AUTOMATED ATTENDANCE SYSTEM

Automated using Face Recognition.

Report |Case Study |22 May 2021

AUTOMATED ATTENDANCE SYSTEM

A Report Submitted in Partial Fulfilment for the Course Entitled Case Study in VI Semester
for the Degree in Bachelor of Computer Science and Engineering.

By,

Ms. Komali Beeram, 1601 18 733 067

BE ¾ Computer Science and Engineering

Ms. K Tina, 1601 18 733 086

BE ¾ Computer Science and Engineering

Under the Guidance of,

Dr. Kolla Morarjee

Assistant Professor

Chaitanya Bharathi Institute of Technology

Department of Computer Science and Engineering,
Chaitanya Bharathi Institute of Technology
Gandipet, Hyderabad, Telangana State, India, PIN-500075
Website: www.cbit.ac.in
May 2021

CHAITANYA BHARATHI
INSTITUTE OF TECHNOLOGY (A)
Affliated to Osmania University

CERTIFICATE

This is to certify that the project title "Automated Attendance System", submitted to Department of Computer Science and Engineering, Chaitanya Bharathi Institute of Technology, in partial fulfilment of the requirements for the B.E VI Semester course titled Case Study, is a bonafide record of work carried out by Komali Beeram (1601 18 733 067) and K Tina (1601 18 733 086), during the academic year 2021-2022 under our guidance and supervision.

No part of this project has formed the basis for the reward previously of any other degree, diploma, fellowship, or any other similar title.

Mentor

Sri. B Sateesh
Assistant Professor
Department of Computer Science and Engineering,
Chaitanya Bharathi Institute of Technology.

Supervisor

Dr. Kolla Morarjee
Assistant Professor
Department of Computer Science and Engineering,
Chaitanya Bharathi Institute of Technology.

CHAITANYA BHARATHI
INSTITUTE OF TECHNOLOGY (A)
Affliated to Osmania University

DECLARATION

We, here by, declare that the project report entitled “Automated Attendance System” submitted to Department of Computer Science and Engineering, Chaitanya Bharathi Institute of Technology, in partial fulfilment of the requirements for the B.E VI Semester course titled Case Study, is a bonafide record of work carried out by Komali Beeram (1601 18 733 067) and K Tina (1601 18 733 086), during the academic year 2021-2022 under the guidance and supervision of Sri. B Sateesh and Dr. Kolla Morarjee. We further declare that the work reported in this project has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

Komali Beeram
1601 18 733 067
BE ¾ Computer Science and Engineering
Chaitanya Bharathi Institute of Technology

Hyderabad
22 May 2021

K Tina
1601 18 733 086
BE ¾ Computer Science and Engineering
Chaitanya Bharathi Institute of Technology

ACKNOWLEDGMENT

The success and the outcome of this project required a lot of guidance and assistance from many people and sources, and we are extremely fortunate to have received this all along the completion of the project.

We would like to express our special thanks of gratitude to our supervisor, Dr. Kolla Morarjee, Assistant Professor, Department of Computer Science and Engineering, Chaitanya Bharathi Institute of Technology, for his invaluable guidance, moral support and patience while suggesting the outlines of this project and in meeting the deadline. We thank for his overall support.

We would also like to express our sincere indebtedness towards our mentor Mr. B Sateesh for his approval of the project, and time to time counselling and advice.

We also convey our sincere thanks to the Head of Department of Computer Science and Engineering, Dr. Y Rama Devi, for her constant feedback and assistance.

We also thank our college, Chaitanya Bharathi Institute of Technology, for giving us the opportunity to work on an innovative idea under the course Case Study.

At last, we would like to thank our families and friends for their constant support and guidance from time to time in making this project well above simplicity and into something concrete, despite their busy schedules.

ABSTRACT

Daily attendance marking is a common and important activity in schools and colleges for checking the performance of students. Manual attendance maintenance is a difficult process, especially for a large group of students. Some automated systems which were developed to overcomes these difficulties, have drawbacks like cost, fake attendance, accuracy, intrusiveness. Traditional face recognition systems employ methods to identify a face from the given input, but the results are not usually accurate and precise as desired and are time-consuming. To overcome these drawbacks, there is a need for a smart and automated attendance system.

We aimed to deviate from such traditional systems and introduce a new approach to identify a student using a face recognition system by Multi-Task Cascaded Convolutional Neural Networks, the generation of a facial mode which can further be deployed as an Automated Attendance System in a classroom environment which will be more accurate and bring out faster results when compared to the other competitive machine learning models.

TABLE OF CONTENTS

List of figures

INTRODUCTION

Background

Attendance maintenance is a significant function in all the institutions to monitor the performance of students. Every institute does this in its own way. Some of the institutes use the old paper or file-based systems and some have adopted strategies of automatic attendance using some biometric techniques.

