.place(x=180, y=80)
    cancel=tk.Button(master,text="Cancel", command=master.destroy ,fg="black"   ,bg="red" ,height=1,width=25 , activebackground = "white" ,font=('times', 10, ' bold '))
    cancel.place(x=200, y=120)
    save1 = tk.Button(master, text="Save", command=save_pass, fg="black", bg="#3ece48", height = 1,width=25, activebackground="white", font=('times', 10, ' bold '))
    save1.place(x=10, y=120)
    master.mainloop()

####################################################################

def psw():
    assure_path_exists("TrainingImageLabel/")
    exists1 = os.path.isfile("TrainingImageLabel\psd.txt")
    if exists1:
        tf = open("TrainingImageLabel\psd.txt", "r")
        key = tf.read()
    else:
        new_pas = tsd.askstring('Old Password not found', 'Please enter a new password below', show='*')
        if new_pas == None:
            mess._show(title='No Password Entered', message='Password not set!! Please try again')
        else:
            tf = open("TrainingImageLabel\psd.txt", "w")
            tf.write(new_pas)
            mess._show(title='Password Registered', message='New password was registered successfully!!')
            return
    password = tsd.askstring('Password', 'Enter Password', show='*')
    if (password == key):
        TrainImages()
    elif (password == None):
        pass
```

```
    else:
        mess._show(title='Wrong Password', message='You have entered wrong password')

######################################################################

def clear():
    txt.delete(0, 'end')
    res = "1)Take Images  >>>  2)Save Profile"
    message1.configure(text=res)


def clear2():
    txt2.delete(0, 'end')
    res = "1)Take Images  >>>  2)Save Profile"
    message1.configure(text=res)

######################################################################

def TakeImages():
    check_haarcascadefile()
    columns = ['SERIAL NO.', '', 'ID', '', 'NAME']
    assure_path_exists("StudentDetails/")
    assure_path_exists("TrainingImage/")
    serial = 0
    exists = os.path.isfile("StudentDetails\StudentDetails.csv")
    if exists:
        with open("StudentDetails\StudentDetails.csv", 'r') as csvFile1:
            reader1 = csv.reader(csvFile1)
            for l in reader1:
                serial = serial + 1
        serial = (serial // 2)
        csvFile1.close()
    else:
        with open("StudentDetails\StudentDetails.csv", 'a+') as csvFile1:
            writer = csv.writer(csvFile1)
            writer.writerow(columns)
            serial = 1
        csvFile1.close()
    Id = (txt.get())
    name = (txt2.get())
    if ((name.isalpha()) or (' ' in name)):
        cam = cv2.VideoCapture(0)
        harcascadePath = "haarcascade_frontalface_default.xml"
        detector = cv2.CascadeClassifier(harcascadePath)
```

```
        sampleNum = 0
        while (True):
            ret, img = cam.read()
            gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
            faces = detector.detectMultiScale(gray, 1.3, 5)
            for (x, y, w, h) in faces:
                cv2.rectangle(img, (x, y), (x + w, y + h), (255, 0, 0), 2)
                # incrementing sample number
                sampleNum = sampleNum + 1
                # saving the captured face in the dataset folder TrainingImage
                cv2.imwrite("TrainingImage\ " + name + "." + str(serial) + "." + Id + '.' +
str(sampleNum) + ".jpg",
                            gray[y:y + h, x:x + w])
                # display the frame
                cv2.imshow('Taking Images', img)
            # wait for 100 miliseconds
            if cv2.waitKey(100) & 0xFF == ord('q'):
                break
            # break if the sample number is morethan 100
            elif sampleNum > 100:
                break
        cam.release()
        cv2.destroyAllWindows()
        res = "Images Taken for ID : " + Id
        row = [serial, '', Id, '', name]
        with open('StudentDetails\StudentDetails.csv', 'a+') as csvFile:
            writer = csv.writer(csvFile)
            writer.writerow(row)
        csvFile.close()
        message1.configure(text=res)
    else:
        if (name.isalpha() == False):
            res = "Enter Correct name"
            message.configure(text=res)

######################################################################

def TrainImages():
    check_haarcascadefile()
    assure_path_exists("TrainingImageLabel/")
    recognizer = cv2.face_LBPHFaceRecognizer.create()
    harcascadePath = "haarcascade_frontalface_default.xml"
    detector = cv2.CascadeClassifier(harcascadePath)
    faces, ID = getImagesAndLabels("TrainingImage")
```

```
    try:
        recognizer.train(faces, np.array(ID))
    except:
        mess._show(title='No Registrations', message='Please Register someone first!!!')
        return
    recognizer.save("TrainingImageLabel\Trainner.yml")
    res = "Profile Saved Successfully"
    message1.configure(text=res)
    message.configure(text='Total Registrations till now  : ' + str(ID[0]))

##########################################################################

def getImagesAndLabels(path):
    # get the path of all the files in the folder
    imagePaths = [os.path.join(path, f) for f in os.listdir(path)]
    # create empth face list
    faces = []
    # create empty ID list
    Ids = []
    # now looping through all the image paths and loading the Ids and the images
    for imagePath in imagePaths:
        # loading the image and converting it to gray scale
        pilImage = Image.open(imagePath).convert('L')
        # Now we are converting the PIL image into numpy array
        imageNp = np.array(pilImage, 'uint8')
        # getting the Id from the image
        ID = int(os.path.split(imagePath)[-1].split(".")[1])
        # extract the face from the training image sample
        faces.append(imageNp)
        Ids.append(ID)
    return faces, Ids

##########################################################################

def TrackImages():
    check_haarcascadefile()
    assure_path_exists("Attendance/")
    assure_path_exists("StudentDetails/")
    for k in tv.get_children():
        tv.delete(k)
    msg = ''
    i = 0
    j = 0
    recognizer = cv2.face.LBPHFaceRecognizer_create()  # cv2.createLBPHFaceRecognizer()
```

```
    exists3 = os.path.isfile("TrainingImageLabel\Trainner.yml")
    if exists3:
        recognizer.read("TrainingImageLabel\Trainner.yml")
    else:
        mess._show(title='Data Missing', message='Please click on Save Profile to reset data!!')
        return
    harcascadePath = "haarcascade_frontalface_default.xml"
    faceCascade = cv2.CascadeClassifier(harcascadePath);

