1. INTRODUCTION

Maintaining the attendance is very important in all the institutes for checking the performance of students. Every institute has its own method in this regard. Some are taking attendance manually using the old paper or file-based approach and some have adopted methods of automatic attendance using some biometric techniques. But in these methods students have to wait for long time in making a queue at time they enter the class. Many biometric systems are available but the key authentications are same is all the techniques. Every biometric system consists of enrolment process in which unique features of a person is stored in the database and then there are processes of identification and verification. These two processes compare the biometric feature of a person with previously stored template captured at the time of enrollment. Biometric templates can be of many types like Fingerprints, Eye Iris, Face, Hand Geometry, Signature, Gait and voice. Our system uses the face recognition approach for the automatic attendance of students. Face recognition consists of two steps, in first step faces are detected in the image and then these detected faces are compared with the database for verification. A number of methods have been proposed for face detection i.e. Ada Boost algorithm, the Float Boost algorithm, the S-Ada Boost algorithm Support Vector Machines (SVM), and the Bayes classifier. The efficiency of face recognition algorithm can be increased with the fast face detection algorithm. In all the above methods Histogram of the Oriented Gradient (HOG) is most efficient. Our system utilized this algorithm for the detection of faces. Face recognition techniques can be Divided into two types Appearance based which use texture features that is applied to whole face or some specific Regions, other is Feature based which uses geometric features like mouth, nose, eyes, eye brows, cheeks and Relation between them. Statistical tools such as Linear Discriminant Analysis (LDA), Principal Component Analysis (PCA), Kernel Methods, and Neural Networks, Eigen-faces have been used for construction of face templates.

1.1 PROBLEM DEFINITION

Traditional manual methods of monitoring student attendance in lectures are tedious as the signed attendance sheets have to be manually logged in to a computer system for analysis. This is tedious, time consuming and prone to inaccuracies as some students in the department often sign for their absent colleagues, rendering this method ineffective in tracking the students' class attendance. Use of the face detection and recognition system will provide a fast and effective method of capturing student attendance accurately while offering a secure, stable and robust storage of the system records.

1.3 SCOPE

System comprising is mainly of two modules. The first module is a face- detection, which is basically uses camera application that captures student faces and stores them in a file using computer vision face detection algorithms and face extraction techniques. The second module is face-recognition of the captured images (faces) in the file, marks the students register and then stores the results in a database for future analysis.

1.4 MOTIVATION

Motivation is mainly focused to solve the slow and inefficient traditional manual attendance system. This made us to think why not make it automated fast and mush efficient. Also, such face detection techniques are in use by department like crime investigation where they use CCTV footages and detect the faces from the crime scene and compare those with criminal database to recognize them. Also, Facebook, it uses an algorithm called deep face whose accuracy to recognize is 97.25% which is as close as what humans have that is 97.53%.

1.5 SYSTEM ARCHITECTURE

1.5.1 REGISTER STUDENT

The student face image will be captured, generate the face encodings and store those face encodings in the database for future reference.

1.5.2 TAKING ATTENDANCE

The student will be given attendance by capturing real time face of student, detect face in the captured image, generate face encodings of current face image, compare generated face encodings with the stored face encodings, getting roll number of matched student face encodings and given attendance to that roll number.

1.5.3 GENERATING REPORTS

The required report will be generated by getting the stored reports, performing relevant mathematical operations on the stored reports and get relevant information.

1.5.4 GENERATING GRAPHS

The required graph will be generated by getting the stored reports and plot the required graph on a new window.

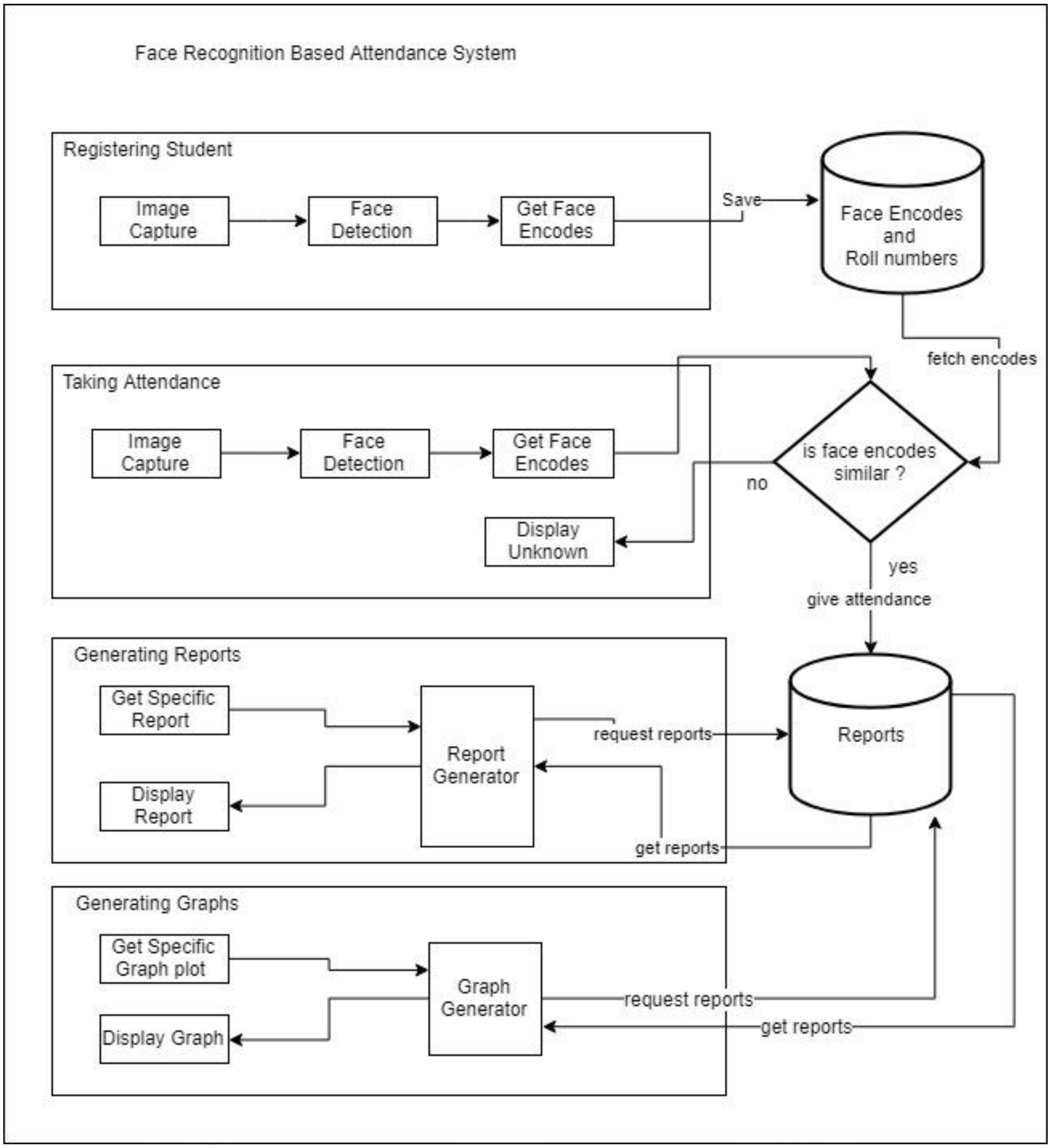


Fig 1.1: System Architecture

2. LITERATURE SURVEY

Literature survey helps to provide solutions with an easy way and it also provides some knowledge regarding the technology.

2.1 INTRODUCTION

Face detection is the process of identifying and locating all the present faces in a single image or video regardless of their position, scale, orientation, age and expression. Furthermore, the detection should be irrespective of extraneous illumination conditions and the image and video content.

Face Recognition is a visual pattern recognition problem, where the face, represented as a three-dimensional object that is subject to varying illumination, pose and other factors, needs to be identified based on acquired images. Face Recognition is therefore simply the task of identifying an already detected face as a known or unknown face and in more advanced cases telling exactly whose face it is.

Face detection answers the question, where is the face? It identifies an object as a “face” and locates it in the input image. Face Recognition on the other hand answers the question who is this? Or whose face is it? It decides if the detected face is someone known or unknown based on the database of faces it uses to validate this input image. It can therefore be seen that face detections output (the detected face) is the input to the face recognizer and the face Recognition’s output is the final decision i.e. face known or face unknown

2.2 MACHINE LEARNING

Machine learning is the idea that there are generic algorithms that can tell you something interesting about a set of data without you having to write any custom code specific to the problem. Instead of writing code, you feed data to the generic algorithm and it builds its own logic based on the data. For example, one kind of algorithm is a classification algorithm. It can put data into different groups. The same classification algorithm used to recognize handwritten numbers could also be used to classify emails into spam and not-spam without changing a line of code. It's the same algorithm but it's fed different training data so it comes up with different classification logic.

"Machine learning" is an umbrella term covering lots of these kinds of generic algorithms.

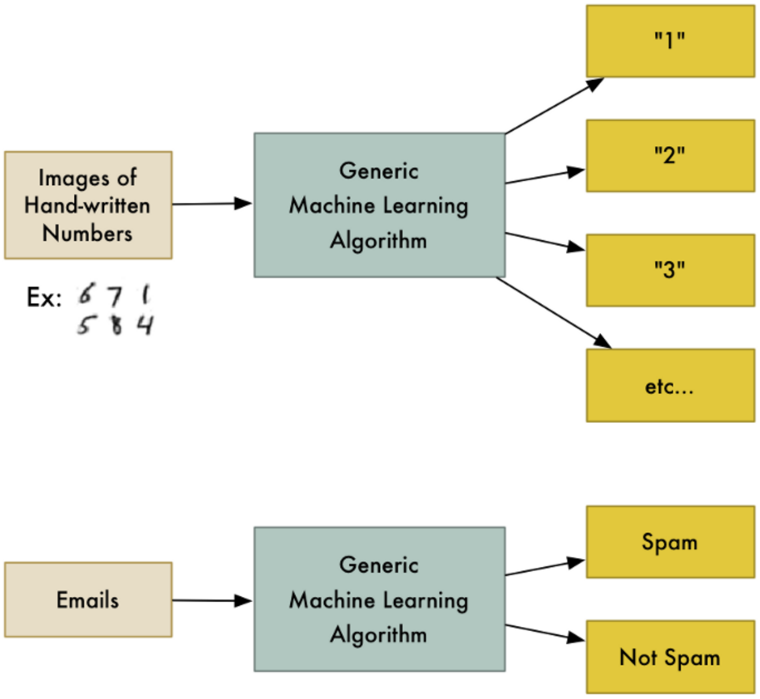

Images of Hand-written Numbers
Ex: 6 7 1 5 8 4
Generic Machine Learning Algorithm
"1"
"2"
"3"
etc...
Emails
Generic Machine Learning Algorithm
Spam
Not Spam

Fig 2.1: Example of Classification Machine Learning Algorithm

Machine learning algorithms are falling into one of three main categories **Supervised learning**, **Unsupervised learning** and **Semi-supervised learning**.

2.2.1 SUPERVISED LEARNING

The majority of practical machine learning uses supervised learning. Supervised learning is where you have input variables (x) and an output variable (Y) and you use an algorithm to learn the mapping function from the input to the output. The model can be an equation Y = f(X). The goal is to approximate the mapping function so well that when you have new input data (x) that you can predict the output variables (Y) for that data.

It is called supervised learning because the process of an algorithm learning from the training dataset can be thought of as a teacher supervising the learning process. We know the correct answers; the algorithm iteratively makes predictions on the training data and is corrected by the teacher. Learning stops when the algorithm achieves an acceptable level of performance.

2.2.2 UNSUPERVISED MACHINE LEARNING

Unsupervised learning is where you only have input data (X) and no corresponding output variables. The goal for unsupervised learning is to model the underlying structure or distribution in the data in order to learn more about the data. These are called unsupervised learning because unlike supervised learning above there is no correct answers and there is no teacher. Algorithms are left to their own devises to discover and present the interesting structure in the data. Unsupervised learning problems can be further grouped into clustering and association problems.

- **Clustering**: A clustering problem is where you want to discover the inherent groupings in the data, such as grouping customers by purchasing behaviour.
- **Association**: An association rule learning problem is where you want to discover rules that describe large portions of your data, such as people that buy X also tend to buy Y.

Some popular examples of unsupervised learning algorithms are:

- k-means for clustering problems.
- Apriori algorithm for association rule learning problems.

2.2.3 SEMI-SUPERVISED MACHINE LEARNING

Problems where you have a large amount of input data (X) and only some of the data is labelled (Y) are called semi-supervised learning problems. These problems sit in between both supervised and unsupervised learning. A good example is a photo archive where only some of the images are labelled, (e.g. dog, cat, person) and the majority are unlabelled. Many real-world machine learning problems fall into this area. This is because it can be expensive or time-consuming to label data as it may require access to domain experts. Whereas unlabelled data is cheap and easy to collect and store.

We can use unsupervised learning techniques to discover and learn the structure in the input variables. We can also use supervised learning techniques to make best guess predictions for the unlabelled data, feed that data back into the supervised learning algorithm as training data and use the model to make predictions on new unseen data.

2.3 FACE DETECTION

Face detection went mainstream in the early 2000's when Paul Viola and Michael Jones invented a way to detect faces that was fast enough to run on cheap cameras. However, much more reliable solutions exist now. We're going to use a method invented in 2005 called Histogram of Oriented Gradients—or just **HOG** for short.

To find faces in an image, we'll start by making our image black and white because we don't need colour data to find faces.

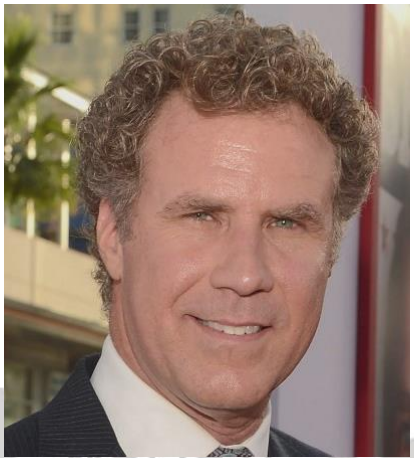


Fig 2.2: sample image

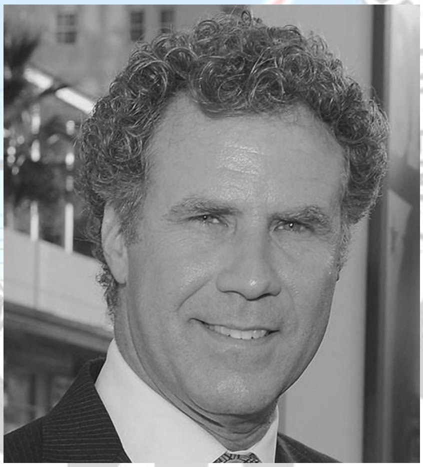


Fig 2.3: grayscale version of sample image

Then we'll look at every single pixel in our image one at a time. For every single pixel, we want to look at the pixels that directly surrounding it.