In this modern era of automation many scientific advancements and inventions have taken place to save labour, increase the accuracy and to ameliorate our lives. Automated Attendance System is the advancement that has taken place in the field of automation replacing traditional attendance marking activity. Automated Attendance Systems are generally bio-metric based, smart-card based, and web based. These systems are widely used in different organizations. Traditional method of attendance marking is very time consuming and becomes complicated when the strength is more. Automation of Attendance System has edge over traditional method as it saves time and can be used for security purposes. This also helps to prevent fake attendance.

A facial recognition system is a biometric software which is suited for determining or validating a person by performing comparison on patterns based on their facial appearances. Face recognition systems have upgraded appreciably in their management over the recent years and this technology is now vastly used for various objectives like security and in commercial operations. Face recognition is a powerful field of research which is a computer based digital technology. Face recognition for the intent of marking attendance is a resourceful application of attendance system. It is widely used in security systems and it can be compared with other biometrics such as fingerprint or eye iris recognition systems. As the number of students in an educational institute or employees at an organisation increases, the needs for lecturers or to the organisation also increase the complication of attendance control. This project maybe helpful for the explanation of these types of problems. The number of students present in a lecture hall is observed, each person is identified and then the information about the number of students who are present is maintained.

Problem statement

According to the previous attendance management system, the accuracy of the data collected is the biggest issue. This is because the attendance might not be recorded personally by the original person, in another word, the attendance of a particular person can be taken by a third party without the realization of the institution which violates the accuracy of the data. For example, student A is lazy to attend a particular class, so student B helped him/her to sign for the attendance which in fact student A did not attend the class, but the system overlooked this matter due to no enforcement practiced. Supposing the institution establish an enforcement, it might need to waste a lot of human resource and time which in turn will not be practical at all. Thus, all the recorded attendance in the previous system is not reliable for analysis usage. The second problem of the previous system is where it is too time consuming. Assuming the time taken for a student to sign his/her attendance on a 3-4 paged name list is approximately 1 minute. In 1 hour, only approximately 60 students can sign their attendance which is obviously inefficient and time consuming.

Thus, face recognition student attendance system is proposed 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. Many attendance management systems that already exists lack efficiency and information sharing. Therefore, in this current proposed system, those limitations are overcome and also further improved and made much efficient.

Proposed System

The proposed system will provide login credentials to faculty who are in the university with which they can interact with the system. The student details and face images will be stored in the database.

The proposed system will firstly, capture a picture of face and discern all the faces in it. It then concentrates on one face at a time and understand that even if the face is turned in a strange direction or in bad lighting, it is still the same person. The shadows on face due to low lighting can affect the image but it wont change the person. The next step is to determine various features of the face that can help in distinguishing it from another person. These characteristics could be size of eyes, nose, length of face, skin colour etc. Comparing these distinctive features of that face to all the faces of people we have in database; we will find out the person's name. Our brain, as a human is made to do all these things automatically and instantaneously. Computers are incapable to generalise this kind of high-level techniques, so we need to teach the computer by programming each step of facial recognition separately. Face recognition has three categories: detection, verification, and identification. Face verification is a one-to-one match that compares a face image against a template face images, whose identity is being claimed. On the contrary, face identification is a one-to-many problem that compares a query face image.

Objectives

The following are the objectives of the proposed system:

- To detect the face segment from image.
- To extract the useful features from the face detected.
- To classify the features to recognise the face detected.
- To record the attendance of the identified student.
- To develop a system that is scalable, accurate, efficient, secured.
- To make sure that the speed of the present attendance system is quicker and correct than the previous systems.
- To have enough memory house to store the info so we can retrieve the information at any time.

BACKGROUND INFORMATION

This section documents the available relevant literature concerning the problem domain. The implication was that we devoted sufficient time to reviewing research already taken on related problems. This was done to find out what data and other materials are already available from earlier research and identify gaps that the present research may fill.

Manual Recording

This is the most used for attendance tracking method even before the age of technology. It is a method of recording the student's attendance in the register or in am Excel sheet. Now, as we have adapted to an automated solution, this method has been outdated. The complexities and requirements are increasing day by day. Manual recording of student's presence is no longer a simple solution. Since it is manual recording, it may lead the students giving proxies.

Mechanised systems

When the inputs are provided to the system, it automatically generates a record of presence of students. The student will have to swipe/register every entry into the class. However, an automated system of tracking the student's attendance would be more effective than others. Thus, the manual work that needs to be done is less, and it is more reliable compared to other methods.

Biometrics

It provides verification and identification of the student to the faculty in the institute. The template is captured and saved either on a smart card or database for verification purpose of the registered user. As a result, identification of users is done from the biometric characteristics alone without using smart cards, usernames, or id etc. All the records in a database are compared to the template and the closest match score is returned. Therefore, authenticating the closest match within the allowed threshold.

There are two different classes in biometrics:

- Physiological
- Behavioural

Physiological biometrics include capturing and comparing the fingerprints, hand, face, iris etc., of the person. The most widely used physiological biometrics is fingerprints.