    cam = cv2.VideoCapture(0)
    font = cv2.FONT_HERSHEY_SIMPLEX
    col_names = ['Id', '', 'Name', '', 'Date', '', 'Time']
    exists1 = os.path.isfile("StudentDetails\StudentDetails.csv")
    if exists1:
        df = pd.read_csv("StudentDetails\StudentDetails.csv")
    else:
        mess._show(title='Details Missing', message='Students details are missing, please
check!')
        cam.release()
        cv2.destroyAllWindows()
        window.destroy()
    while True:
        ret, im = cam.read()
        gray = cv2.cvtColor(im, cv2.COLOR_BGR2GRAY)
        faces = faceCascade.detectMultiScale(gray, 1.2, 5)
        for (x, y, w, h) in faces:
            cv2.rectangle(im, (x, y), (x + w, y + h), (225, 0, 0), 2)
            serial, conf = recognizer.predict(gray[y:y + h, x:x + w])
            if (conf < 50):
                ts = time.time()
                date = datetime.datetime.fromtimestamp(ts).strftime('%d-%m-%Y')
                timeStamp = datetime.datetime.fromtimestamp(ts).strftime('%H:%M:%S')
                aa = df.loc[df['SERIAL NO.'] == serial]['NAME'].values
                ID = df.loc[df['SERIAL NO.'] == serial]['ID'].values
                ID = str(ID)
                ID = ID[1:-1]
                bb = str(aa)
                bb = bb[2:-2]
                attendance = [str(ID), '', bb, '', str(date), '', str(timeStamp)]

            else:
                Id = 'Unknown'
                bb = str(Id)
            cv2.putText(im, str(bb), (x, y + h), font, 1, (255, 255, 255), 2)
```

```
        cv2.imshow('Taking Attendance', im)
        if (cv2.waitKey(1) == ord('q')):
            break
    ts = time.time()
    date = datetime.datetime.fromtimestamp(ts).strftime('%d-%m-%Y')
    exists = os.path.isfile("Attendance\Attendance_" + date + ".csv")
    if exists:
        with open("Attendance\Attendance_" + date + ".csv", 'a+') as csvFile1:
            writer = csv.writer(csvFile1)
            writer.writerow(attendance)
        csvFile1.close()
    else:
        with open("Attendance\Attendance_" + date + ".csv", 'a+') as csvFile1:
            writer = csv.writer(csvFile1)
            writer.writerow(col_names)
            writer.writerow(attendance)
        csvFile1.close()
    with open("Attendance\Attendance_" + date + ".csv", 'r') as csvFile1:
        reader1 = csv.reader(csvFile1)
        for lines in reader1:
            i = i + 1
            if (i > 1):
                if (i % 2 != 0):
                    iidd = str(lines[0]) + '   '
                    tv.insert('', 0, text=iidd, values=(str(lines[2]), str(lines[4]), str(lines[6])))
    csvFile1.close()
    cam.release()
    cv2.destroyAllWindows()
```

################################ **USED STUFFS** ####################

```
global key
key = ''

ts = time.time()
date = datetime.datetime.fromtimestamp(ts).strftime('%d-%m-%Y')
day,month,year=date.split("-")

mont={'01':'January',
      '02':'February',
      '03':'March',
      '04':'April',
      '05':'May',
      '06':'June',
```

```
    '07':'July',
    '08':'August',
    '09':'September',
    '10':'October',
    '11':'November',
    '12':'December'
    }

############################ GUI FRONT-END ######################

window = tk.Tk()
window.geometry("1920x1080")
window.resizable(True,False)
window.title("Attendance System")
window.configure(background='#262523')

frame1 = tk.Frame(window, bg="#00aeff")
frame1.place(relx=0.11, rely=0.17, relwidth=0.39, relheight=0.80)

frame2 = tk.Frame(window, bg="#00aeff")
frame2.place(relx=0.51, rely=0.17, relwidth=0.38, relheight=0.80)

message3 = tk.Label(window, text="Face Recognition Based Attendance System" ,fg="white",bg="#262523" ,width=55 ,height=1,font=('times', 29, ' bold '))
message3.place(x=10, y=10)

frame3 = tk.Frame(window, bg="#c4c6ce")
frame3.place(relx=0.52, rely=0.09, relwidth=0.09, relheight=0.07)

frame4 = tk.Frame(window, bg="#c4c6ce")
frame4.place(relx=0.36, rely=0.09, relwidth=0.16, relheight=0.07)

datef = tk.Label(frame4, text = day+"-"+mont[month]+"-"+year+" | ", fg="orange",bg="#262523" ,width=55 ,height=1,font=('times', 22, ' bold '))
datef.pack(fill='both',expand=1)

clock = tk.Label(frame3,fg="orange",bg="#262523" ,width=65 ,height=1,font=('times', 22, ' bold '))
clock.pack(fill='both',expand=1)
tick()

head2 = tk.Label(frame2, text="                    For New Registrations                    ", fg="black",bg="#3ece48" ,font=('times', 17, ' bold ') )
head2.grid(row=0,column=0)
```

```
head1 = tk.Label(frame1, text="                For Already Registered                ",
fg="black",bg="#3ece48" ,font=('times', 17, ' bold ') )
head1.place(x=0,y=0)

lbl = tk.Label(frame2, text="Enter ID",width=20  ,height=1  ,fg="black"  ,bg="#00aeff"
,font=('times', 17, ' bold ') )
lbl.place(x=80, y=55)

txt = tk.Entry(frame2,width=32 ,fg="black",font=('times', 15, ' bold '))
txt.place(x=30, y=88)

lbl2 = tk.Label(frame2, text="Enter Name",width=20  ,fg="black"  ,bg="#00aeff"
,font=('times', 17, ' bold '))
lbl2.place(x=80, y=140)

txt2 = tk.Entry(frame2,width=32 ,fg="black",font=('times', 15, ' bold ')  )
txt2.place(x=30, y=173)


message1 = tk.Label(frame2, text="1)Take Images  >>>  2)Save Profile" ,bg="#00aeff"
,fg="black"  ,width=39 ,height=1, activebackground = "yellow" ,font=('times', 15, ' bold '))
message1.place(x=7, y=230)

message = tk.Label(frame2, text="" ,bg="#00aeff" ,fg="black"  ,width=39,height=1,
activebackground = "yellow" ,font=('times', 16, ' bold '))
message.place(x=7, y=450)

lbl3 = tk.Label(frame1, text="Attendance",width=20  ,fg="black"  ,bg="#00aeff"  ,height=1
,font=('times', 17, ' bold '))
lbl3.place(x=100, y=115)

res=0
exists = os.path.isfile("StudentDetails\StudentDetails.csv")
if exists:
    with open("StudentDetails\StudentDetails.csv", 'r') as csvFile1:
        reader1 = csv.reader(csvFile1)
        for l in reader1:
            res = res + 1
    res = (res // 2) - 1
    csvFile1.close()
else:
    res = 0
message.configure(text='Total Registrations till now  : '+str(res))
```

```
##################### MENUBAR ################################

menubar = tk.Menu(window,relief='ridge')
filemenu = tk.Menu(menubar,tearoff=0)
filemenu.add_command(label='Change Password', command = change_pass)
filemenu.add_command(label='Contact Us', command = contact)
filemenu.add_command(label='Exit',command = window.destroy)
menubar.add_cascade(label='Help',font=('times', 29, ' bold '),menu=filemenu)