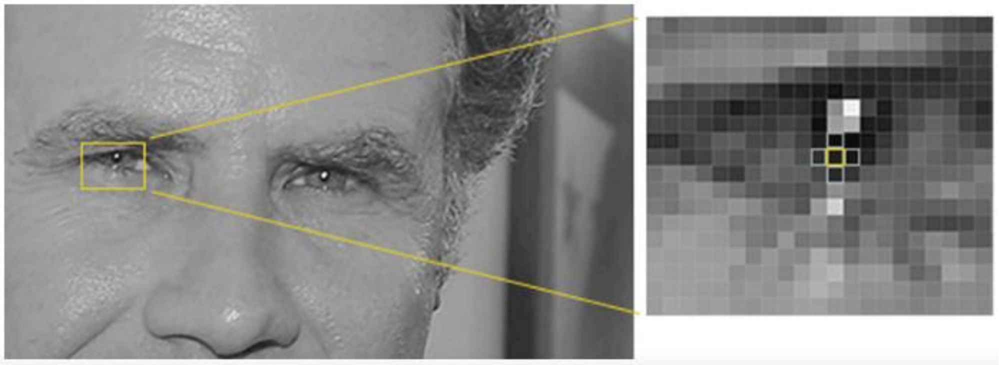


Fig 2.4: looking at surrounding pixels for each pixel in the image

Our goal is to figure out how dark the current pixel is compared to the pixels directly surrounding it. Then we want to draw an arrow showing in which direction the image is getting darker.

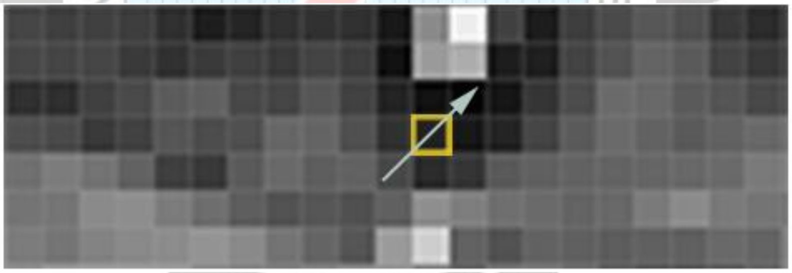


Fig 2.5: Looking at just this one pixel and the pixels touching it, the image is getting darker towards the upper right

If we repeat that process for every single pixel in the image, you end up with every pixel being replaced by an arrow. These arrows are called gradients and they show the flow from light to dark across the entire image.

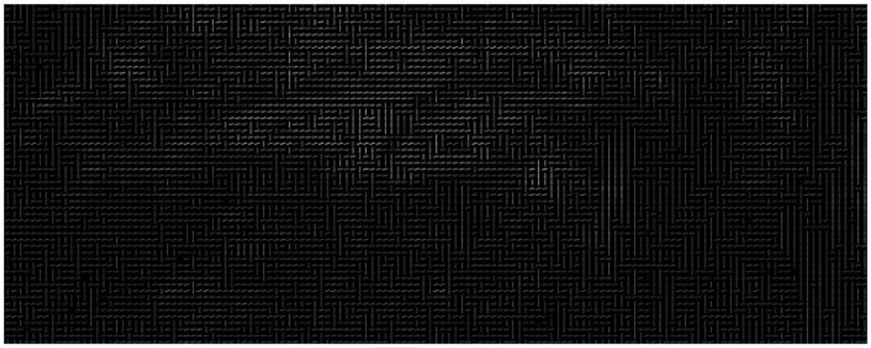


Fig 2.6: Drawing arrows for eye part in image

This might seem like a random thing to do, but there's a really good reason for replacing the pixels with gradients. If we analyse pixels directly, really dark images and really light images of the same person will have totally different pixel values. But by only considering the direction that brightness changes, both really dark images and really bright images will end up with the same exact representation. That makes the problem a lot easier to solve.

But saving the gradient for every single pixel gives us way too much detail. We end up missing the forest for the trees. It would be better if we could just see the basic flow of lightness/darkness at a higher level so we could see the basic pattern of the image.

To do this, we'll break up the image into small squares of 16x16 pixels each. In each square, we'll count up how many gradients point in each major direction (how many points up, point up-right, point right, etc…). Then we'll replace that square in the image with the arrow directions that were the strongest.

The end result is we turn the original image into a very simple representation that captures the basic structure of a face in a simple way.

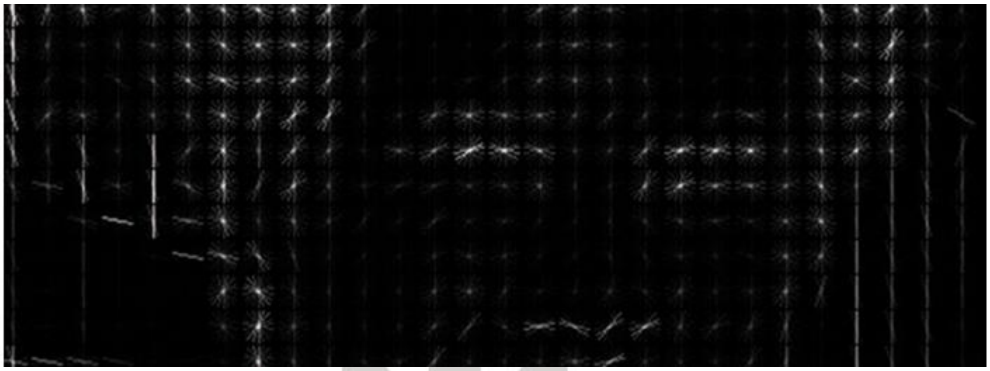


Fig 2.7: The original image is turned into a HOG representation that captures the major features of the image regardless of image brightness.

To find faces in this HOG image, all we have to do is find the part of our image that looks the most similar to a known HOG pattern that was extracted from a bunch of other training faces.

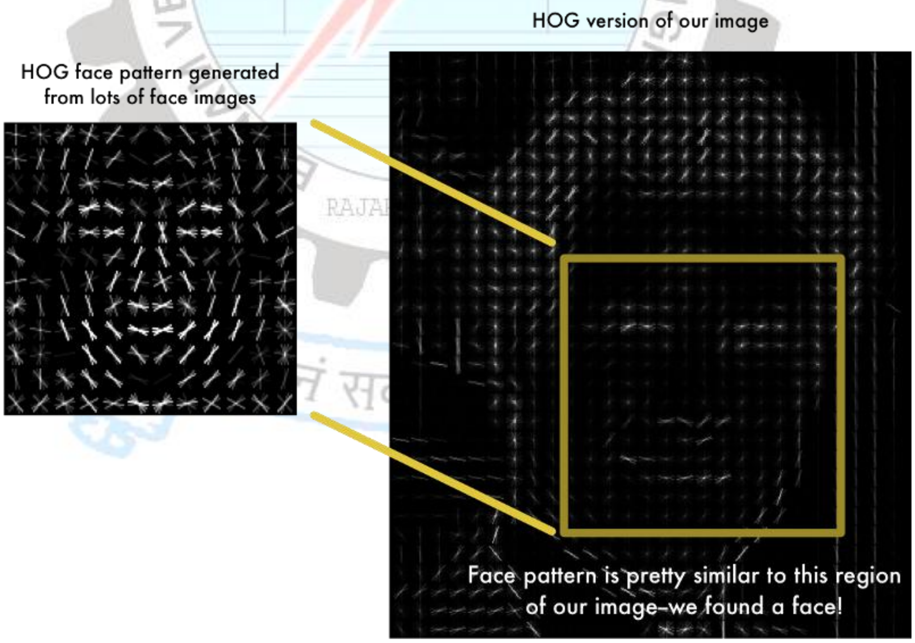

HOG version of our image
HOG face pattern generated from lots of face images
Face pattern is pretty similar to this region of our image–we found a face!

Fig 2.8: Finding face pattern for sample image

Using Histogram of Oriented Gradients (HOG) technique, we can now easily find faces in any image.

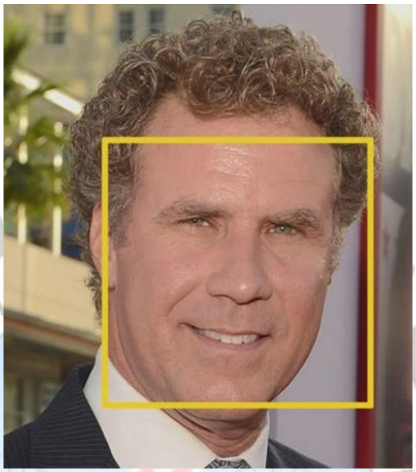


Fig 2.9: Finding face in sample image and draws rectangle around face

2.4 POSING AND ROJECTING FACES

When, we isolated the faces in our image. But now we have to deal with the problem that faces turned different directions look totally different to a computer:

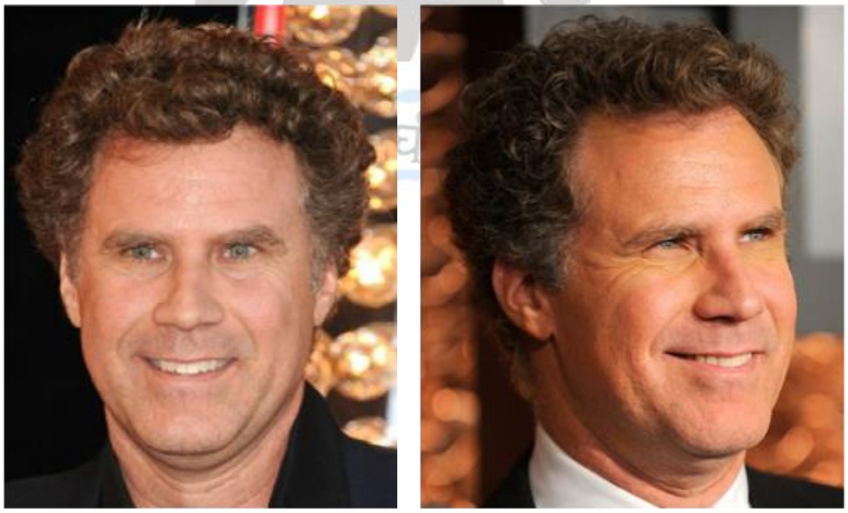


Fig 2.10 Same person face image with different angles

To account for this, we will try to warp each picture so that the eyes and lips are always in the sample place in the image. This will make it a lot easier for us to compare faces.

To do this, we are going to use an algorithm called **face landmark estimation**. There are lots of ways to do this, but we are going to use the approach invented in 2014 by Vahid Kazemi and Josephine Sullivan.

The basic idea is we will come up with 68 specific points (called landmarks) that exist on every face—the top of the chin, the outside edge of each eye, the inner edge of each eyebrow, etc. Then we will train a machine learning algorithm to be able to find these 68 specific points on any face:

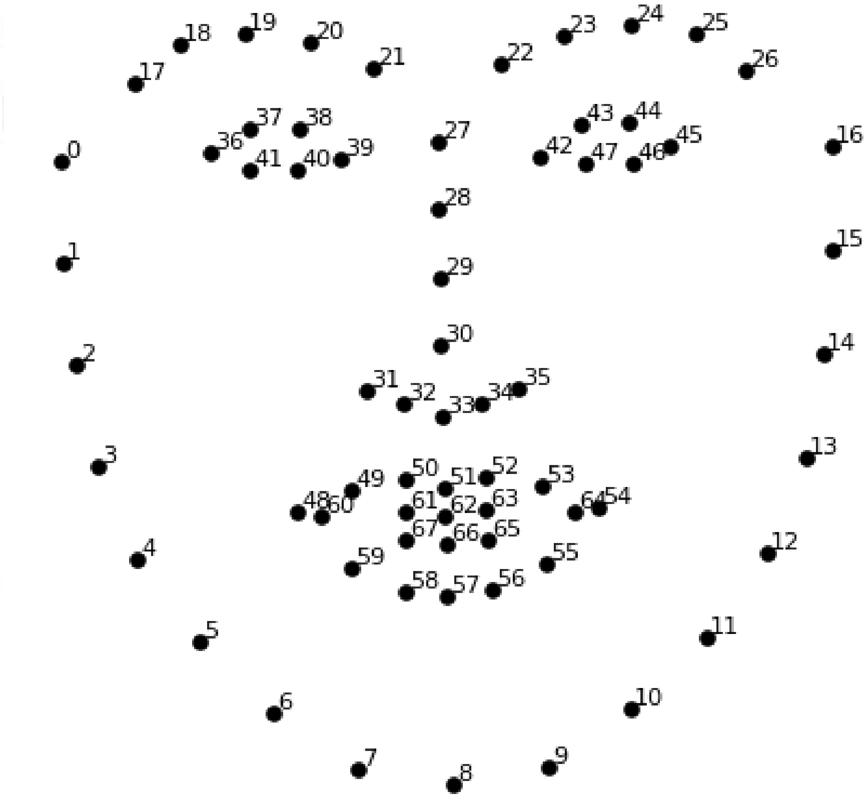


Fig 2.11: The 68 landmarks we will locate on every face, Image created by Brandon Amos of CMU who works on OpenFace.

Here’s the result of locating the 68 face landmarks on our test image

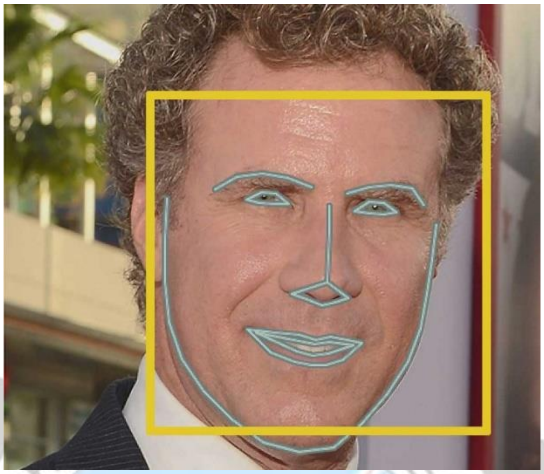


Fig 2.12: Sample image with showing face landmarks

Now that we know where the eyes and mouth are, we’ll simply rotate, scale and shear the image so that the eyes and mouth are centred as best as possible. We won’t do any fancy 3d warps because that would introduce distortions into the image. We are only going to use basic image transformations like rotation and scale that preserve parallel lines (called affine transformations)

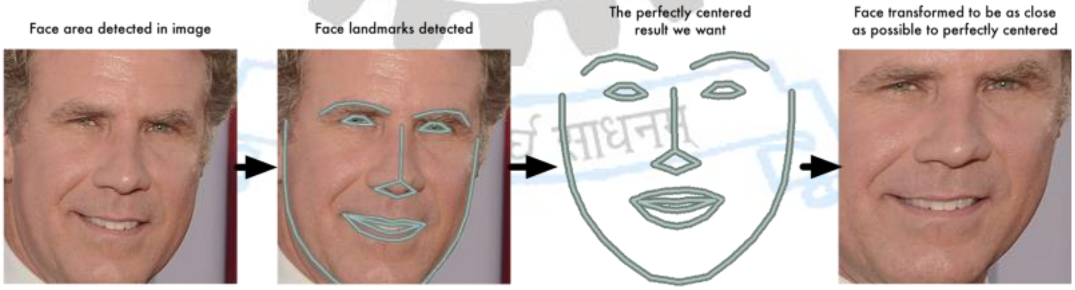

Face area detected in image
Face landmarks detected
The perfectly centered result we want
Face transformed to be as close as possible to perfectly centered

Fig 2.13: Sample image face transformation to centre face

Now no matter how the face is turned, we are able to center the eyes and mouth are in roughly the same position in the image.

2.5 FACE RECOGNITION

2.5.1 ENCODING FACES

It turns out that the measurements that seem obvious to us humans (like eye colour) don’t really make sense to a computer looking at individual pixels in an image. Researchers have discovered that the most accurate approach is to let the computer figure out the measurements to collect itself. Deep learning does a better job than humans at figuring out which parts of a face are important to measure.