Behavioural biometrics related to the behaviour of the person which includes recognising voice, signature, and keystrokes etc. [2]

Accordingly, biometrics is used in organisation for physiological verification of fingerprints and retinal recognition.[1] [3]

Access cards/ swipe cards

These cards are used to provide access to the user after swiping the card. It stores the data on a magnetic stripe, and it is also called a magnetic stripe card. [4]

Proximity card readers

They are contactless integrated circuit devices used for secured access. Proximity cards/ proxy cards have a range of 0-3 inches, and it allows the user to leave the card in a wallet or purse, etc. therefore, this technology is used in applications such as ID cards, key cards etc.

Existing Face Recognition Algorithms

The following are the existing algorithms:

- ***Eigen face***
  This algorithm extracts the necessary information from an image and efficiently encodes it. To obtain variations, several pictures of a single person is taken. For the set of images of faces, eigenvectors and its covariance matrix are calculated and stored. Since every image represents an eigen vector, the data set helps produce variety for the system. A representation of these eigen vectors is called eigen faces. [5]
- ***Line Edge Map***
  One of the popular methods is using the Line Edge Maps algorithm. In this method, line matching is done to map the features of the face. This algorithm mainly uses the most prominent features of the face; mainly the eyes, nose and mouth having high characteristics. The colour images are converted to greyscales to observe and extract the similarities in the faces. Sobel edge detection algorithm is made use of to encode the greyscale images into binary edge maps. This technique was developed by studying how we human beings remember other people faces.
- ***Histogram of oriented gradients (HOG)***
  This technique can be applied to detecting objects as well as faces. All images used are converted to greyscale and every pixel in this image is assigned an integer. Every pixel compares its value to its neighbouring pixels. The primary motive is to find the dark regions of the face in the image. The direction pointing to that dark region will have a white arrow pointing towards it. This treatment is done for each pixel of the picture. [6]

Identified Gaps

- Lack of efficiency
- Time consuming
- Lack of robust and error free systems
- Fail to recognise faces for primal issues like scaling, pose, illumination, rotations, occlusions.
- Less accuracy.
- Detection process is slow, and computation is complex in terms of neural networks.
- Long training time.
- Large database is required to achieve high accuracy.

SCOPE

Introduction

This system will be an automated attendance system using facial recognition for Chaitanya Bharathi Institute of Technology. This system will be designed to facilitate the attendance system of students by providing tools to assist in automating the manual attendance system. More specifically, this system is designed to allow the faculty to manage and control by taking pictures of students who attended the class for collecting their daily attendance. The system utilizes video and image processing to provide inputs to the system. The main aim of this system is to dissolve the traditional method of manual attendance and overcome the previous existing systems of biometric attendance, and various other face recognition models which have drawbacks like cost, proxies, accuracy, intrusiveness, and speed.

The goal is to provide a professor with an easy, effective, accurate, cost and time efficient portable solution to attendance recording and maintenance.

Product Perspective

The Automated Attendance System is intended to replace the manual mode of attendance, the biometric system of attendance and the existing face recognition-based attendance systems. The roll calls for students shall be replaced with a single interaction between the professor and the system. Professors must just upload a picture of the class and the attendance of students will be reflected in the excel sheet.

Product Functions

The professor will have to upload an image of the class students faces and after the face detection and face recognition of the faces from the database stored, an excel sheet will be created with a list of students present and absent for that class/day.

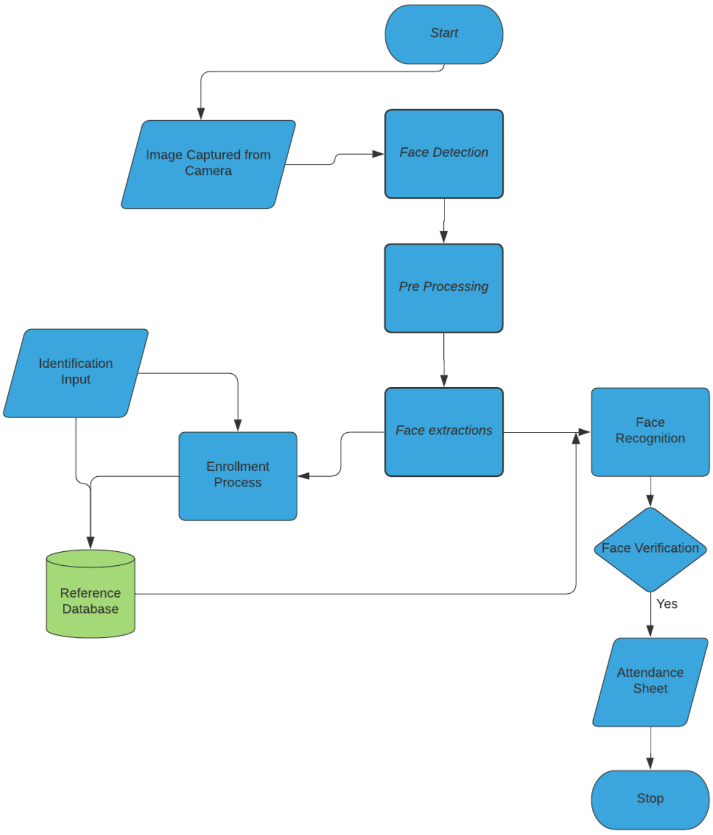


Figure 1 Flow Chart

User Classes and Characteristics

The users of this system will be able to retrieve an excel sheet of students present and absent. The system for now only supports a single user that is professors who must upload a picture of the class students to get the list of students present. The admin will also be able to add/enroll new students faces and details to the database.