################## TREEVIEW ATTENDANCE TABLE #############

tv= ttk.Treeview(frame1,height =13,columns = ('name','date','time'))
tv.column('#0',width=82)
tv.column('name',width=130)
tv.column('date',width=133)
tv.column('time',width=133)
tv.grid(row=2,column=0,padx=(0,0),pady=(150,0),columnspan=4)
tv.heading('#0',text ='ID')
tv.heading('name',text ='NAME')
tv.heading('date',text ='DATE')
tv.heading('time',text ='TIME')

######################SCROLLBAR################################

scroll=ttk.Scrollbar(frame1,orient='vertical',command=tv.yview)
scroll.grid(row=2,column=4,padx=(0,100),pady=(150,0),sticky='ns')
tv.configure(yscrollcommand=scroll.set)

######################BUTTONS##################################

clearButton = tk.Button(frame2, text="Clear", command=clear  ,fg="black"  ,bg="#ea2a2a"
,width=11 ,activebackground = "white" ,font=('times', 11, ' bold '))
clearButton.place(x=335, y=86)
clearButton2 = tk.Button(frame2, text="Clear", command=clear2  ,fg="black"  ,bg="#ea2a2a"
,width=11 , activebackground = "white" ,font=('times', 11, ' bold '))
clearButton2.place(x=335, y=172)
takeImg = tk.Button(frame2, text="Take Images", command=TakeImages  ,fg="white"
,bg="blue"  ,width=34  ,height=1, activebackground = "white" ,font=('times', 15, ' bold '))
takeImg.place(x=30, y=300)
trainImg = tk.Button(frame2, text="Save Profile", command=psw ,fg="white"  ,bg="blue"
,width=34  ,height=1, activebackground = "white" ,font=('times', 15, ' bold '))
trainImg.place(x=30, y=380)
trackImg = tk.Button(frame1, text="Take Attendance", command=TrackImages  ,fg="black"
```

```
,bg="yellow"  ,width=35  ,height=1, activebackground = "white" ,font=('times', 15, ' bold '))
trackImg.place(x=30,y=50)
quitWindow = tk.Button(frame1, text="Quit", command=window.destroy  ,fg="black"
,bg="red"  ,width=35 ,height=1, activebackground = "white" ,font=('times', 15, ' bold '))
quitWindow.place(x=30, y=450)