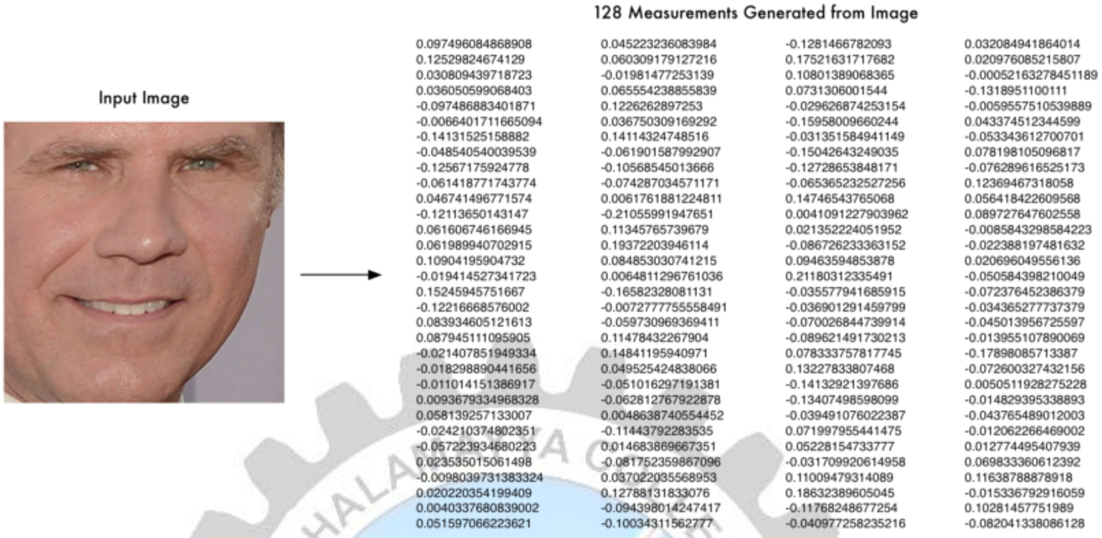
The solution is to use a trained Deep Convolutional Neural Network. The model is trained, to generate 128 measurements for each face.

Machine learning people call the 128 measurements of each face an **embedding**. The idea of reducing complicated raw data like a picture into a list of computer-generated numbers comes up a lot in machine learning (especially in language translation). The exact approach for faces we are using was invented in 2015 by researchers at Google but many similar approaches exist.

This process of training a convolutional neural network to output face embeddings requires a lot of data and computer power. Even with an expensive NVidia Telsa video card, it takes about 24 hours of continuous training to get good accuracy.

But once the network has been trained, it can generate measurements for any face, even ones it has never seen before. So, this step only needs to be done once. Lucky for us, the fine at OpenFace already did this and they published several trained networks which we can directly use.

So, all we need to do ourselves is run our face images through their pre-trained network to get the 128 measurements for each face. Here’s the measurements for our test image.



0.097496084868908	0.045223236083984	-0.1281466782093	0.032084941864014
0.12529824674129	0.060309179127216	0.17521631717682	0.020976085215807
0.030809439718723	-0.01981477253139	0.10801389068365	-0.00052163278451189
0.036050599068403	0.065554238855839	0.0731306001544	-0.1318951100111
-0.097486883401871	0.1226262897253	-0.029626874253154	-0.0059557510539889
-0.0066401711665094	0.036750309169292	-0.15958009660244	0.043374512344599
-0.14131525158882	0.14114324748516	-0.031351584941149	-0.053343612700701
-0.048540540039539	-0.061901587992907	-0.15042643249035	0.078198105096817
-0.12567175924778	-0.10568545013666	-0.12728653848171	-0.076289616525173
-0.061418771743774	-0.074287034571171	-0.065365232527256	0.12369467318058
0.046741496771574	0.0061761881224811	0.14746543765068	0.056418422609568
-0.12113650143147	-0.21055991947651	0.0041091227903962	0.089727647602558
0.061606746166945	0.11345765739679	0.021352224051952	-0.0085843298584223
0.061989940702915	0.19372203946114	-0.086726233363152	-0.022388197481632
0.10904195904732	0.084853030741215	0.09463594853878	0.020696049556136
-0.019414527341723	0.0064811296761036	0.21180312335491	-0.050584398210049
0.15245945751667	-0.16582328081131	-0.035577941685915	-0.072376452386379
-0.12216668576002	-0.0072777755558491	-0.036901291459799	-0.034365277737379
0.083934605121613	-0.059730969369411	-0.070026844739914	-0.045013956725597
0.087945111095905	0.11478432267904	-0.089621491730213	-0.013955107890069
-0.021407851949334	0.14841195940971	0.078333757817745	-0.17898085713387
-0.018298890441656	0.049525424838066	0.13227833807468	-0.072600327432156
-0.011014151386917	-0.051016297191381	-0.14132921397686	0.0050511928275228
0.0093679334968328	-0.062812767922878	-0.13407498598099	-0.014829395338893
0.058139257133007	0.0048638740554452	-0.039491076022387	-0.043765489012003
-0.024210374802351	-0.11443792283535	0.071997955441475	-0.012062266469002
-0.057223934680223	0.014683869667351	0.05228154733777	0.012774495407939
0.023535015061498	-0.081752359867096	-0.031709920614958	0.069833360612392
-0.0098039731383324	0.037022035568953	0.11009479314089	0.11638788878918
0.020220354199409	0.12788131833076	0.18632389605045	-0.015336792916059
0.0040337680839002	-0.094398014247417	-0.11768248677254	0.10281457751989
0.051597066223621	-0.10034311562777	-0.040977258235216	-0.082041338086128

Fig 2.14: 128 Face Measurements

So, what parts of the face are these 128 numbers measuring exactly? It turns out that we have no idea. It doesn’t really matter to us. All that we care is that the network generates nearly the same numbers when looking at two different pictures of the same person.

2.5.2 FINDING THE PERSON’S FACE ENCODINGS

This step is actually the easiest step in the whole process. All we have to do is find the person in our database of known people who has the closest measurements to our test image.

We can do that by using any basic machine learning classification algorithm. No fancy deep learning tricks are needed. We’ll use a simple linear SVM classifier, but lots of classification algorithms could work.

All we need to do is train a classifier that can take in the measurements from a new test image and tells which known person is the closest match. Running this classifier takes milliseconds. The result of the classifier is the name of the person.

2.6 PYTHON

Python is an interpreted, high-level, general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace. It provides constructs that enable clear programming on both small and large scales. Van Rossum led the language community until stepping down as leader in July 2018.

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural. It also has a comprehensive standard library. The following diagram will illustrate how python program execution works.

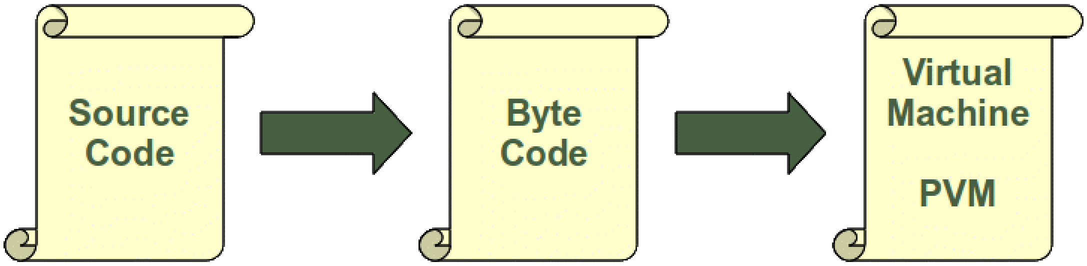


Fig 2.15: Python program execution process

Python interpreters are available for many operating systems. CPython, the reference implementation of Python, is source software and has a community-based development model, as do nearly all of Python's other implementations. Python and CPython are managed by the non-profit Python Software Foundation.The language's core philosophy is summarized as:

- Beautiful is better than ugly
- Explicit is better than implicit
- Simple is better than complex

- Complex is better than complicated
- Readability counts

2.6.1 DATA SCIENCE AND MACHINE LEARNING WITH PYTHON

Sophisticated data analysis has become one of fastest-moving areas of IT and one of Python's star use cases. The vast majority of the libraries used for data science or machine learning have Python interfaces, making the language the most popular high-level command interface to for machine learning libraries and other numerical algorithms.

MATPLOTLIB

Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like Tkinter, wxPython, Qt, or GTK+. There is also a procedural "pylab" interface based on a state machine (like OpenGL), designed to closely resemble that of MATLAB, though its use is discouraged. SciPy makes use of Matplotlib.

Ex: import matplotlib.pyplot

PANDAS

Pandas is a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series.

Ex: import pandas

OPENCV

OpenCV (Open source computer vision) is a library of programming functions mainly aimed at real-time computer vision. Originally developed by Intel, it was later supported by Willow Garage then Itseez (which was later acquired by Intel). The library is cross-platform and free for use under the open-source BSD license.

Ex: import cv2

2.7 ATTENDANCE SYSTEM BASED ON FACE RECOGNITION USING EIGEN FACE AND PCA ALGORITHMS

Authors - Priyanka Wagh, Roshani Thakare, Jagruti Chaudhari, Shweta Patil.

International Conference on Green Computing and Internet of Things (ICGCIoT).

The attendance maintaining system is difficult process if it is done manually. The smart and automated attendance system for managing the attendance can be implemented using the various ways of biometrics. Face recognition is one of them. By using this system, the issue of fake attendance and proxies can be solved. In the previous face recognition-based attendance system, there were some disadvantages like intensity of light problem and head pose problem. Therefore, to overcome these issues, various techniques like illumination invariant, Viola and Jones algorithm, Principle component analysis are used.

2.8 A FACE RECOGNITION ATTENDANCE SYSTEM WITH GSM NOTIFICATION

Authors - Kennedy Okokpujie, Etinosa Noma-Osaghae, Samuel John.

IEEE 3rd International Conference on Electro-Technology for National Development

Current biometric methods for attendance are too intrusive. This paper presents a stress-free non-intrusive way of taking class attendance using face as the biometric It also has the added novelty of relaying vital information about class attendance to handheld devices via any available cellular network. During enrollment, a camera was used to acquire facial images that were made into templates using Fisherfaces algorithm. These templates were stored in a database.

3. SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

3.1.1 FINGER PRINT BASED ATTENDANCE SYSTEM

In The fingerprint system may not be able to register some users, as the fingerprint image does not contain sufficient fingerprint patterns for good quality verification**.**

The fingerprint system may register some users but when they try to verify their finger, it fails or false matches with other’s finger. At the time of registration, the fingerprint scanner may have registered the low-quality image but later it does not match with the finger images captured at the time of verification.

The fingerprint system may register and verify the fingerprint but after few days it start failing. This is due to regular changes in the finger skin due to dry finger problem

Disadvantages

1. The system takes a lot of time to take attendance.
2. The system very unhygienic because, everyone is actually sharing germs via finger print scanner device
3. Environment and usage can affect the finger print scanning device

3.2 PROPOSED SYSTEM

Face Recognition based attendance system offer several advantages over other biometric methods, almost all these technologies require some voluntary action by the user, i.e., the user needs to place his hand on a hand-rest for fingerprinting or hand geometry detection and has to stand in a fixed position in front of a camera for iris or retina identification. However, face recognition can be done passively without any explicit action or participation on the part of the user since face images can be acquired from a distance by a camera.

Advantages

1. Time Consumption to take attendance is very less compared to other system. Entire class attendance can be taken at once
2. The system is fully automated, attendance will be given automatically if the face is captured and recognized.
3. The system is very hygienic compared to finger print based attendance system

3.3 SYSTEM REQUIREMENT ANALYSIS

3.3.1 FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

- Economical feasibility
- Technical feasibility
- Social feasibility

ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus, the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement; as only minimal or null changes are required for implementing this system.

TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement; as only minimal or null changes are required for implementing this system.

SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

3.4 HARDWARE REQUIRMENTS

Minimum hardware requirements are

➢ Processor	- 1.8 Ghz Dual Core Intel Pentium/AMD Athlon 64 X2
➢ RAM	- 2 GB
➢ Camera	- 8 Mega-Pixel
➢ Hard Disk	- 160 GB

3.5 SOFTWARE REQUIRMENTS

Minimum software requirements are

➢ Operating System	- Windows 7 or Linux or Mac
➢ Coding Language	- Python 3.6.8
➢ Front End (GUI)	- Python (TkInter)
➢ Back End	- Excel (CSV)

4. SYSTEM DESIGN

Design involves identification of classes, their relationships as well as their collaboration. In objector, classes are divided into entity classes, interface classes and control classes.

4.1 DESIGN OVERVIEW

The Computer Aided Software Engineering (CASE) tools that are available commercially do not provide any assistance in this transition. CASE tools take advantage of meta modeling that are helpful only after the construction of the class diagram. In the Fusion method, some object-oriented approaches like Object Modeling Technique (OMT), Classes, Responsibilities, Collaborators (CRC), etc., are used. Objector used the term “agents” to represent some of the hardware and software systems. In Fusion method, there is no requirement phase, where a user will supply the initial requirement document. Any software project is worked out by both the analyst and the designer. The analyst creates the use case diagram. The designer creates the class diagram. But the designer can do this only after the analyst creates the use case diagram. Once the design is over, it is essential to decide which software is suitable for the application.

4.2 INPUT DESIGN

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

1. The dialog to guide the operating personnel in providing input.
2. Methods for preparing input validations and steps to follow when error occur.

OBJECTIVES

1. Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.
2. It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.
3. When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus, the objective of input design is to create an input layer that is easy to follow

4.3 OUTPUT DESIGN

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.

Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively.

Create document, report, or other formats that contain information produced by the system. The output form of an information system should accomplish one or more of the following objectives.

1. Convey information about past activities, current status or projections of the
2. Future.
3. Signal important events, opportunities, problems, or warnings.
4. Trigger an action.
5. Confirm an action.

4.4 UNIFIED MODELING LANGUAGE (UML) DIAGRAMS

4.4.1 UML CONCEPTS

The Unified Modeling Language (UML) is a standard language for writing software blue prints. The UML is a language for

1. Visualizing
2. Specifying
3. Constructing
4. Documenting

4.4.2 BUILDING BLOCKS OF UML

The Unified Modeling Language (UML) is a standard language for writing software blueprints. The vocabulary of the UML encompasses three kinds of building blocks

1. Things
2. Relationships
3. Diagrams

Things are the abstractions that are first-class citizens in a model, Relationships tie these things together, and diagrams group interesting collections of things.

4.4.3 THINGS IN UML

There are four kinds of things in UML:

1. Structural things
2. Behavioral things
3. Grouping things
4. Annotational thigs

4.4.4 USECASE

A use case diagram shows a set of use cases and actors and their relationships.Use case diagrams commonly contain

1. Use cases
2. Actors
3. Dependency, generalization and association relationships

Usecase

A use case describes a set of sequences, in which each sequence represents the interaction of the things outside the system (its actors) with the system itself

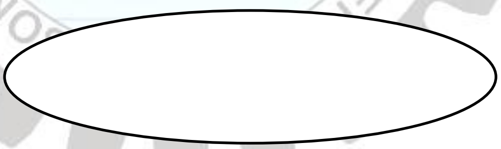


Fig 4.1: Usecase

Actor

An actor represents a coherent set of roles that users of use cases play when interacting with these use cases. Actors can be human or they can be automated systems.

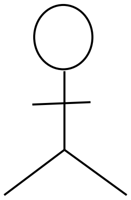


Fig 4.2: Actor

Dependency

A dependency is a semantic relationship between two things in which a change to one thing (the independent thing) may affect the semantics of the other thing (the dependent thing).