System Features

Enrolling New Student

- Description and Priority

  The main function of this feature is to enroll a new student by uploading a picture of his/her to the database to use it in the future for reference to recognize the student in class for attendance. The priority for this feature should be given first.

- Stimulus/Response Sequence

  Once the picture of the student is uploaded, a messaged will be displayed, and the details of the student will be stored in the database.

- Functional Requirements

  To add a new student to the system, a latest picture of them is a must. Any other requirements are yet to be determined.

Attendance System

- Description and Priority

  The main function of this feature is recognizing every face present in the class through camera or a picture of the class. This is another topmost priority feature.

- Stimulus/Response Sequence

  Once the picture of the class is uploaded, the excel sheet of the attendance report gets updated.

- Functional Requirements

  The pictures and details of the students should be already present in the database. The other functional requirements are yet to be determined.

DEVELOPMENT METHODOLOGY

The smooth functioning of the project depended on the development of multiple components; the following sections will briefly discuss each one. The development methodology used is the waterfall. This included the analysis, design, implementation, and the testing steps.

Analysis

In this case we analysed the requirements, and fully understood the problems. Analysis was conducted on the current systems failures and strengths. This allowed a better understanding of the expected improvements. Further analysis was also conducted on the problem definitions to clearly understand what to tackle. This phase is usually accompanied by documentation for each requirement, which enabled us to review it for validation.

Logical Design

Logical design characteristically looked at the intended system from a logical perspective without considering physical requirement. The project needed a logical design that modelled the flow of data and information through the system from input to output. Logical design also modelled the security checks that the system will be using as well as the formats for all the data items in the system.

Physical Design

The physical design is concerned with the physical architecture of the entire system interacted to achieve its objectives. It modelled the testing, training, user interfaces and the database models.

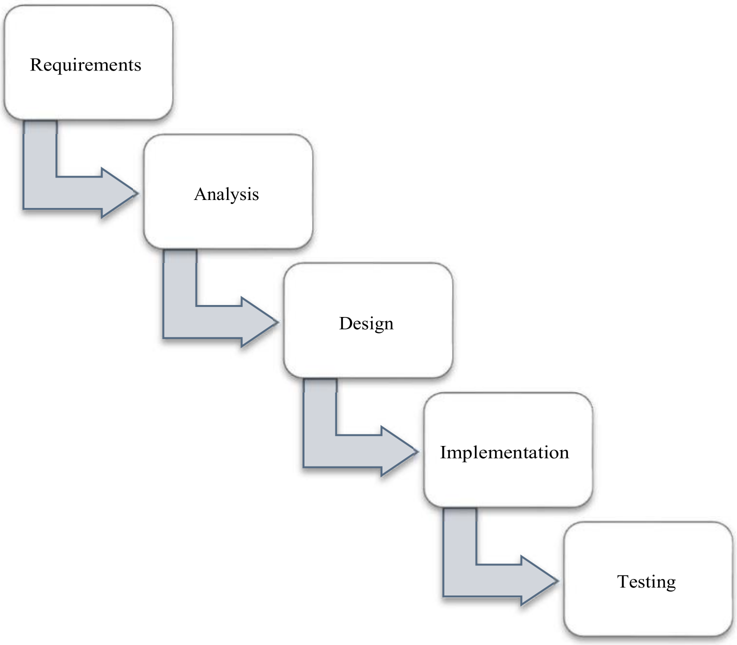


Figure 2 Waterfall Development Methodology

SYSTEM DESIGN

We used the design documents, personas, user case scenarios etc. to run comprehensive tests including components testing and on the finished applications.

Data Dictionary

A data dictionary contains metadata i.e., data about the database. The data dictionary is very important as it contains information such as what is in the database, who can access it, where is the database physically stored, when and where database is created, database table constraints such as primary key attributes, foreign key information etc. The users of the database normally do not interact with the data dictionary, it is only handled by the database administrators.