###################### END ######################################

window.configure(menu=menubar)
window.mainloop()
```

6. TESTING

6.1 TESTING DEFINATION:

- Software testing is the process of valuating and verifying that a software product or application does what it is supposed to do.
- The benefits of testing include preventing bugs, reducing development costs and improving performance.

TESTING AND TEST CASES:

- Software testing is the process of valuating and verifying that a software product or application does what it is supposed to do.
- The benefits of testing include preventing bugs, reducing development costs andimproving performance.

6.2 TYPES OF TESTING:

White Box Testing

In white-box testing, the developer will inspect every line of code before handing it over to the testingteam or the concerned test engineers.

Black Box Testing

Another type of manual testing is black-box testing. In this testing, the test engineer will analyze the software against requirements, identify the defects or bug, and sends it back to the developmentteam.

Functional Testing

The test engineer will check all the components systematically against requirement specifications is known as functional testing. Functional testing is also known as Component testing.

Non-function Testing

The next part of black-box testing is non-functional testing. It provides detailed information onsoftware product performance and used technologies.

Grey Box Testing

Another part of manual testing is Grey box testing. It is a collaboration of black box and whitebox testing.
Since, the grey box testing includes access to internal coding for designing test cases. Grey box testingis performed by a person who knows coding as well as testing.