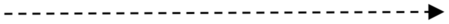


Fig 4.3: Dependency

Generalization

A generalization is a specialization/generalization relationship in which objects of the specialized element (the child) are substitutable for objects of the generalized element (the parent). With this the child shares the structure and the behavior of the parent.

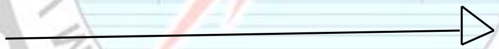


Fig 4.4: Generalization

4.4.5 USECASE DIAGRAM

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. A use case diagram can identify the different types of users of a system and the different use cases and will often be accompanied by other types of diagrams as well. The use cases are represented by either circles or ellipses.

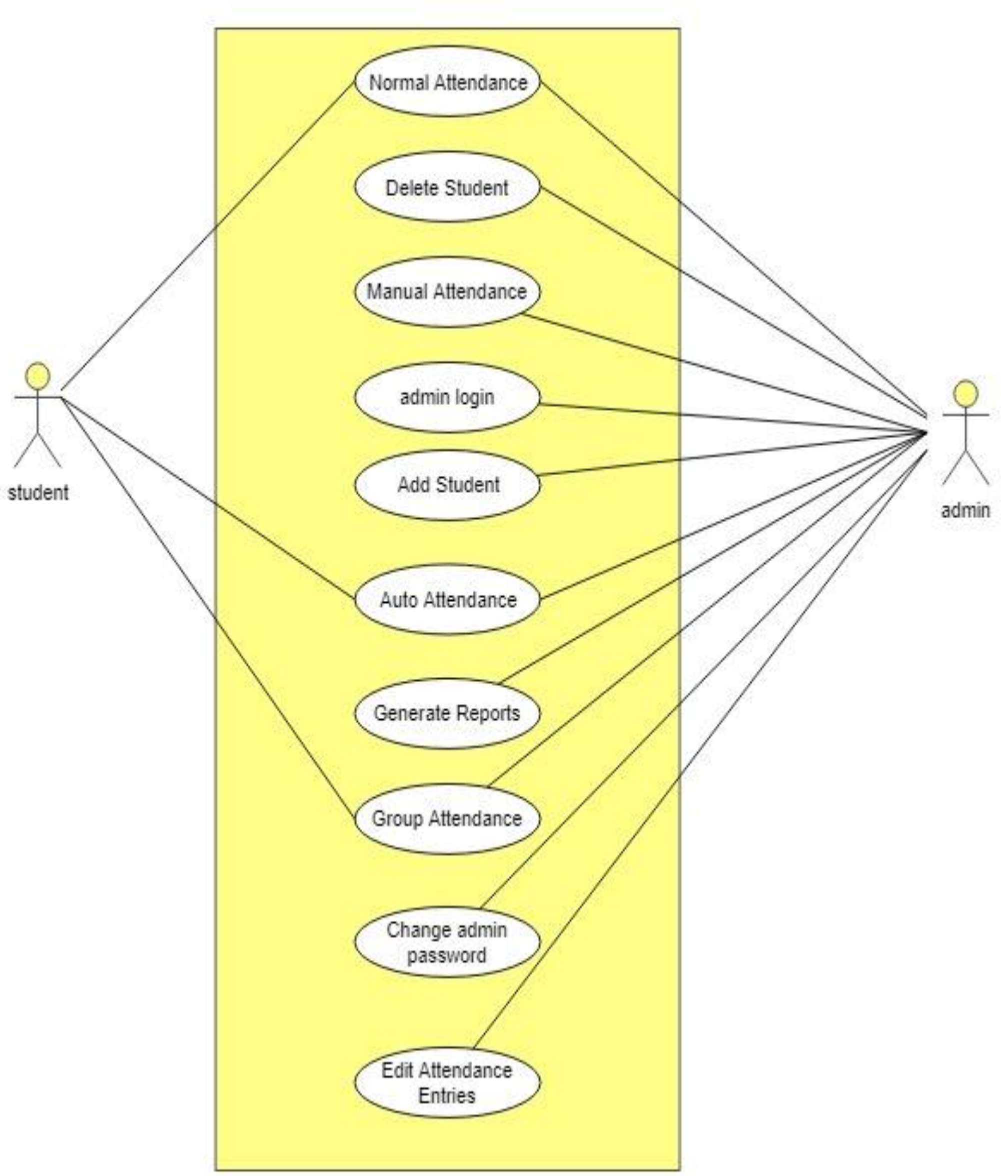


Fig 4.5: Usecase Diagram of Face Recognition Based Attendance System

The above use case diagram shows that, the student will get attendance and admin will perform activities like adding, updating, and deleting student, generating and plotting reports for analysis.

4.4.6 CLASS DIAGRAM

A class diagram shows a set of classes, interfaces, and collaborations and their relationships.

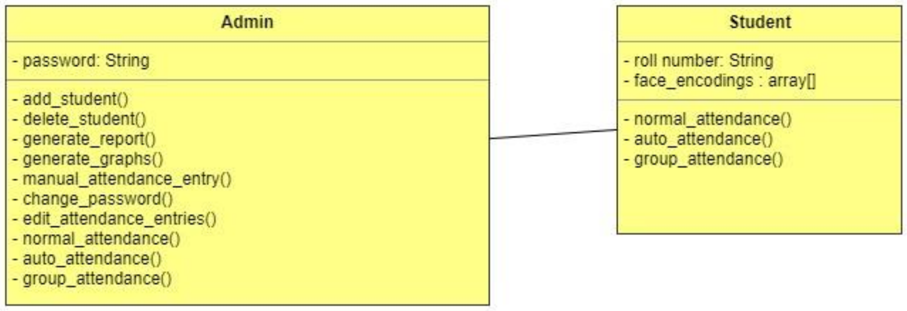


Fig 4.6: Class Diagram of the Face Recognition Based Attendance System

In the above class diagram, the relationship that is association between each one of the classes is sketched out. Additionally, even the operations performed in each and every class is similarly appeared.

4.4.7 SEQUENCE DIAGRAM

UML Sequence diagrams are interaction diagrams that detail how operations are carried out. As sequence diagrams can be used to capture the interaction between objects in the context of a collaboration, one of the primaries uses of sequence diagrams is in the transition from requirements expressed as use cases to the next and more formal level of refinement. Use cases are often refined into one or more sequence diagrams.

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.

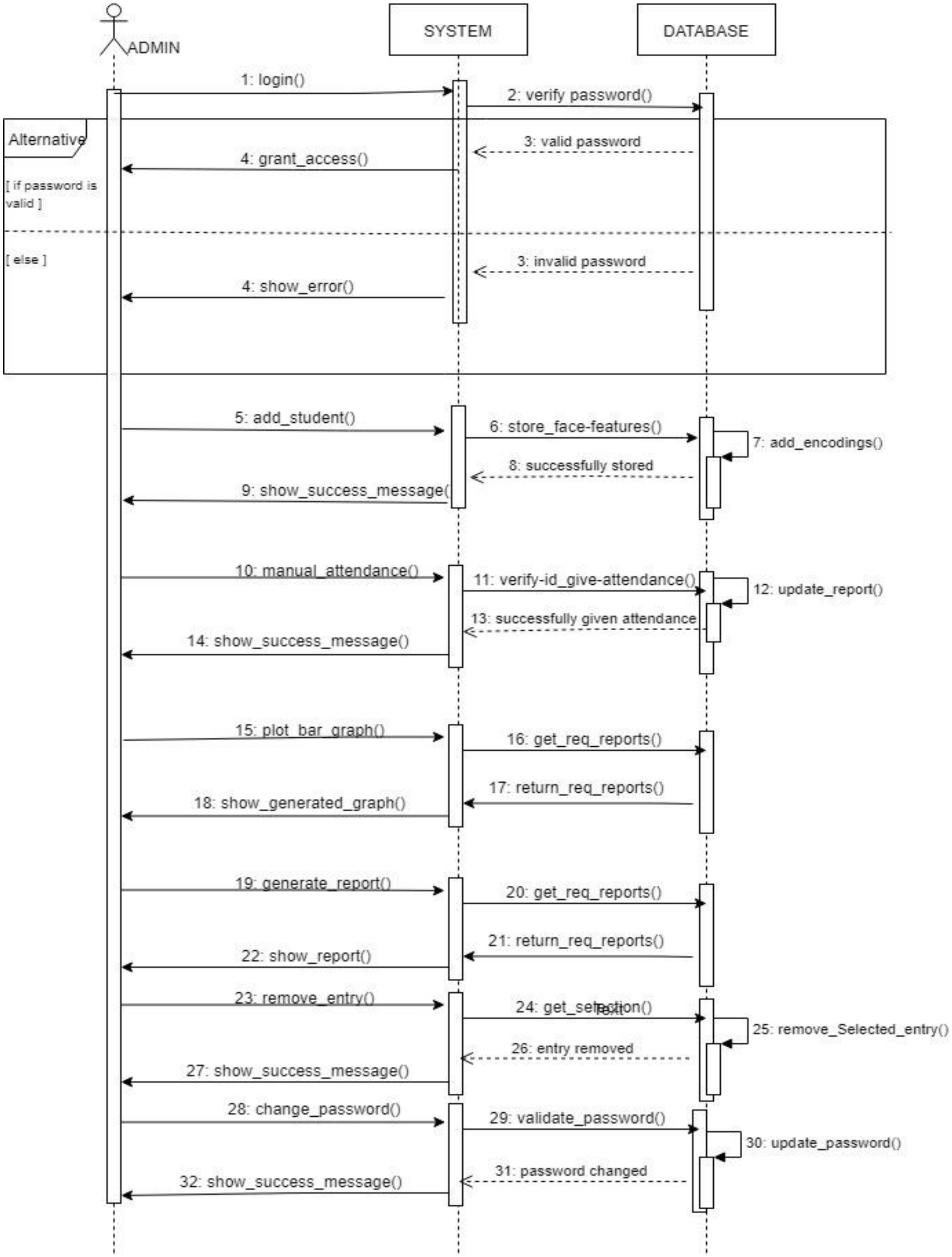


Fig 4.7: sequence diagram of admin

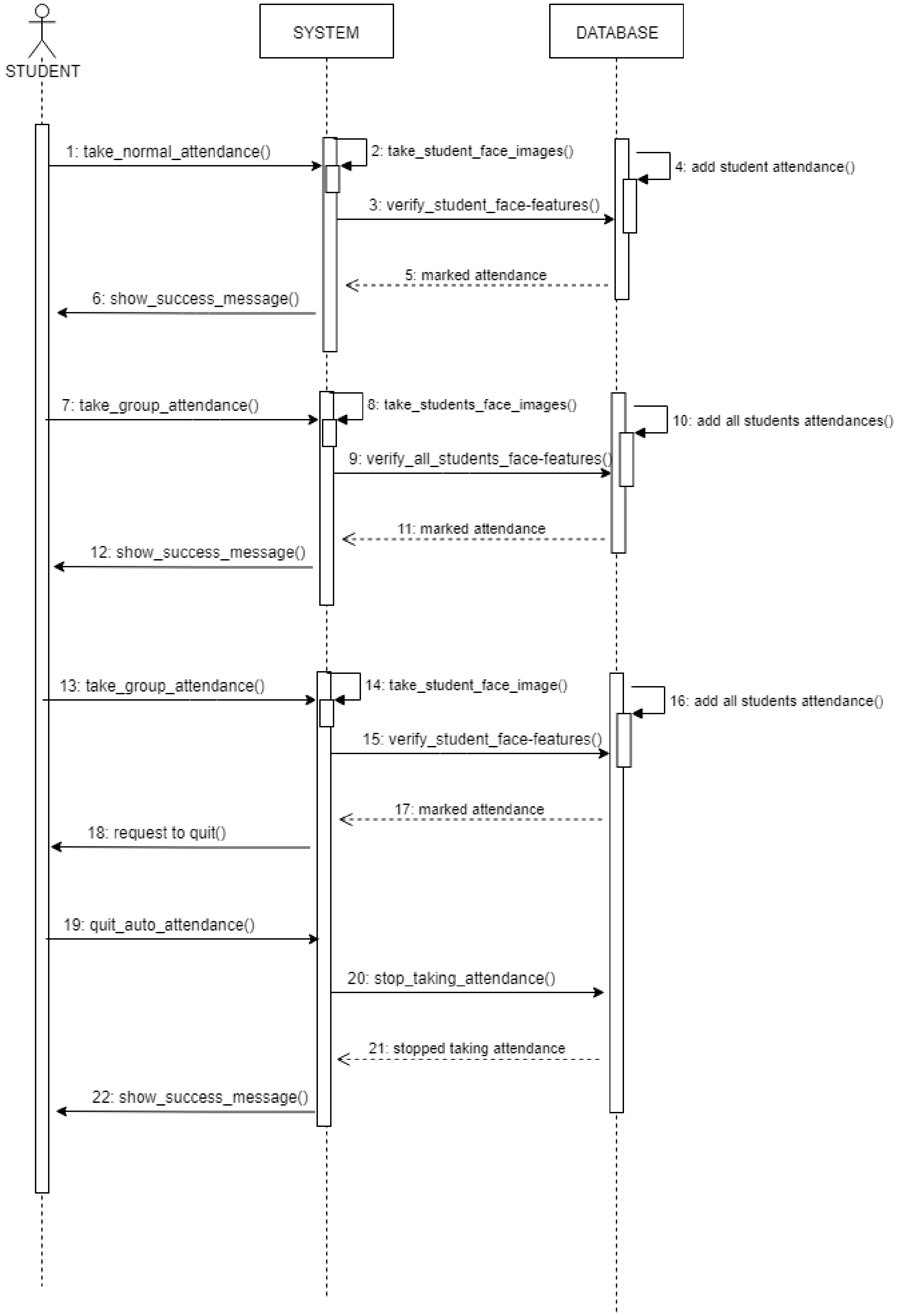

STUDENT
SYSTEM
DATABASE
1: take_normal_attendance()
2: take_student_face_images()
3: verify_student_face-features()
4: add student attendance()
5: marked attendance
6: show_success_message()
7: take_group_attendance()
8: take_students_face_images()
9: verify_all_students_face-features()
10: add all students attendances()
11: marked attendance
12: show_success_message()
13: take_group_attendance()
14: take_student_face_image()
15: verify_student_face-features()
16: add all students attendance()
17: marked attendance
18: request to quit()
19: quit_auto_attendance()
20: stop_taking_attendance()
21: stopped taking attendance
22: show_success_message()

Fig 4.8: sequence diagram of student

The figure 4.5 shows that, the admin will be login with a valid admin password then admin will add student into to database by taking his face image and then he plot some graphs and generate reports.

The fig 4.6 shows that, the student will get attendance by three ways provided by system they are Normal Attendance, Group Attendance and Auto Attendance

4.4.8 E-R DIAGRAM

Entity Relationship Diagram, also known as ERD,ER Diagram or ER model, is a type of structuraldiagram for use in database design. An ERDcontains different symbols and connectors that visualize two important information, The major entities within the system scope, and the inter-relationships among these entities.