FIELD NAME	DATA TYPE	NULL	CONSTRAINT	DESCRIPTION
student_id	int(auto_increment)	NO	PRIMARY KEY	Unique ID of student.
student_name	varchar(50)	NO	LENGTH>0	Full name of the student.
student_email	varchar(50)	NO	UNIQUE	Email address of the student.
student_photo	image	NO	UNIQUE	A photograph of student face.
Date	date	NO	LENGTH>0	Date of the attendance.
Attendance	varchar(10)	NO	Present or Absent	Attendance of the student.
faculty_id	int(auto_increment)	NO	PRIMARY KEY	Unique ID of faculty.
faculty_name	varchar(50)	NO	LENGTH>0	Full name of the faculty.

Data Flow Diagram

It is the graphical representation of the flow of data between various processes through an information system. It differs from flowchart as it shows the data flow instead of control of program. It can also be used for visualisation of data programming.

DFDs are drawn at different levels. Level 0 shows the main process in the system. Further levels describe the subprocesses and the development in the system.

All the processes must have at least 1 data flow in and out; should modify the incoming data producing new forms of outgoing data. Each data stored must be involved with at least one data flow. Each external entity must be involved with at least one data flow. The data flow must be attached to at least one process.

A circle or bubble shows a process that transforms data inputs into data outputs. A curved line with arrowhead shows the flow of data into or out of a process or data store. A set of parallel lines shows a place for the collection of data items. A data store indicates that the data is stored which can be used at a later stage or by the other processes in a different order. The data store can have an element or group of elements. Source or sink is an external entity and acts as a source of system inputs or sink of system outputs which is represented by a rectangle.

NAME	**SYMBOL**
Process	○
Data flow	⟶
Data store	═
Source or sink	▭

Level 0

This is the fundamental system model or context diagram that represents the entire software requirement as a single bubble with input and output data denoted by incoming and outgoing arrows. Then the system is decomposed and described as a DFD with multiple bubbles. Parts of the system represented by each of these bubbles are then decomposed and documented as more and more detailed DFDs. The below diagram depicts how students, faculty can access the system and how a student can give attendance through camera and both faculty and students can view the attendance report.

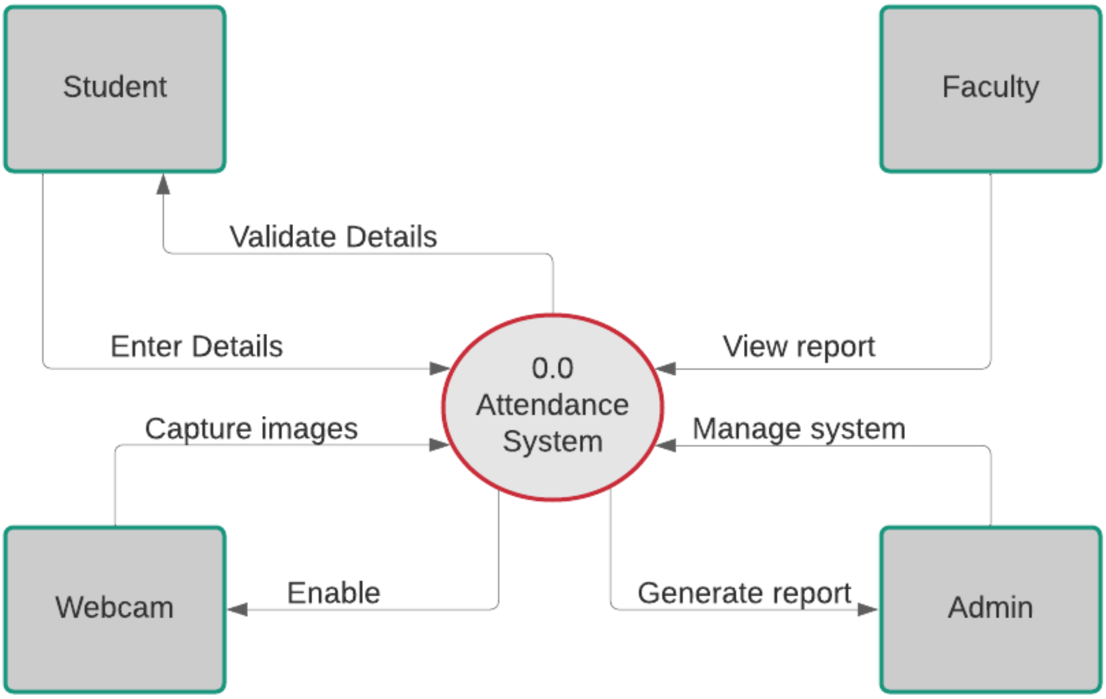


Figure 3 Data Flow Diagram Level 0

Level 1

In Level 1 DFD, a context diagram is decomposed into multiple bubbles/processes. In this level, the main objectives of the system are broken down to high level process of Level 0 DFD into subprocesses. It also projects or records the specific/necessary detail about the system's functioning.

The below DFD shows how a student login into the student, and how the web camera is enabled, and the student pictures are captured and then the image acquisition, detection, pre processing and recognition is done with the help of student database. The diagram also illustrates the activities or functions, or modules managed by faculty and admin.