6.3 UNIT TESTING

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

The unittest unit testing framework was originally inspired by JUnit and has a similar flavor as majorunit testing frameworks in other languages. It supports test automation, sharing of setup and shutdown
code for tests, aggregation of tests into collections, and independence of the tests from the reportingframework.

To achieve this, unittest supports some important concepts in an object-oriented way:

6.3.1 Test fixture

A *test fixture* represents the preparation needed to perform one or more tests, and any associated cleanup actions. This may involve, for example, creating temporary or proxy databases, directories, orstarting a server process.

6.3.2 Test case

A *test case* is the individual unit of testing. It checks for a specific response to a particular set of inputs. unittest provides a base class, TestCase, which may be used to create new test cases.

6.3.3 Test suite

A *test suite* is a collection of test cases, test suites, or both. It is used to aggregate tests that should be executed together.

6.3.4 Test runner

A *test runner* is a component which orchestrates the execution of tests and provides the outcome to theuser. The runner may use a graphical interface, a textual interface, or return a special value to indicatethe results of executing the tests.

Outcomes Possible:

There are three types of possible test outcomes:

- OK – This means that all the tests are passed.
- FAIL – This means that the test did not pass and an AssertionError exception is raised.
- ERROR – This means that the test raises an exception other than AssertionError.