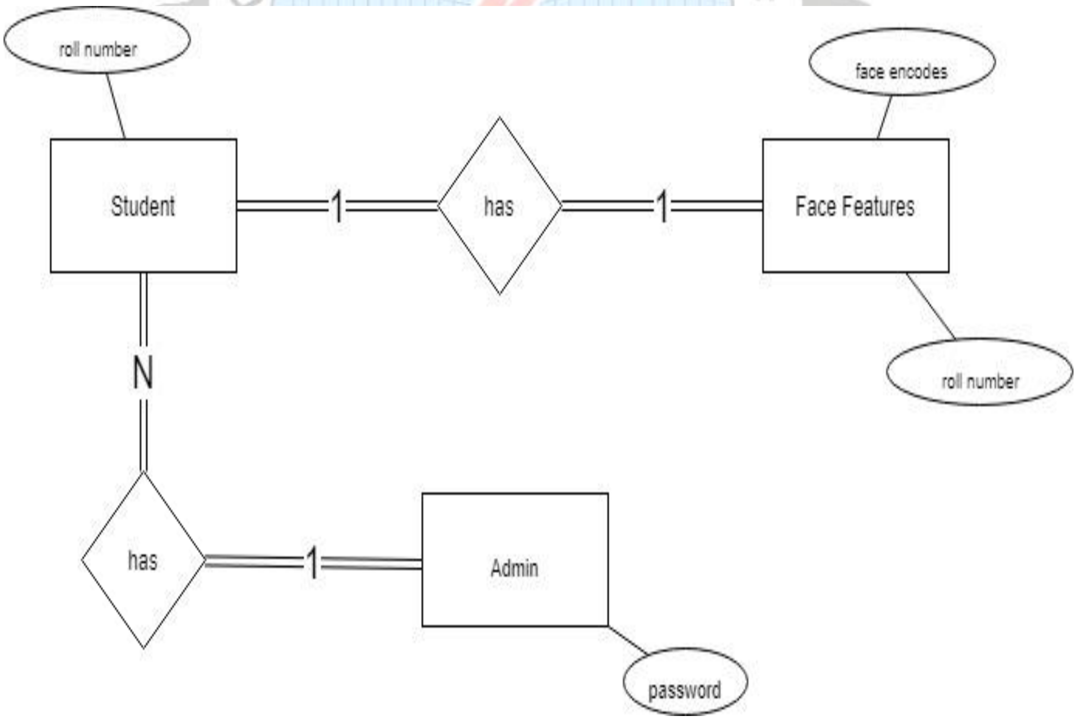


Fig 4.9: ER-Diagram of Face Recognition Based Attendance System

The above ER-Diagram shows that, the entity relationship between student, admin and reports.

5. IMPLEMENTATION AND CODING

5.1 MODULES

The entire project is divided into two modules:

1. Admin
2. Take Attendance

5.2 MODULE DESCRIPTION

5.2.1 ADMIN

Admin module will contain the following functionalities

Normal Attendance

When the admin clicks on take attendance button, then camera will capture images and it automatically detect face, recognize the roll number and show the face and corresponding roll number in a new window continuously. Then, admin press key ‘t’ it will take attendance else if admin press ‘q’ it will stop.

Group Attendance

When the admin clicks on take group attendance button, then camera will capture images and it automatically detect face, recognize the roll number and show the face and corresponding roll number in a new window continuously. If every person is recognized then, if admin press key ‘t’ it will take attendance for every recognized person else if admin press ‘q’ it will stop.

Auto Attendance

When the admin clicks on take auto attendance button, then camera will capture images and it automatically detect face, recognize the roll number and show the face and corresponding roll number in a new window continuously. If a person is recognized then it will respond with a voice whether it has given attendance or not with the roll number.

Add New Student

The admin can add students to the system by entering his roll number and after admin clicks on add student button, a new windows will pop up and shows the images that are captured by the camera, when the system finds the face it will shows a message to admin after seeing that message if admin press key ‘t’ the will capture face and store the 128 face features and the roll number into system else if admin press ‘q’ it will stop.

Delete Student

The admin can delete students from the system by entering his roll number and after admin clicks on remove student button, if student exist in the system with that roll number then it will remove his roll number and corresponding face features from the system.

Generate Reports

The admin can generate reports from past records of attendance. The admin can get the following reports.

1. Get present day attendance percentage.
2. Get specific day attendance percentage.
3. Get average attendance all time.
4. Get all reports save on desktop
5. Get specific day attendance report

Plot Graphs

The admin can plot graphs for statistical analysis. the admin can plot following graphs.

- Plot bar graph for attendances and dates for all reports.
- Plot bar graph between the selected dates range

Manual Attendance

The admin can give attendance for specific roll number manually if that roll number exist

Remove Attendance Entries

In case, if the system has taken wrong attendance, the admin can remove the attendance entries.

Change password

The admin change login password, by entering old password and new password in the given entries. If the old password is correct then, the password will be updated to entered new password.

5.2.2 Take Attendance

Take Attendance module will contain the following functionalities.

Normal Attendance

When the admin clicks on take attendance button, then camera will capture images and it automatically detect face, recognize the roll number and show the face and corresponding roll number in a new window continuously. Then, admin press key ‘t’ it will take attendance else

Group Attendance

When the admin clicks on take group attendance button, then camera will capture images and it automatically detect face, recognize the roll number and show the face and corresponding roll number in a new window continuously. If every person is recognized then, if admin press key ‘t’ it will take attendance for every recognized person else if admin press ‘q’ it will stop.

Auto Attendance

When the admin clicks on take auto attendance button, then camera will capture images and it automatically detect face, recognize the roll number and show the face and corresponding roll number in a new window continuously. If a person is recognized then it will respond with a voice whether it has given attendance or not with the roll number.

5.3 SAMPLE CODING:

The below program is the main page of the Face Recognition Based Attendance system. From this page admin or student will be redirected to his page.

fras.py

```
from tkinter import *
from tkinter import ttk
from os import system
from tkinter import Button
from tkinter import Label
from tkinter import Entry
from tkinter import Tk
from tkinter import messagebox
import pickle
window = Tk()
window.title("FRAS")
window.attributes("-fullscreen",True)
label_title = Label(window, text="FACE RECOGNITION BASED ATTENDANCE SYSTEM",width=60  ,height=2 , bg='green', fg='white' ,font=('times', 30, ' bold ') )
label_title.place(x=0,y=0 )
label_pass = Label(window, text="Enter password   : ",width=13  ,height=2 ,font=('times', 15, ' bold ') )
label_pass.place(relx=0.6,rely=0.3,x=-20 )
entry_pass = Entry(window,width=7  ,bg="white" ,fg="red",font=('times', 13, ' bold '), )
entry_pass.place(relx=0.7,rely=0.3,y=17 )
def clear():
    entry_pass.delete(0,'end')
def get_old_pass():
    with open("./Data/pa.pkl", "rb") as f:
      passwords = pickle.load(f)
      password = passwords[0]
      print("old password",password)
      return password
def admin_area():
    pwd = get_old_pass()
    if pwd == entry_pass.get():
        cmd = "python admin_area.py"
        system(cmd)
        clear()
    else:
        messagebox.showerror("Authentication error","Wrong password !   ")
def student_area():
    cmd = "python student_area.py"
    system(cmd)
take_group_att_btn = Button(window, text="TAKE ATTENDANCE", bg="blue", fg="white", command=student_area, font=('times', 20, ' bold '))
take_group_att_btn.place(relx=0.2, rely=0.4,  )
take_group_att_btn = Button(window, text="ADMIN AREA", bg="red", fg="white", command=admin_area, font=('times', 20, ' bold '))
take_group_att_btn.place(relx=0.6, rely=0.4,  )
take_group_att_btn = Button(window, text="Quit", bg="black", fg="white", command=quit, font=('times', 18, ' bold '))
```

```
take_group_att_btn.place(relx=0.8, rely=0.8,  )
window.mainloop()
```

The below program contain the attendance taking system. The student will be given attendance here.

take_attendance_page.py

```
import get_reports
import facerec
from tkinter import ttk
from tkinter import messagebox
import plot_graphs
import group_attendance
from tkinter import Tk
from tkinter import Button
from tkinter import Label
window = Tk()
window.title("FRAS")
window.attributes("-fullscreen",True)
tab_control = ttk.Notebook(window)
take_attendance_tab = ttk.Frame(tab_control)
group_attendance_tab = ttk.Frame(tab_control)
auto_attendance_tab = ttk.Frame(tab_control)
tab_control.add(take_attendance_tab, text='  Normal Attendance  ')
tab_control.add(group_attendance_tab, text='  Group Attendance  ')
tab_control.add(auto_attendance_tab, text=' Auto Attendance  ')
def quit():
    global window
    window.quit()
take_group_att_btn = Button(window, text="Quit", bg="black", fg="white", command=quit, font=('times', 20, ' bold '))
take_group_att_btn.place(relx=0.8, rely=0.8, x=0,y=30 )
instructions_tab1 = """
Instructions : After clicking on "Take Attendance" button
    1. Place your face infront of camera it will show your face
    2. Adjust your face to get your rollnumber, If you can see your rollnumber
    3. Then press  'T'  to take attencance
    4. If you want quit from taking press 'Q'
"""
tab1_ins_label = Label(take_attendance_tab, text=instructions_tab1, width=60, height=15, font=('times', 15, ' bold ') )
tab1_ins_label.place(x=0,y=0 )
def take_student_attendance():
    Id,status = facerec.take_student_attendance()
    if status == -1:
        messagebox.showerror('No Attendance Given', 'Terminated by user')
    elif status == -2:
        messagebox.showwarning('Already taken', str(Id)+': Student has already given attendance')
    else:
        msg = str(Id)+' : Student has given attendance'
        messagebox.showinfo('sucess ',msg)
take_att_btn = Button(take_attendance_tab, text="Take Attendance", bg="red", fg="white", command=take_student_attendance, font=('times', 15, ' bold '))
```

```
take_att_btn.place(relx=0.2,x=0,y=30 )
instructions_tab2 = """
Instructions : After clicking on "Take Attendance" button
    1. Place your face infront of camera it will show your face
    2. Adjust your face to get your rollnumber
    3. Do 1,2 steps for each person
    4. Then press  'T'  to take attencance for all people
    5. Press 'Q' if you want quit from taking attendance
"""
tab1_ins_label = Label(group_attendance_tab, text=instructions_tab2, width=60, height=15, font=('times', 15, ' bold ') )
tab1_ins_label.place(x=0,y=0 )
def take_group_attendance():
    status = facerec.take_group_attendance()
    if status == -1:
        messagebox.showerror('No Attendance Given', 'Terminated by user !')
    else:
        messagebox.showinfo('sucess ',status)
take_grp_att_btn = Button(group_attendance_tab, text="Take Group Attendance", bg="red", fg="white", command=take_group_attendance, font=('times', 15, ' bold '))
take_grp_att_btn.place(relx=0.2,x=0,y=30 )
instructions_tab3 = """
Instructions : After clicking on "Take Attendance" button
    1. Place your face infront of camera it will show your face
    2. Adjust your face to get your rollnumber and attendance will be automatically given
    3. Do 1,2 steps for each person
    5. Press 'Q' if you want quit from taking attendance
"""
tab3_ins_label = Label(auto_attendance_tab, text=instructions_tab3, width=60, height=15, font=('times', 15, ' bold ') )
tab3_ins_label.place(x=0,y=0 )
def take_auto_attendance_fun():
    messagebox.showinfo("Auto Attendance", "Press 'Q' if you want to  stop taking attnedance")
    facerec.take_auto_attendance()
take_auto_att_btn = Button(auto_attendance_tab, text="Take Auto Attendance", bg="red", fg="white", command=take_auto_attendance_fun, font=('times', 15, ' bold '))
take_auto_att_btn.place(relx=0.2,x=0,y=30 )
tab_control.pack(expand=1, fill='both')
window.mainloop()
```

6 TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring.

Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

6.1 TYPES OF TESTS

6.1.1 UNIT TESTING

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

6.1.2 INTEGRATION TETSING

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

6.1.3 FUNCTIONAL TEST

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

6.2 SYSTEM TEST

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration-oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

6.2.1 WHITE BOX TESTING

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

6.2.2 BLACK BOX TESTING

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box. we cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

6.3 TEST CASES

6.3.1 TAKE ATTENDANCE MODULE TEST CASES

Test Case ID	Test Scenario	Test Data	Expected Results	Actual Results	Pass/ Fail
T1	Take Normal Attendance with valid student	Student face	Student Successfully given attendance	Student Successfully given attendance	pass
T2	Take Normal Attendance with invalid student	Student face	Student Successfully given attendance	Student not recognized	fail
T3	Take Group Attendance with valid students	Student faces	Students Successfully given attendance	Students Successfully given attendance	pass
T4	Take Group Attendance with invalid students	Student faces	Students faces not recognized	Students faces not recognized	pass
T5	Take Auto Attendance with valid students	Student faces	Student Successfully given attendance	Student Successfully given attendance	pass

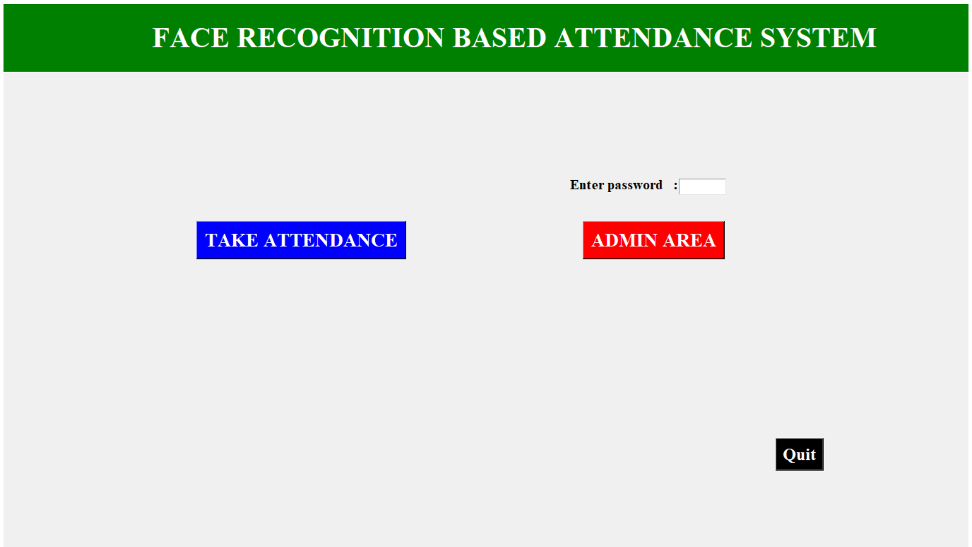
6.3.2 ADMIN MODULE TEST CASES

Test Case ID	Test Scenario	Test Data	Expected Results	Actual Results	Pass/ Fail
1	Check admin login with valid password	Password: ‘fras’	Admin successfully logged in	Admin successfully logged in	pass
2	Check admin login with invalid password	Password: ‘abcd’	Admin login failed	Admin login failed	pass
3	Check Student addition with valid data	Image: Student Face Image, id: ‘529 ‘	Student Added Successfully	Student Added Successfully	pass
4	Check Student addition with invalid data	Image: Student Face Image, id: ‘##@#$’	Student Added Successfully	Invalid Roll Number	fail
5	Change password with valid old password	Old password: ‘fras’ new password: ‘abcd’	Password changed successfully	Password changed successfully	pass
6	Change password with invalid old password	Old password: ‘frassss’ new password: ‘abcd’	Password changed successfully	Wrong old Password	fail