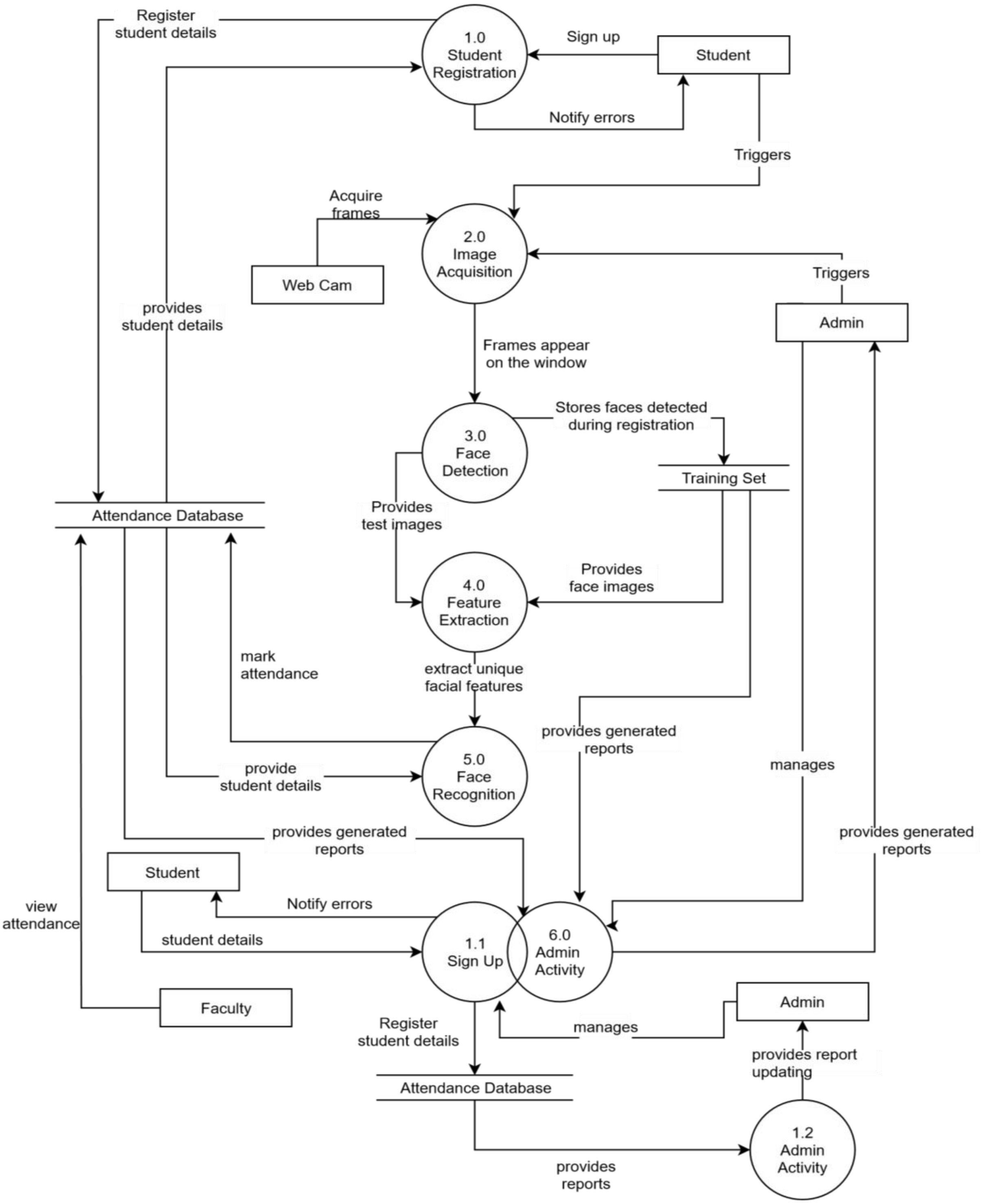


Figure 4 Data Flow Diagram Level 1

Entity Relationship Diagram

It displays the relationship of entity sets stored in a database. In other words, ER diagrams help to explain the logical structure of databases. They are created based on three basic concepts: entities, attributes, and relationships. It is used to represent the entity framework infrastructure.