6.4 TEST CASES

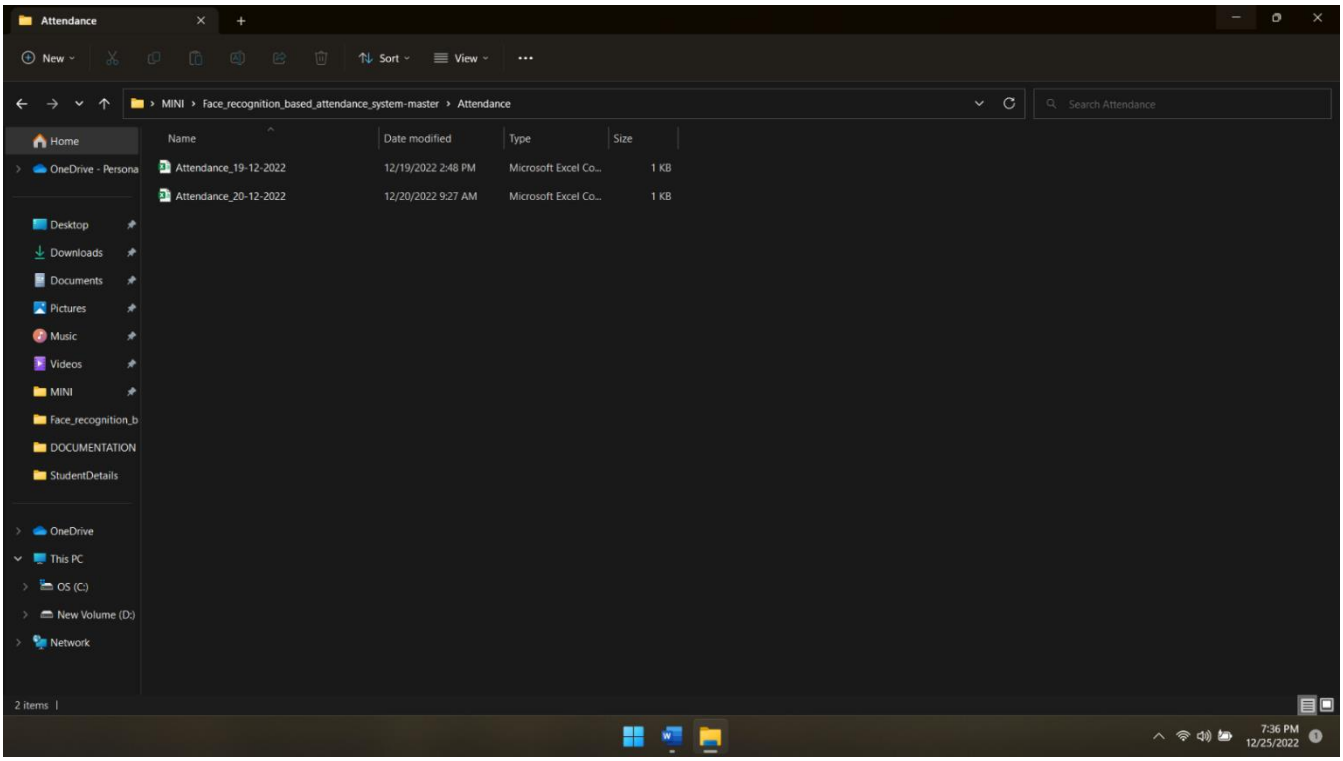
Tested	Test name	Inputs	Expected output	Actual Output	status
1	Capture Images	A person's face	Images are captured and stored	Images are captured and stored	success
2	Train the image dataset	Stored images of a face	Create histogram and store values	Histogram is created and values are stored	success
3	Face Recognition	A live stream of a person's face	Name of detected person is displayed on the screen	Name of detected person is displayed on the screen	success
4	Update attendance for multiple people at once	Multiple faces from a live video stream	Update Attendance for all faces detected	Attendance is updated for all faces	success
5	Detect face in low light conditions	Face in low light environment	Update attendance	Could not recognize	Failed
6	Backgroundtest	Dark Background in	Predict Face or No Face	Not Predicted	Failed

		live detection			
7	TestingCamera	Camera with lowclarity	Predict Face or No Face	Not Predicted	Failed

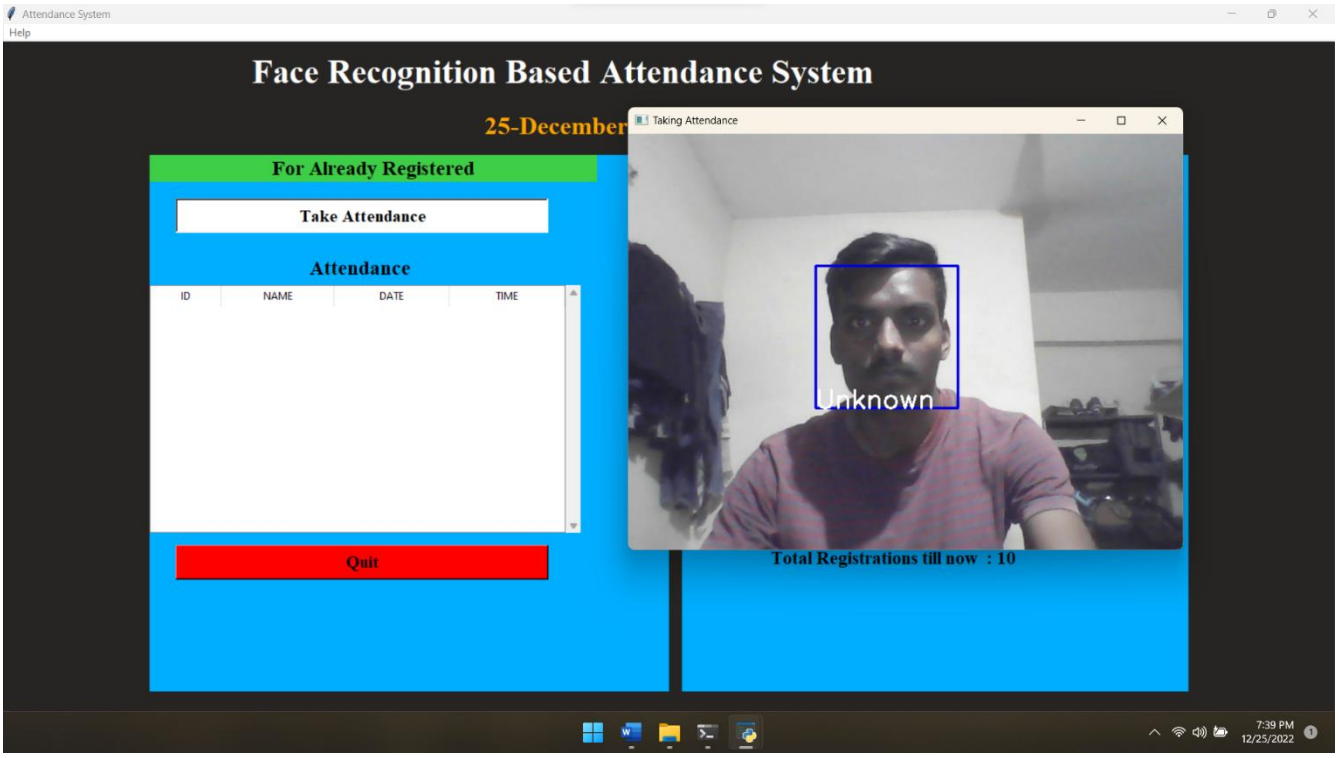
6.4.1 Test cases

7. RESULTS

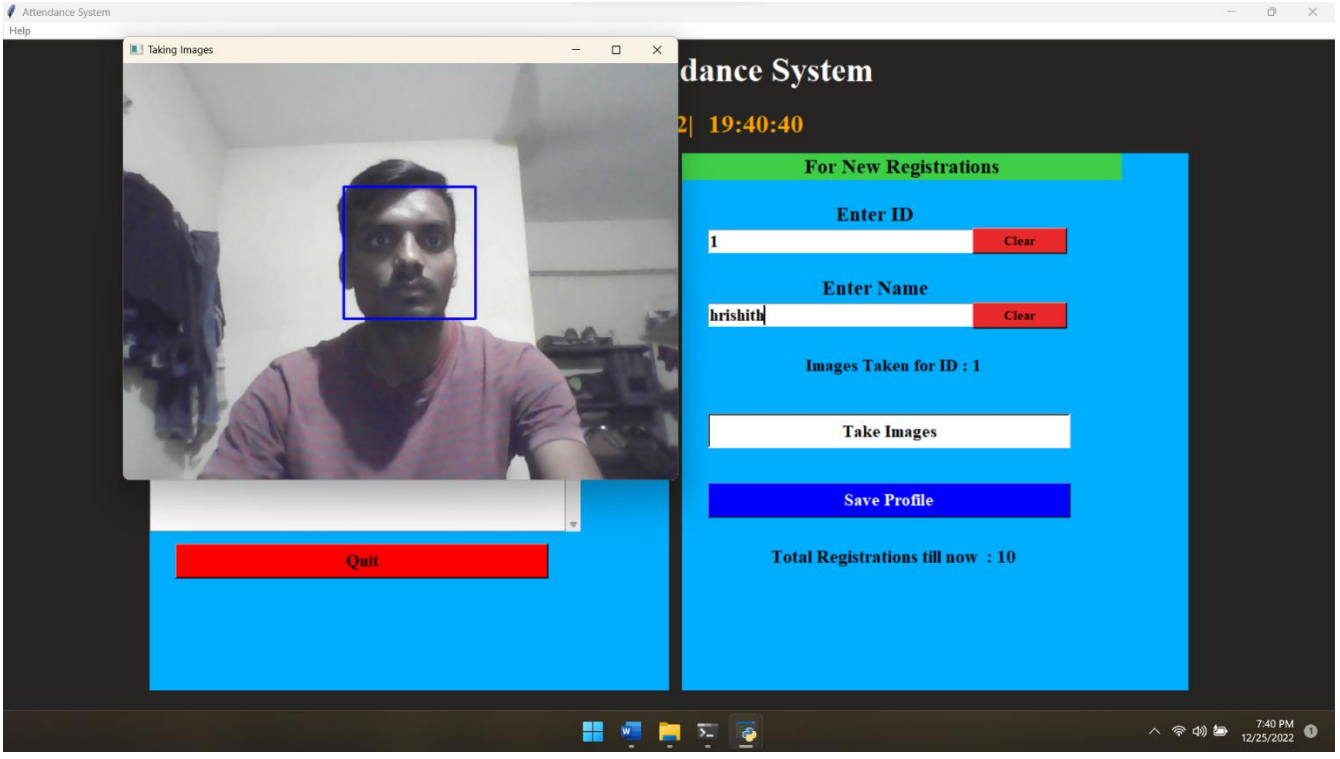
As this was a small-scale project, data structure and implementation did not have many problems. However, it took the author many effort with research and study with different technologies needed as these tools and technologies were new to the author. This caused a delay in the development of the project. Despite the delay and difficulties, the author was able to incorporate those tools and technologies and complete the project. However, the success rate of facial recognition was not as expected. The success rate depended upon the quality of the camera, lighting, and sufficient dataset in the database. When these factors were to be managed properly, the success rate of face recognition increased.