7. OUTPUT SCREENS

7.1 STARTING PAGE

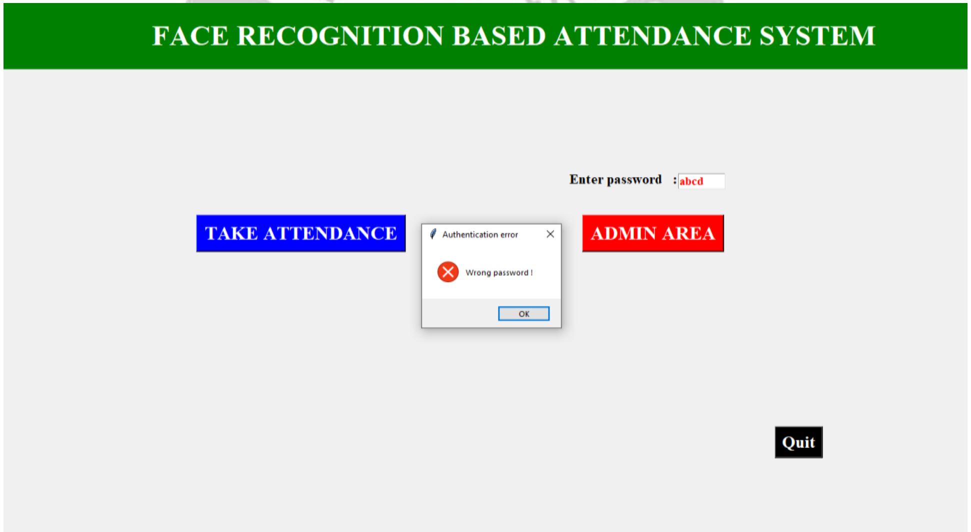
After executing of project, the first page appears like this



FACE RECOGNITION BASED ATTENDANCE SYSTEM

Enter password :

Fig 7.1: home page



FACE RECOGNITION BASED ATTENDANCE SYSTEM

Enter password : abcd

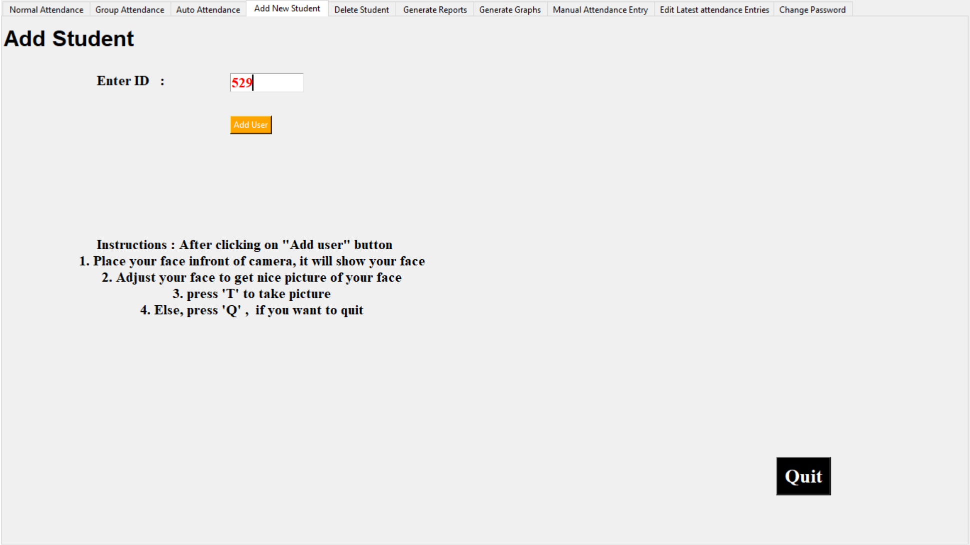
Authentication error

Wrong password !

Fig 7.2: wrong password login

7.2 ADD NEW STUDENT

After admin login with correct password can add new students into system.



Add Student

Enter ID : 529

Instructions : After clicking on "Add user" button
1. Place your face infront of camera, it will show your face
2. Adjust your face to get nice picture of your face
3. press 'T' to take picture
4. Else, press 'Q' , if you want to quit

Fig 7.3: Add new student into system by giving roll number

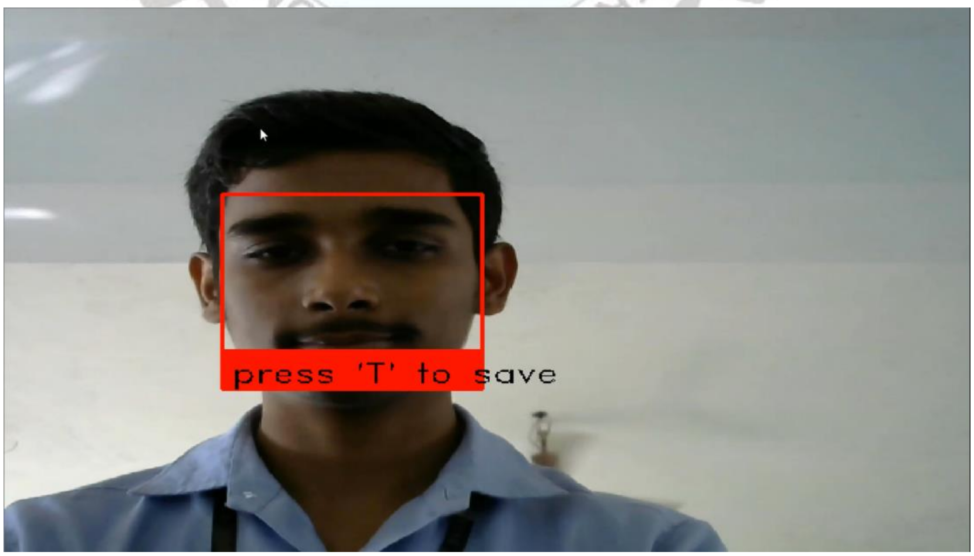
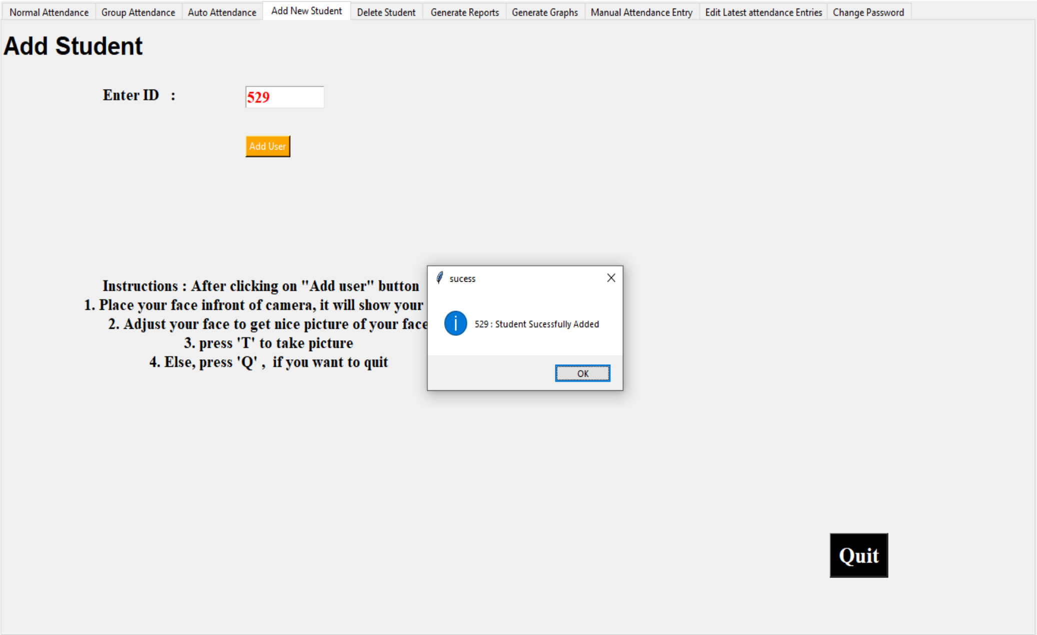


Fig 7.4: Taking picture of student’s face



Add Student

Enter ID : 529

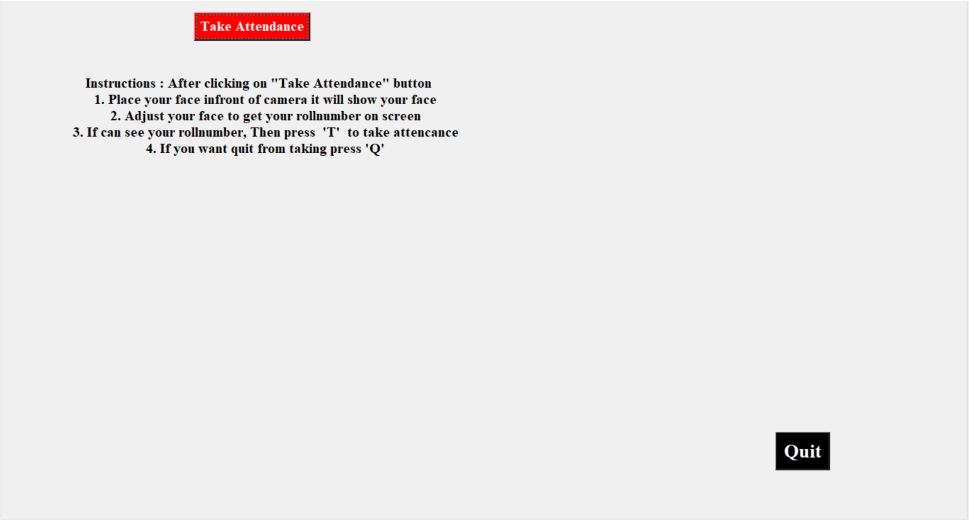
Instructions : After clicking on "Add user" button
1. Place your face infront of camera, it will show your
2. Adjust your face to get nice picture of your face
3. press 'T' to take picture
4. Else, press 'Q' , if you want to quit

529 : Student Sucessfully Added

Fig 7.5: Student added Successfully

7.3 TAKE NORMAL ATTENDANCE

Students can get attendance individually by clicking on ‘Take attendance’ button in ‘Normal Attendance’ tab



Instructions : After clicking on "Take Attendance" button
1. Place your face infront of camera it will show your face
2. Adjust your face to get your rollnumber on screen
3. If can see your rollnumber, Then press 'T' to take attencance
4. If you want quit from taking press 'Q'

Fig 7.6: Take Normal Attendance

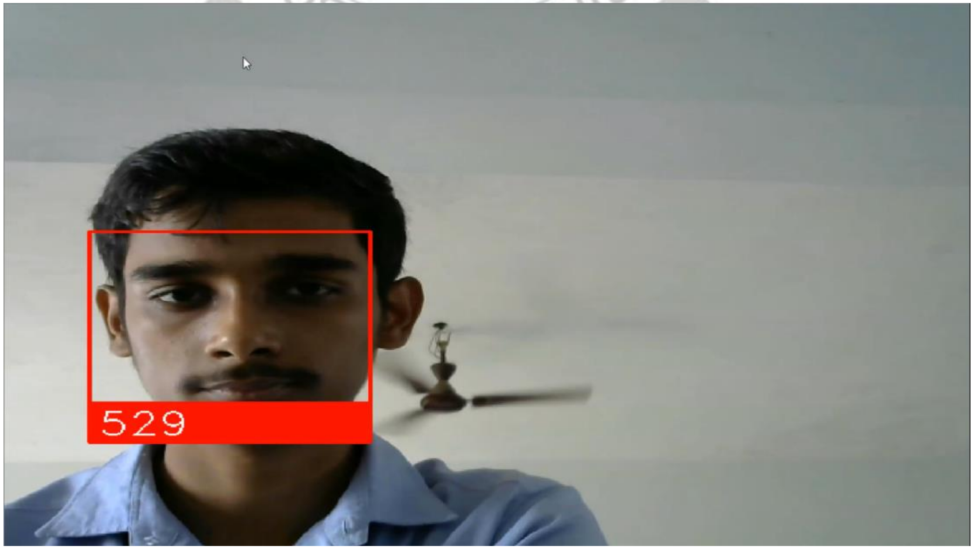
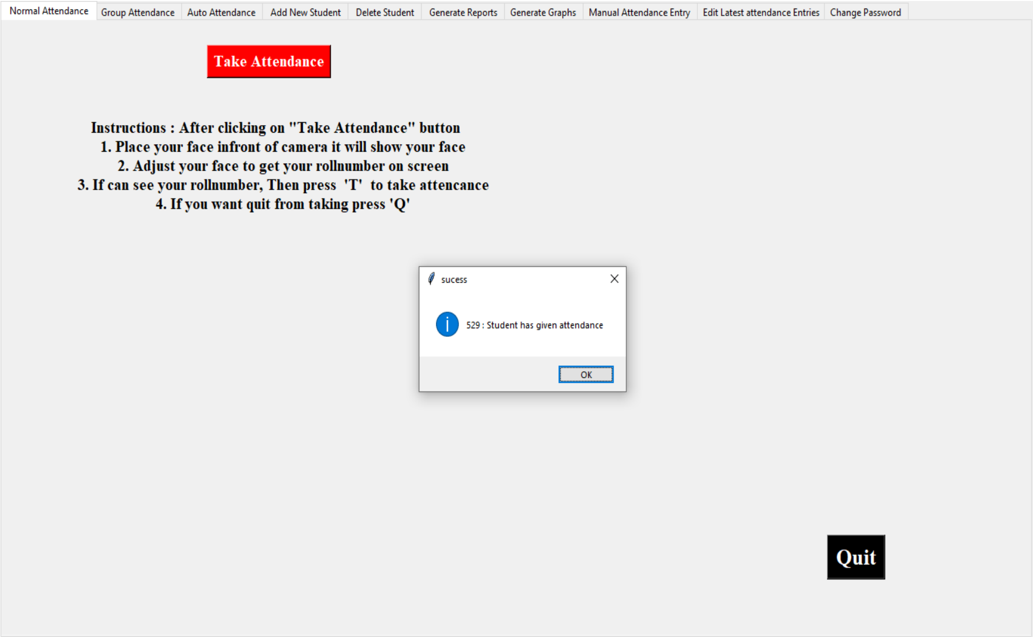


Fig 7.7: Recognizing face of student



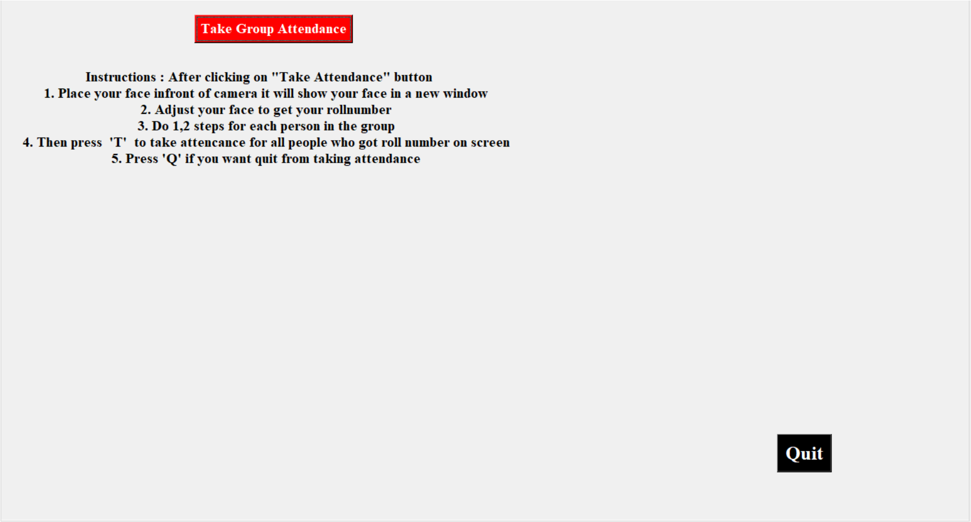
Instructions : After clicking on "Take Attendance" button
1. Place your face infront of camera it will show your face
2. Adjust your face to get your rollnumber on screen
3. If can see your rollnumber, Then press 'T' to take attencance
4. If you want quit from taking press 'Q'