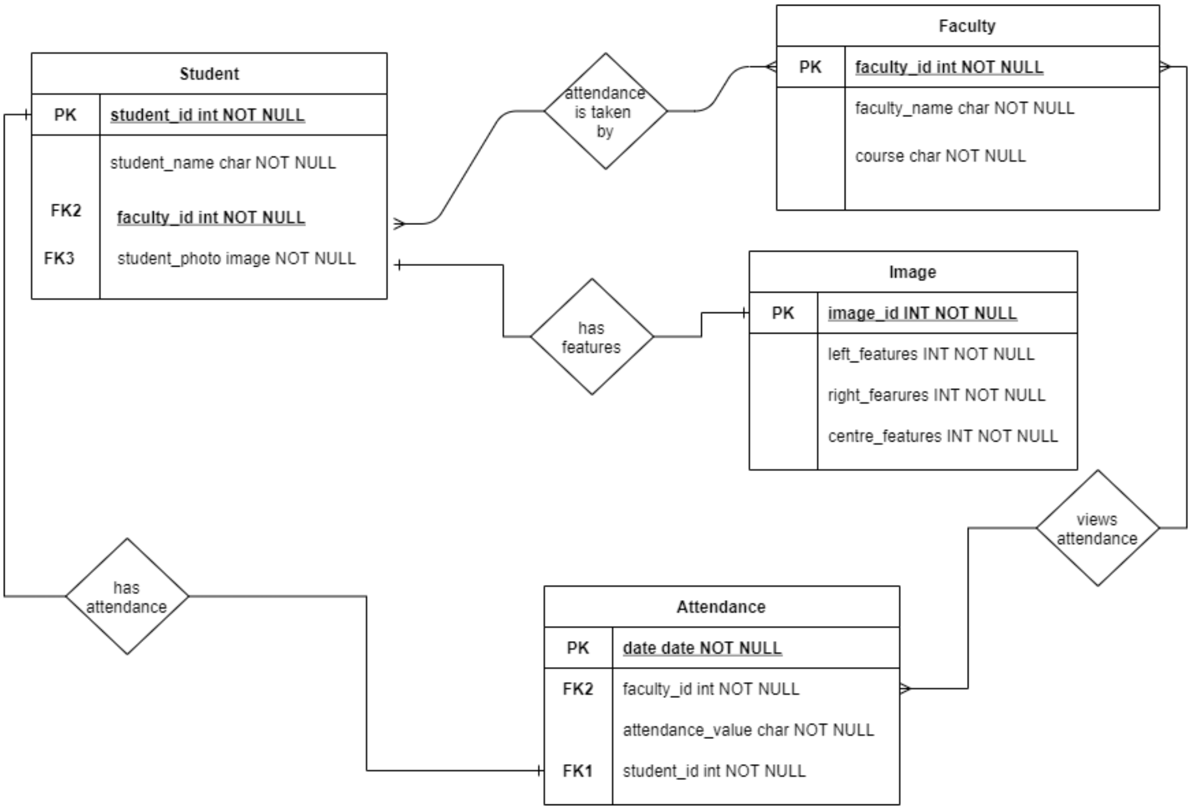


Figure 5 Entity-Relation Diagram

Use Case Diagram

Use case diagram is dynamic in nature and there are some internal or external factors for making the interactions. These internal and external agents are known as actors. Use case diagrams consists of actors, use cases and their relationships. The diagram is used to model the system/subsystem of an application. A single use case diagram captures a particular functionality of a system. Hence to model the entire system, several use case diagrams are used.

Use case diagrams are used to gather the requirements of a system including internal and external influences. These requirements are mostly design requirements. Hence, when a system is analysed to gather its functionalities, use cases are prepared and actors are identified. When the initial task is completed, use case diagrams are modelled to present the outside view.