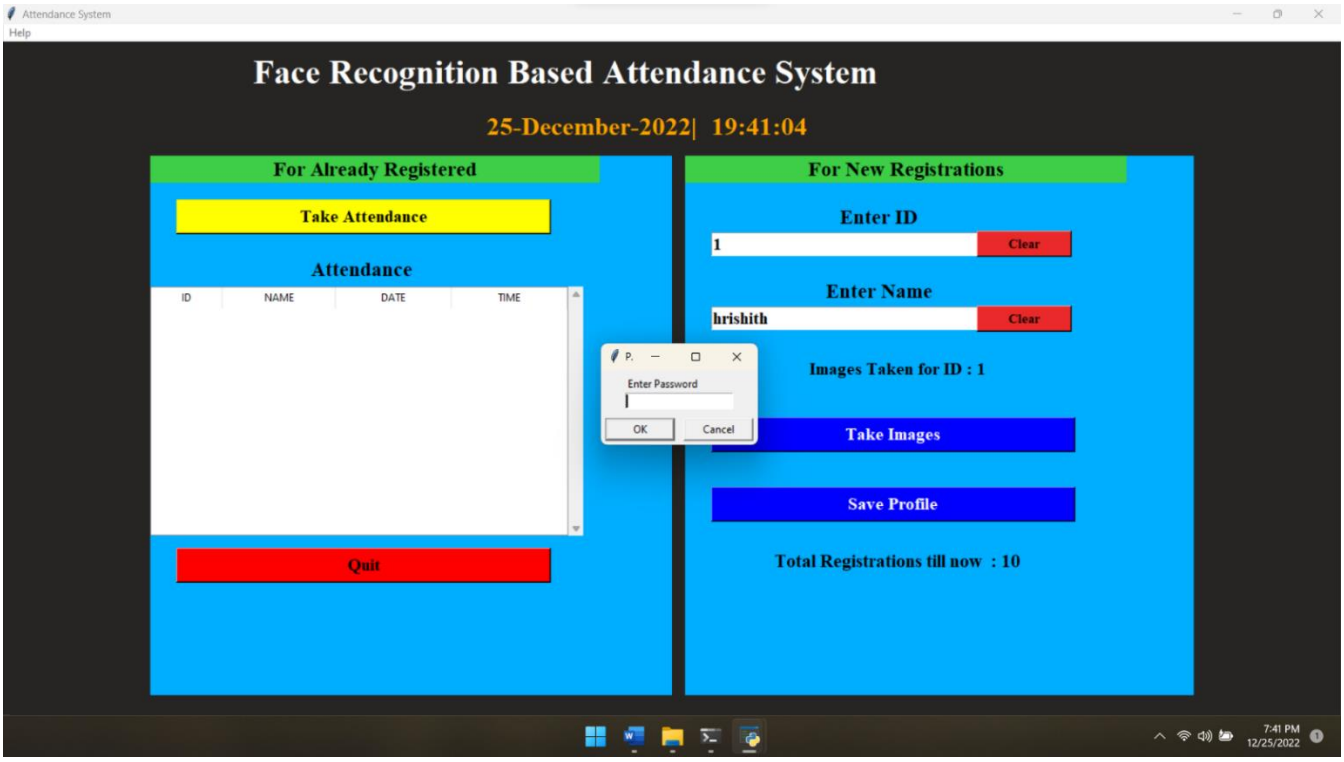
7.1 Attendance before updating



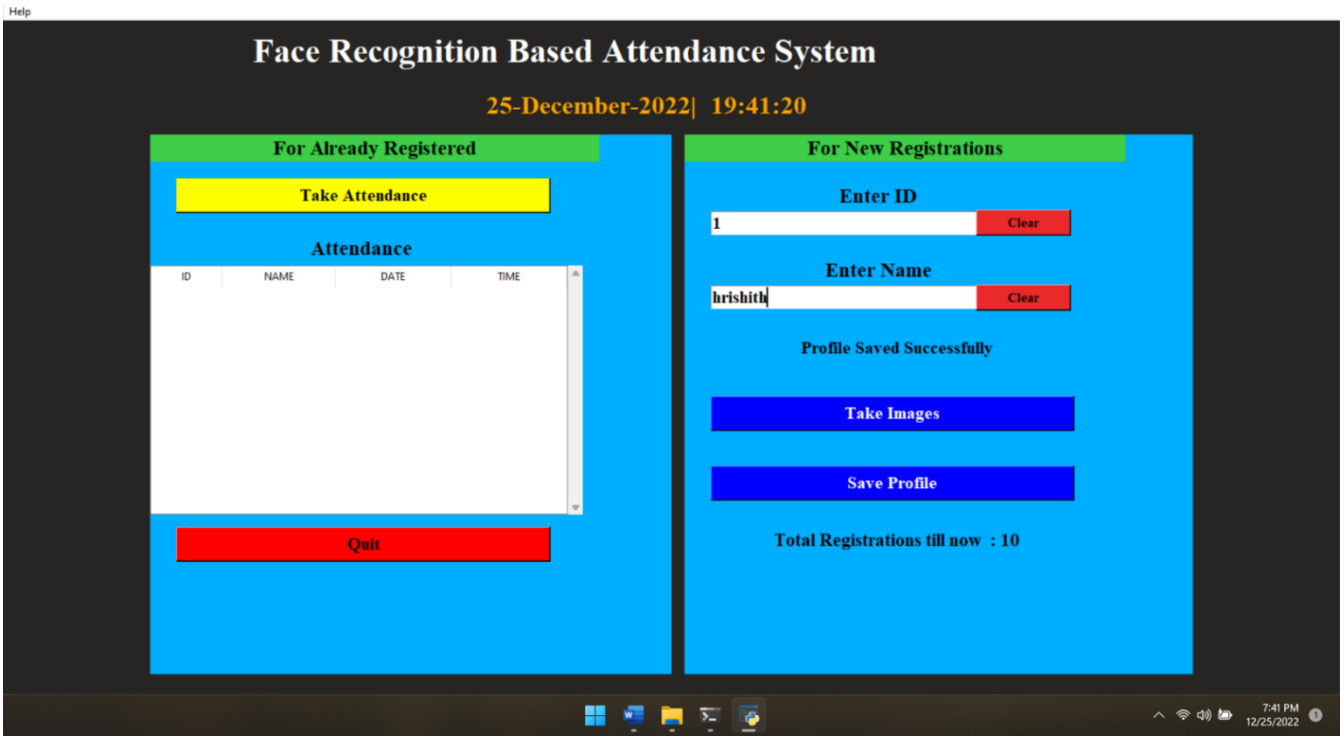
7.2 Candidate trying to take attendance before registering



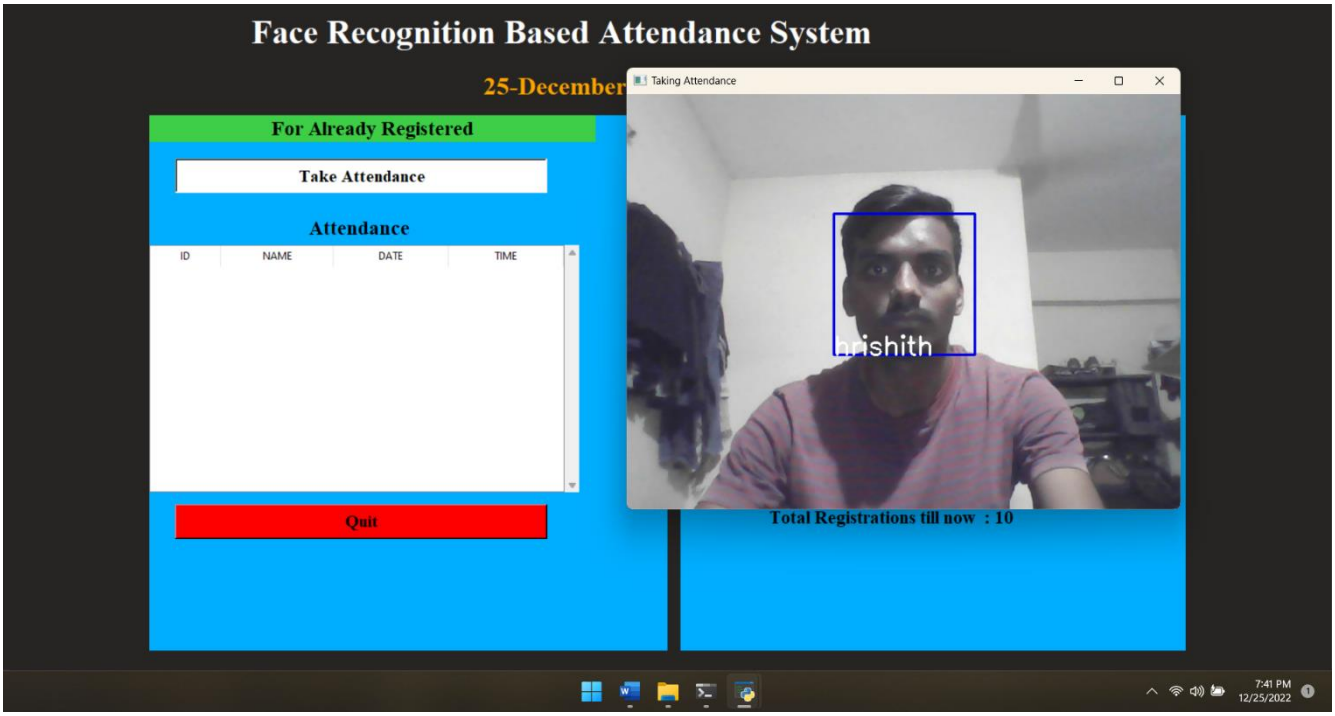
7.3 Registering the candidate



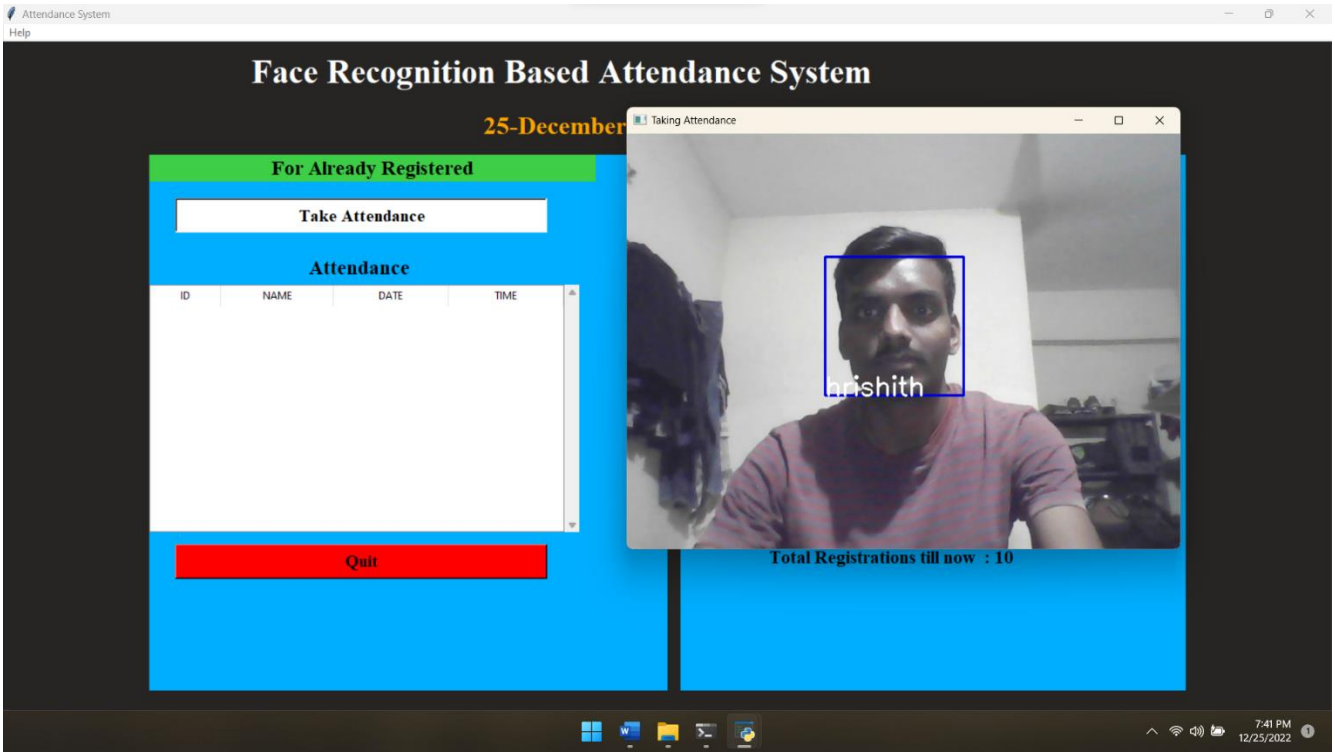
7.4 Password secured registration



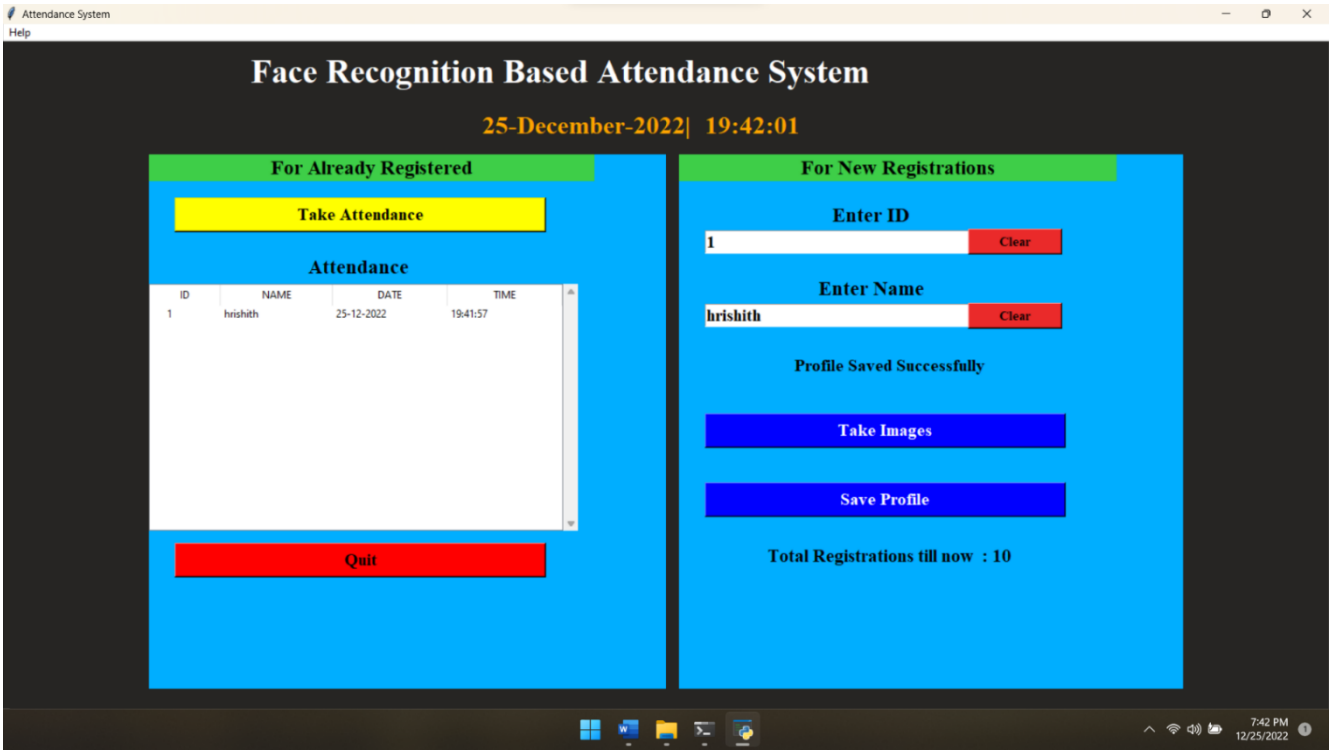
7.5 Profile saved successfully



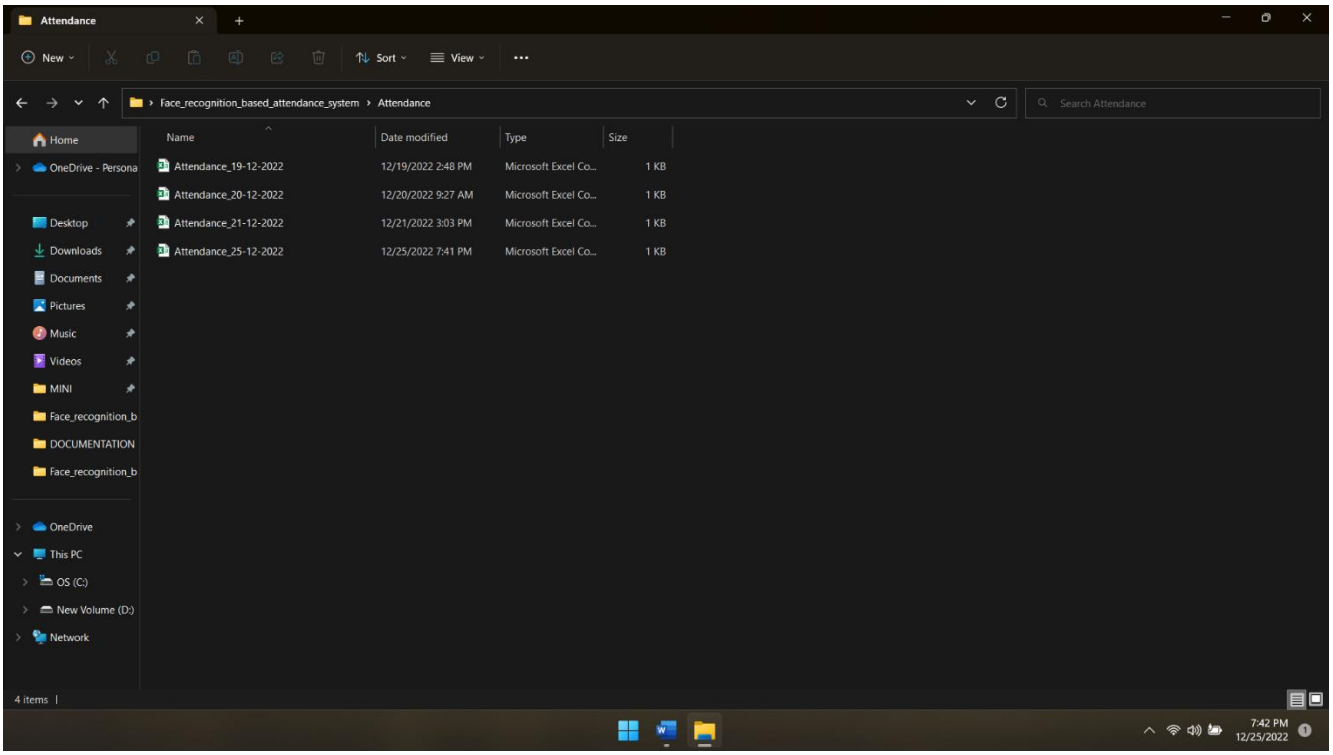
7.6 Candidate taking attendance



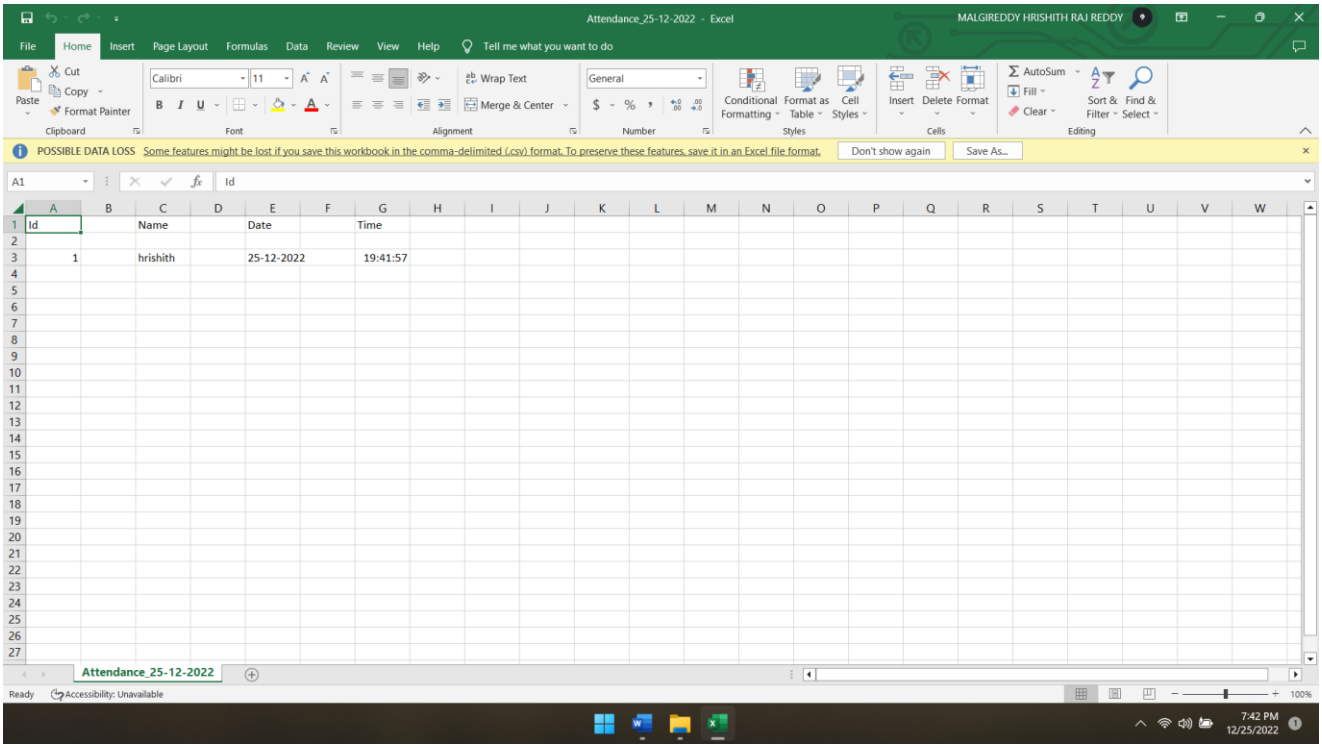
7.7 Candidate detected



7.8 Attendance registered



7.9 Attendance after updated



7.10 Attendance marked in excel sheet

8. CONCLUSION AND FUTURE SCOPE

8.1 CONCLUSION

The proposed method uses face detection and face recognition that helps to maintain the automated attendance system. For detection, Paul–Viola Jones algorithm is used and for face recognition Linear Binary Pattern Histogram (LPBH) algorithm is applied. Lower is the distance, higher is the recognition rate. The aim of this project is to capture the video of the students, 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 Classroom Attendance System helps in increasing the accuracy and speed ultimately achieve the high-precision real-time attendance to meet the need for automatic classroom evaluation.

8.2 FUTURE SCOPE