529 : Student has given attendance

Fig 7.8: Attendance given successfully

7.4 TAKING GROUP ATTENDANCE

A Group of students can get attendance individually by clicking on ‘Take Group Attendance’ button in ‘Group Attendance’ tab



Instructions : After clicking on "Take Attendance" button
1. Place your face infront of camera it will show your face in a new window
2. Adjust your face to get your rollnumber
3. Do 1,2 steps for each person in the group
4. Then press 'T' to take attencance for all people who got roll number on screen
5. Press 'Q' if you want quit from taking attendance

Fig 7.9: Taking group attendance

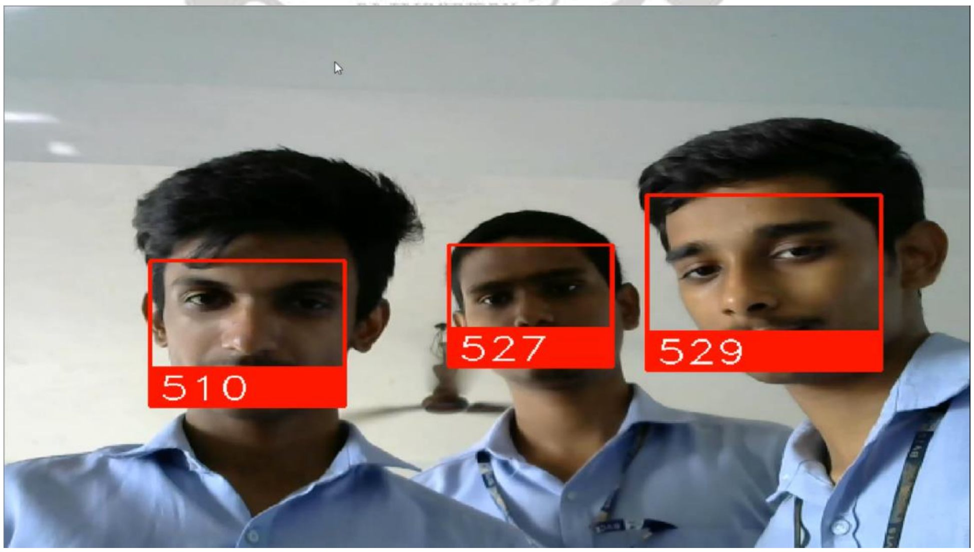
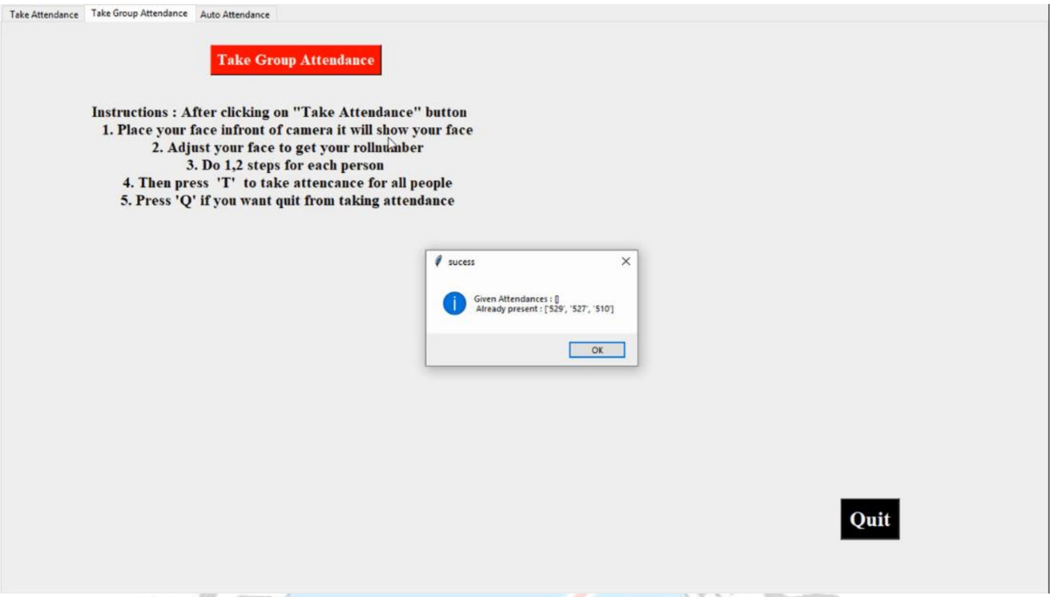


Fig 7.10: Recognizing faces of students



Instructions : After clicking on "Take Attendance" button
1. Place your face infront of camera it will show your face
2. Adjust your face to get your rollnumber
3. Do 1,2 steps for each person
4. Then press 'T' to take attencance for all people
5. Press 'Q' if you want quit from taking attendance

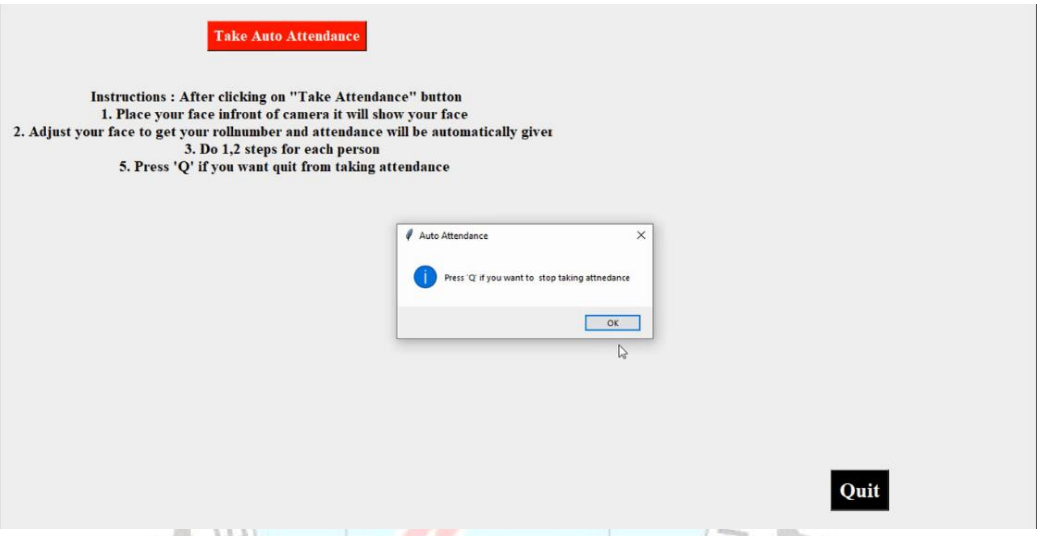
sucess

Given Attendances : []
Already present : ['529', '527', '510']

Fig 7.11: Students given attendance successfully

7.5 TAKING AUTO ATTENDANCE

Student attendance can be taken automatically after recognizing the student’s face by the camera.



Instructions : After clicking on "Take Attendance" button
1. Place your face infront of camera it will show your face
2. Adjust your face to get your rollnumber and attendance will be automatically giver
3. Do 1,2 steps for each person
5. Press 'Q' if you want quit from taking attendance

Press 'Q' if you want to stop taking attnedance

Fig 7.12: Taking auto attendance

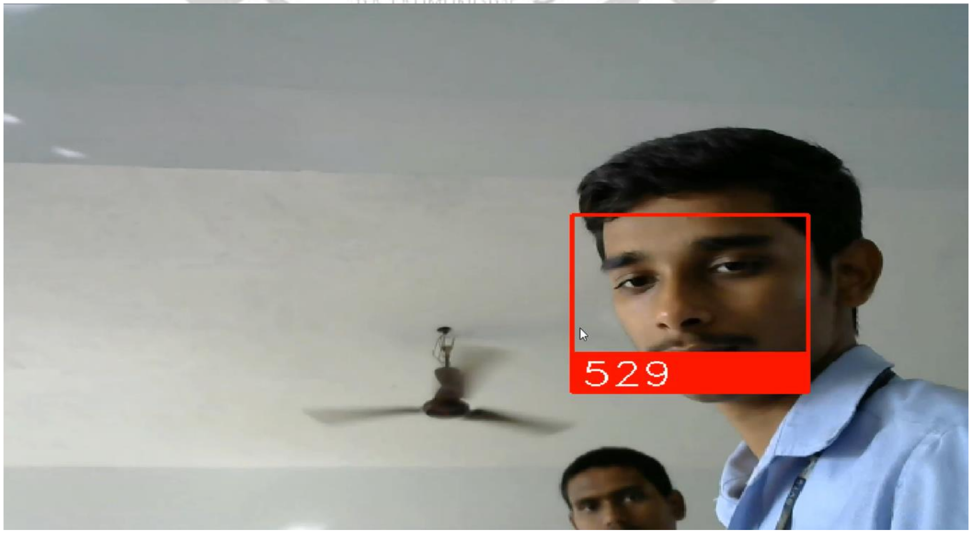
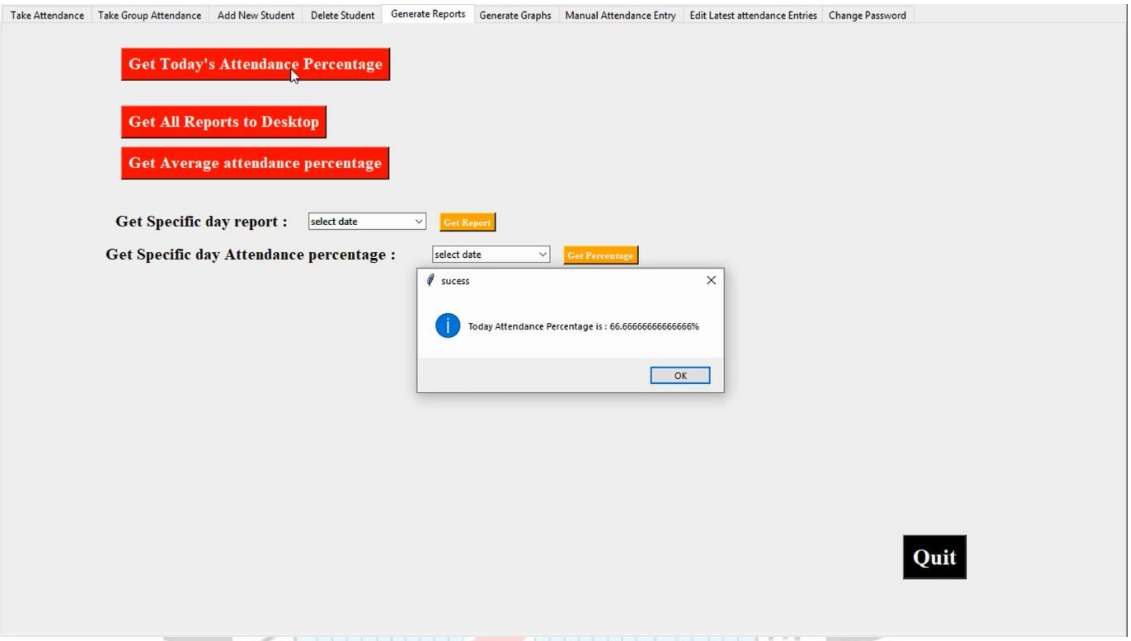


Fig 7.13: Taking attendance automatically

7.6 GENERATING REPORTS



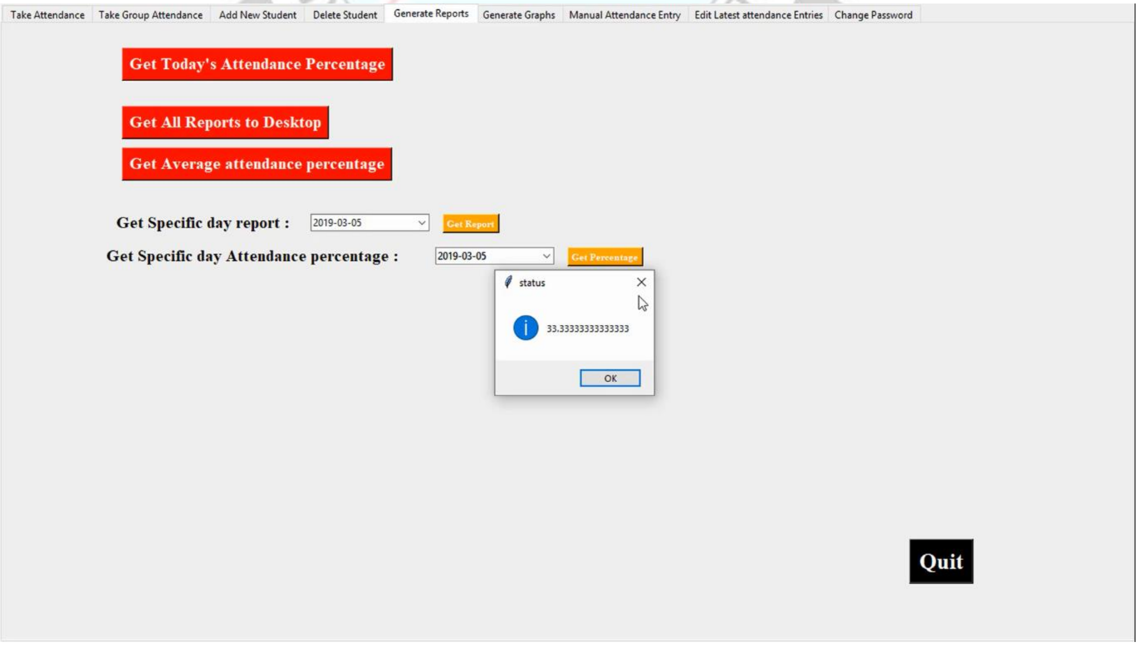
Get Specific day report : select date

Get Specific day Attendance percentage : select date

sucess

Today Attendance Percentage is : 66.66666666666666%

Fig 7.14: Generating present day attendance percentage



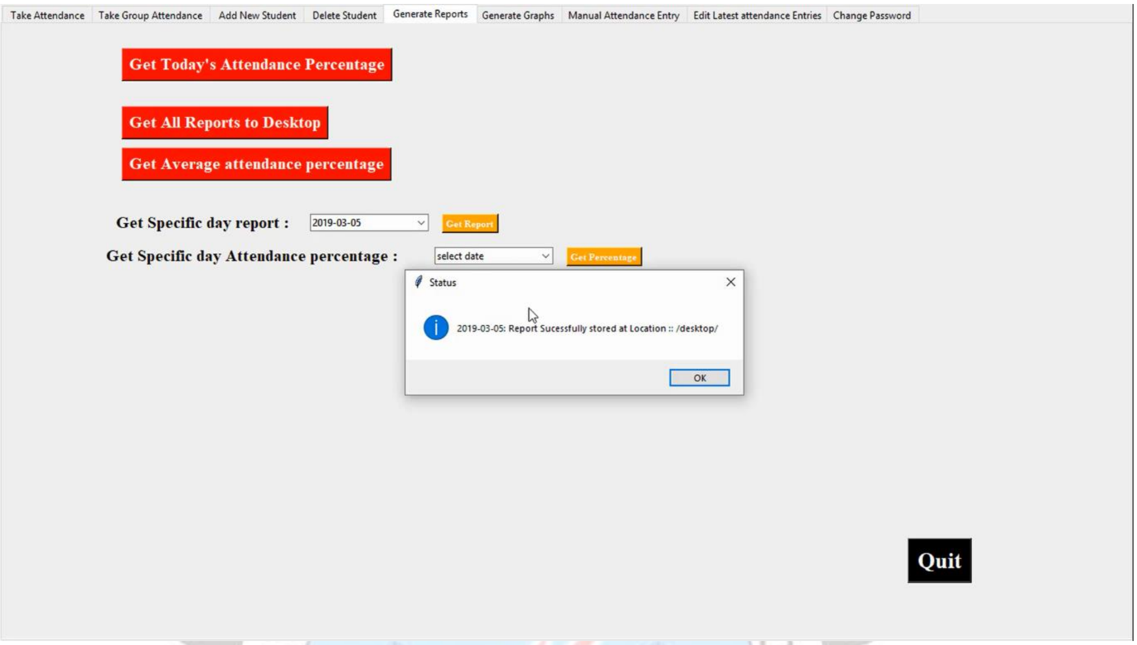
Get Specific day report : 2019-03-05

Get Specific day Attendance percentage : 2019-03-05

status

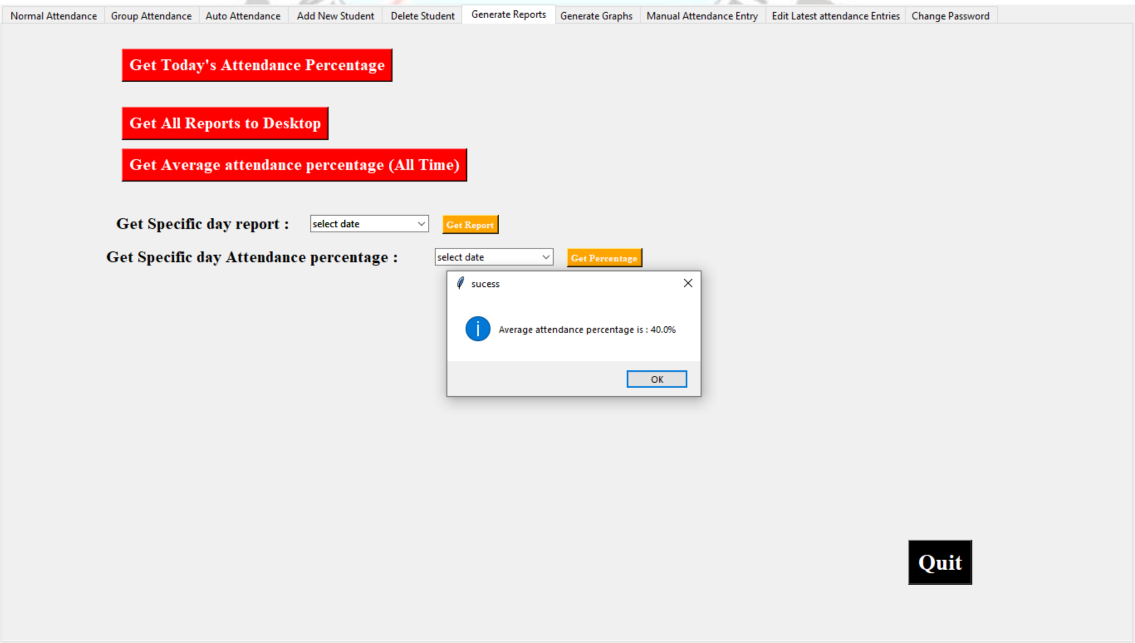
33.33333333333333

Fig 7.15: Get particular day attendance percentage



2019-03-05: Report Sucessfully stored at Location :: /desktop/

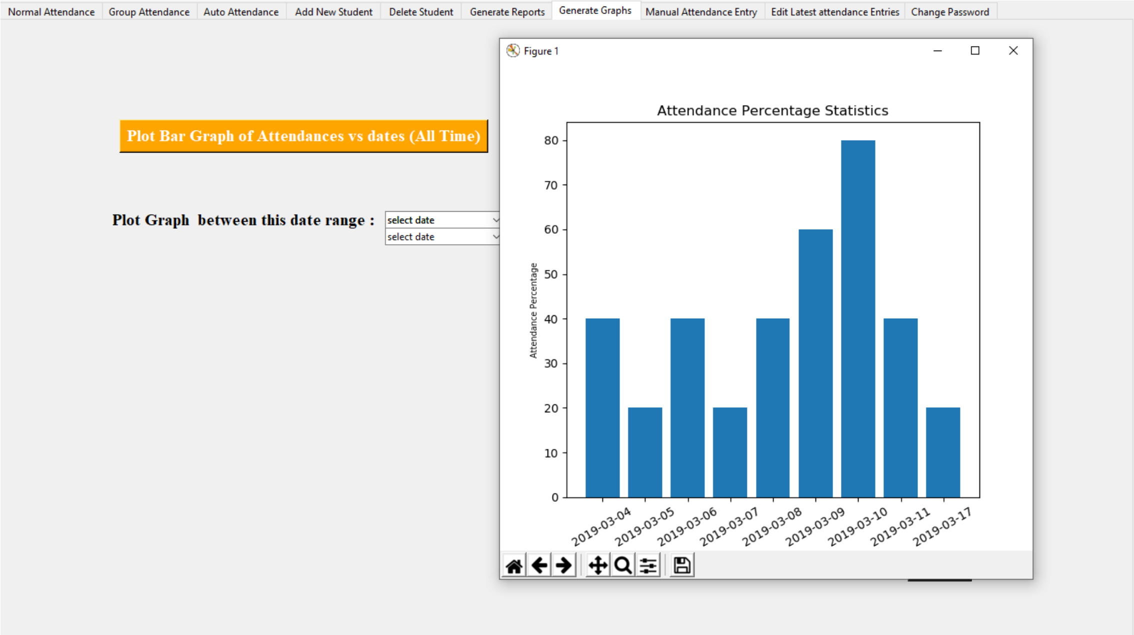
Fig 7.16: Get particular day attendance report store on desktop



Average attendance percentage is : 40.0%

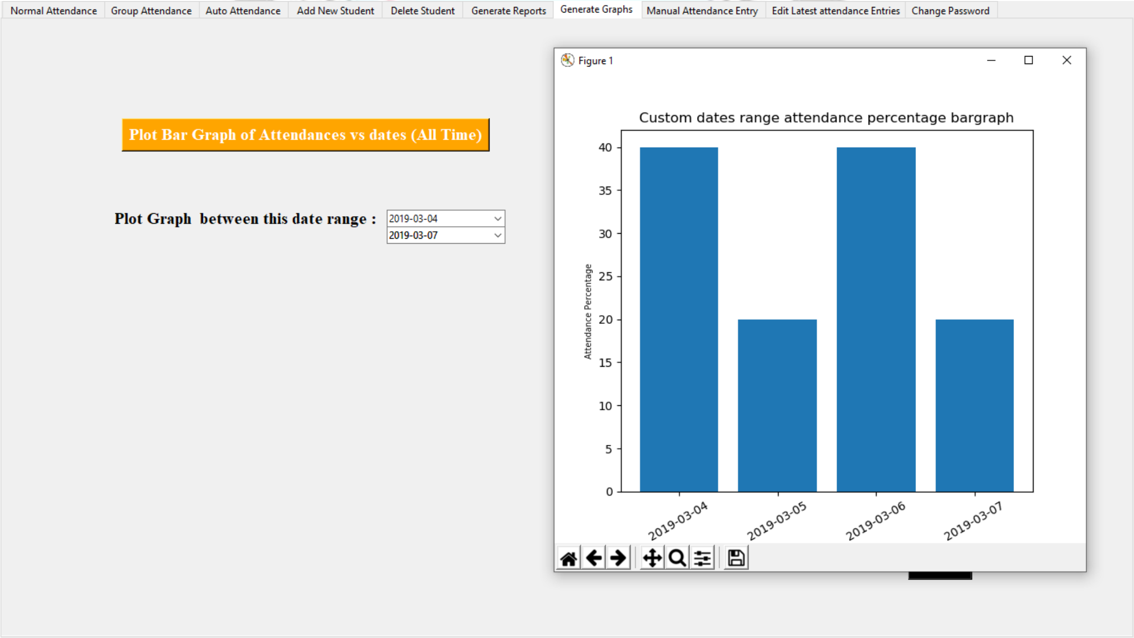
Fig 7.17: Get average attendance percentage (all time)

7.7 PLOTING GRAPHS



Plot Graph between this date range : select date
select date

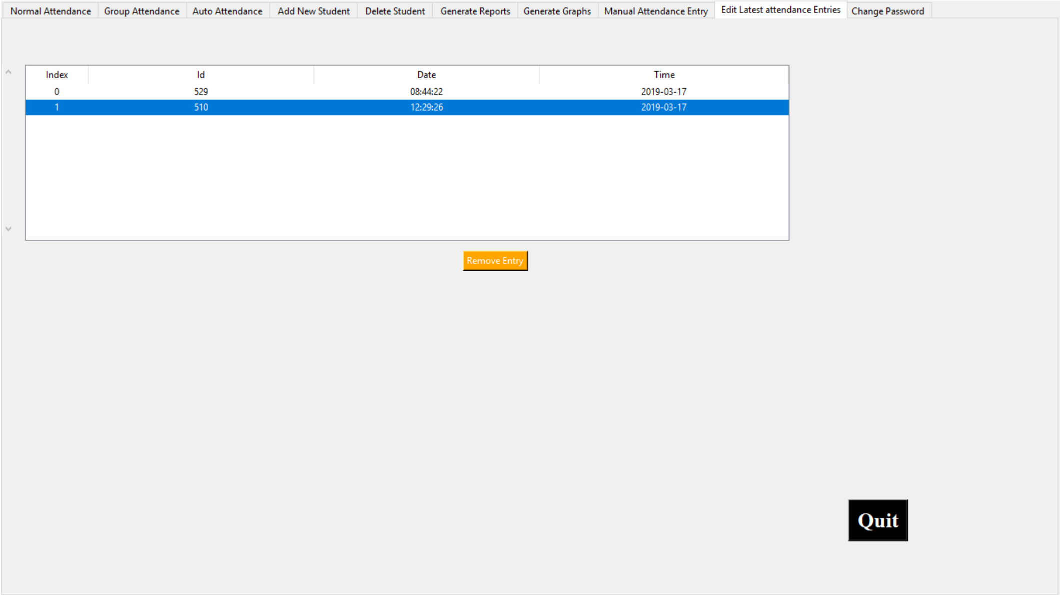
Fig 7.18: Plotting bar graph for all stored reports between attendance percentage and dates



Plot Graph between this date range : 2019-03-04
2019-03-07

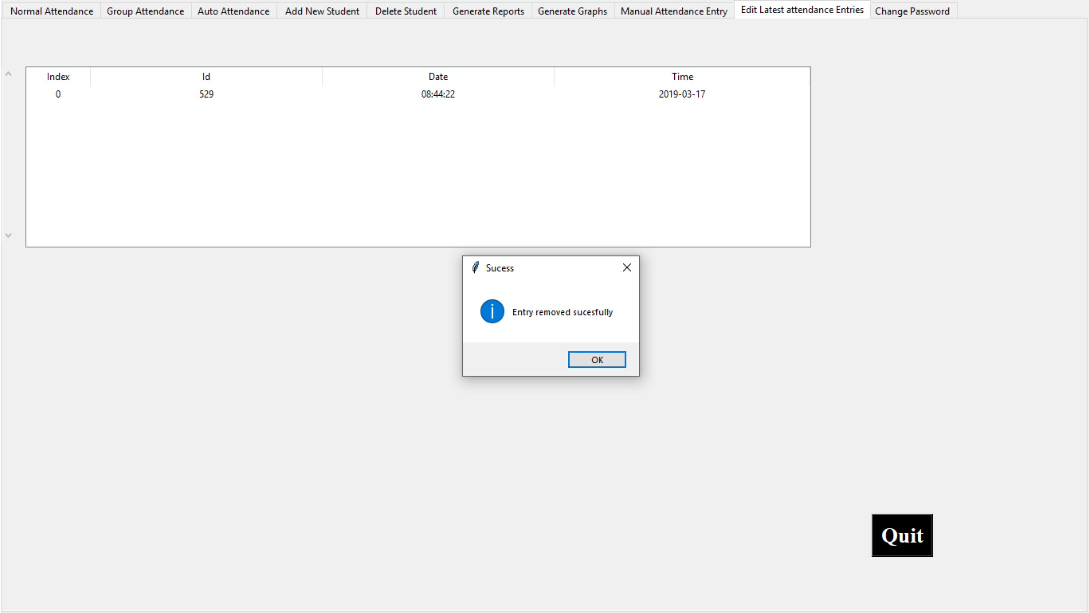
Fig 7.19: Plotting bar graph for a selected dates range and attendance percentage

7.8 REMOVE ATTENDANCE ENTRIES



Index	Id	Date	Time
0	529	08:44:22	2019-03-17
1	510	12:29:26	2019-03-17

Fig 7.21: Selecting attendance entry to remove



Index	Id	Date	Time
0	529	08:44:22	2019-03-17

Entry removed sucesfully

Fig 7.22: Attendance entry removed successfully

8. CONCLUSION AND FUTURE ENHANCEMENT

8.1 CONCLUSION:

In order to take and maintain the attendance this system has been proposed. It replaces the manual system with an automated system which is fast, efficient, cost and time saving as replaces the stationary material and the paper work. Hence this system is expected to give desired results and in future could be implemented for logout. Also, the efficiency could be improved by integrating other techniques with it in near future.

8.2 FUTURE ENHANCEMENT:

The system can be easily implemented at any institute or organization. A method could be proposed to illustrate robustness against the variations that is, in near future we could build a system which would be robust and would work in undesirable conditions too. Here it is proposed for an institute to take the attendance of the students but in future it can be used to do the same work at entry as well as exit points. Authors are working to improve the face recognition effectiveness to build more efficient systems in near future. In further work, authors intend to improve face recognition effectiveness by using the interaction among our system, the users and the administrators. On the other hand, our system can be used in a completely new dimension of face recognition application, mobile based face recognition, which can be an aid for common people to know about any person being photographed by cell phone camera including proper authorization for accessing a centralized database.

9. BIBILOGRAPHY

[1] Rekha A, Dr. Chethan H, Automated Attendance System Using Face Recognition Through Video Surveillance, International Journal For Technological Research In Engineering, Volume 1, Issue 11, and July-2014.

[2] Nirmalya Kar, Mrinal Kanti Debbarma, Ashim Saha, and Dwijen Rudra Pal, Study Of Implementing Automated Attendance System Using Face Recognition Technique, International Journal Of Computer And Communication Engineering, Vol. 1, No. 2, July 2012.

[3] Naveed Khan Balcoh, M. HaroonYousaf, Waqar Ahmad and M. Iram Baig, Algorithm for Efficient Attendance Management: Face Recognition Based Approach, International Journal of Computer Science Issues, Vol. 9, Issue 4, No 1, July 2012.

[4] Kwok-Wai Wong, Kin-Man Lam, Wan-Chi Siu, An efficient algorithm for human face detection and facial feature extraction under different conditions, The Journal Of the Pattern Recognition Society, 25 Aug-2000.

[5] Sarabjit Singh, Amritpal Kaur, Taqdir, A Face Recognition Technique using Local Binary Pattern Method, International Journal of Advanced Research in Computer and Communication Engineering Vol. 4, Issue 3, March 2015.

[6] Learning OpenCV: Computer Vision with the OpenCV Library by Gary Rost Bradski O'REILLY

[7] Priyanka Wagh, Roshani Thakare, Jagruthi Chaudhari, Sweetha Patil, Attendance System Based on Face Recognition Using Eigen Face and PCA Algorithm, IEEE Paper published on Nov 19, 2016.

[8] Kennedy Okokpujie, Etinosa Noma-Osaghae, Samuel John, A Face Recognition Attendance System with GSM Notification, IEEE paper published on Oct 8, 2015.