The future scope of the project can be integrated with the hardware components for example GSM through which a monthly list of the defaulter students can be sent to the mentor. Additionally, an application can be developed to help students to maintain a track of their attendance. It can also be used in offices where a large group of employees sit in a hall and their attendance will be marked automatically by capturing a video but for this the accuracy of the recognition needs to be improved.

9. REFERENCES

[1] Michael Dobson, Douglas Ahlers, Bernie DiDario, <Attendance Tracking System", United States Patent Application Publication, Pub. No.: US 2006/0035205 A1, Feb.16, 2006.

[2] Naveed Khan Balcoh, M. HaroonYousaf, Waqar Ahmad and M. IramBaig, <Algorithm for Efficient Attendance Management: Face Recognition based approach=, IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 4, No 1, pp.146-150, July 2012.

[3] O. Shoewu and O.A. Idowu, <Development of Attendance Management System using Biometrics ", The Pacific Journal of Science and Technology, Vol. 13, Number1, pp.300-307, May 2012 (Spring).

[4] Damir Demirovic, Emir Skejic , Amira Serfovic, <Performance of some images processing algorithms in TensorFlow=, computer vision and pattern recognition, pp, 799-778, 2018.

[5] http://www.openCV.org

[6] Hapani, Smit, et al. "Automated Attendance System Using Image Processing." 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA). IEEE, 2018.

[7] S. Bahrampour, N. Ramakrishnan, L. Schott. And M. Shah, M. Comparative Study of Caffe, Neon, Theano, and Torch for Deep Learning<. CoRR, abs/1511.06435. 2015.

[8] I. K. Park, N. Singhal, M. H. Lee, S. Cho and C. Kim, "Design and PerformanceEvaluation of Image Processing Algorithms on GPUs," in IEEE Transactions on Parallel and Distributed Systems, vol. 22, no. 1, pp. 91-104, Jan. 2011.