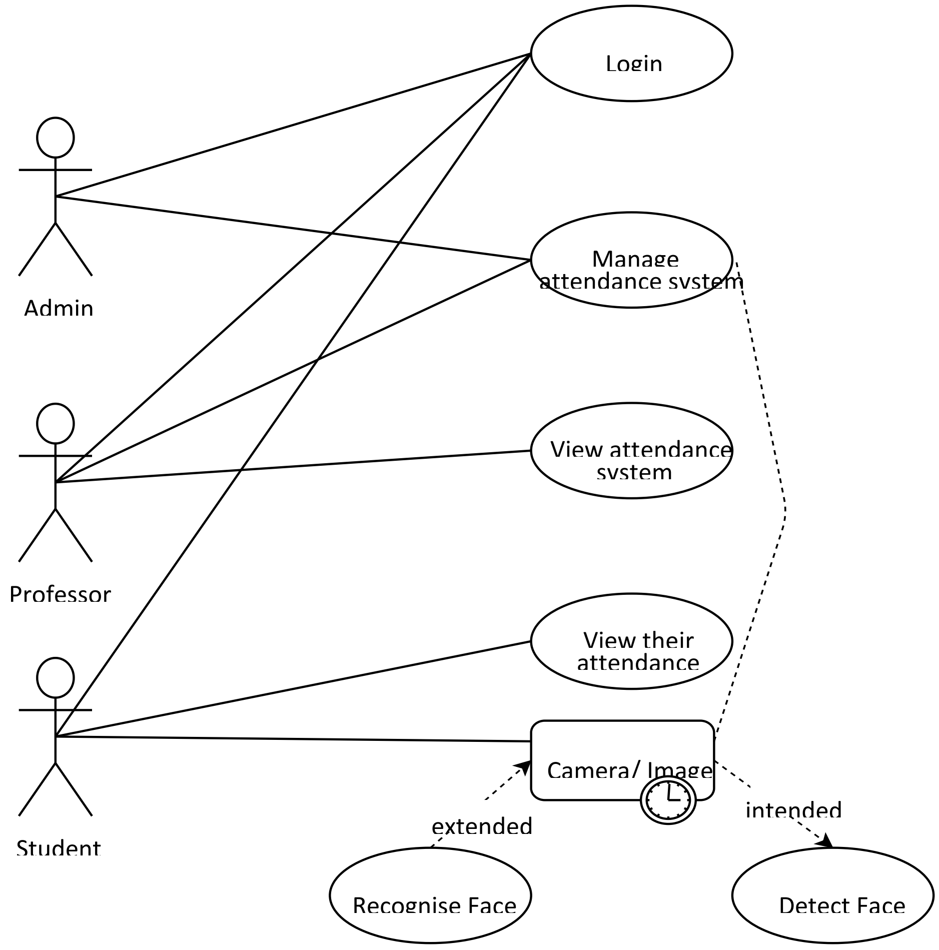


Figure 6 Use Case Diagram

Class Diagram

Class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application.

Class diagrams describes the attributes and operations of a class and the constraints imposed on the system. The class diagrams are widely used in the modelling of object-oriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages. Class diagrams shows a collection of classes, interfaces, associations, collaborations, and constraints. It is also known as a structural diagram.

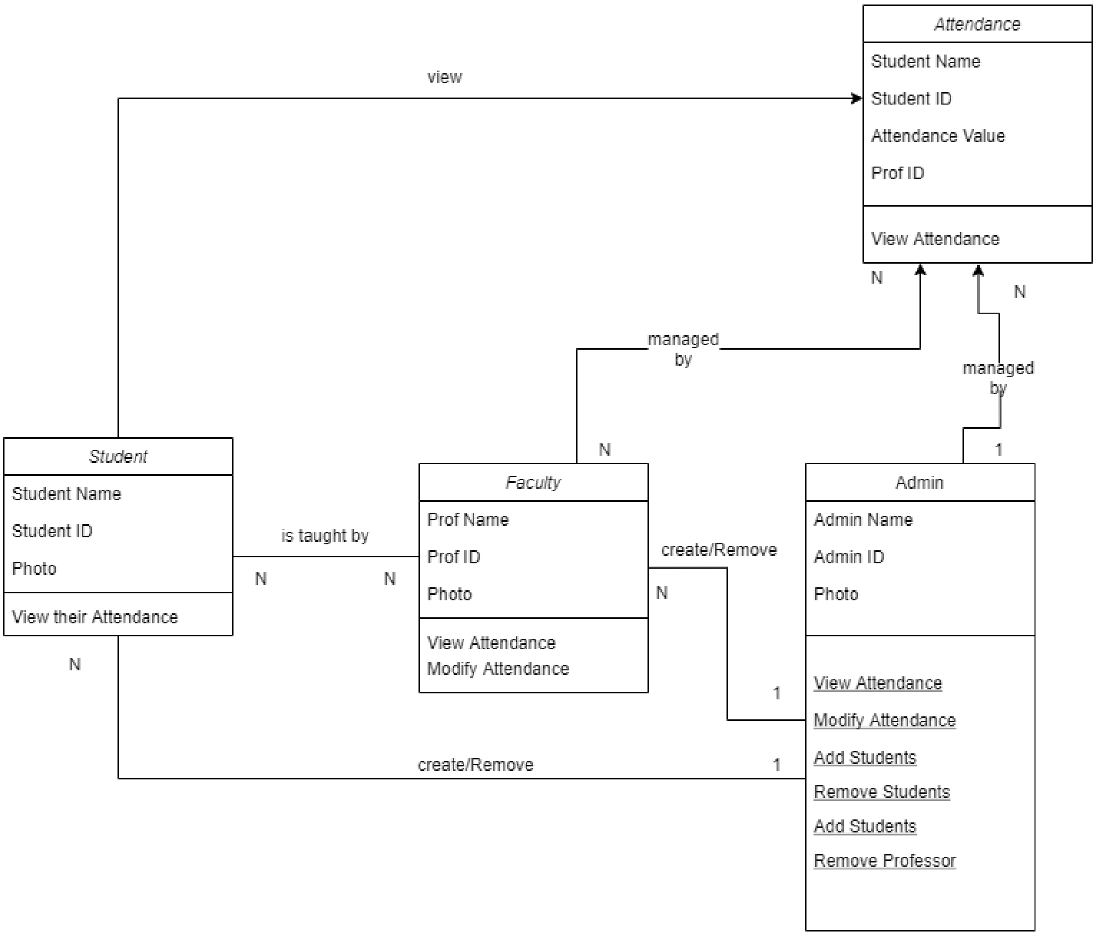


Figure 7 Class Diagram

Sequence Diagram

The diagram depicts interaction between objects in a sequential order i.e., the order in which these interactions take place. Sequence diagrams describe how and in what order the objects in a system function.

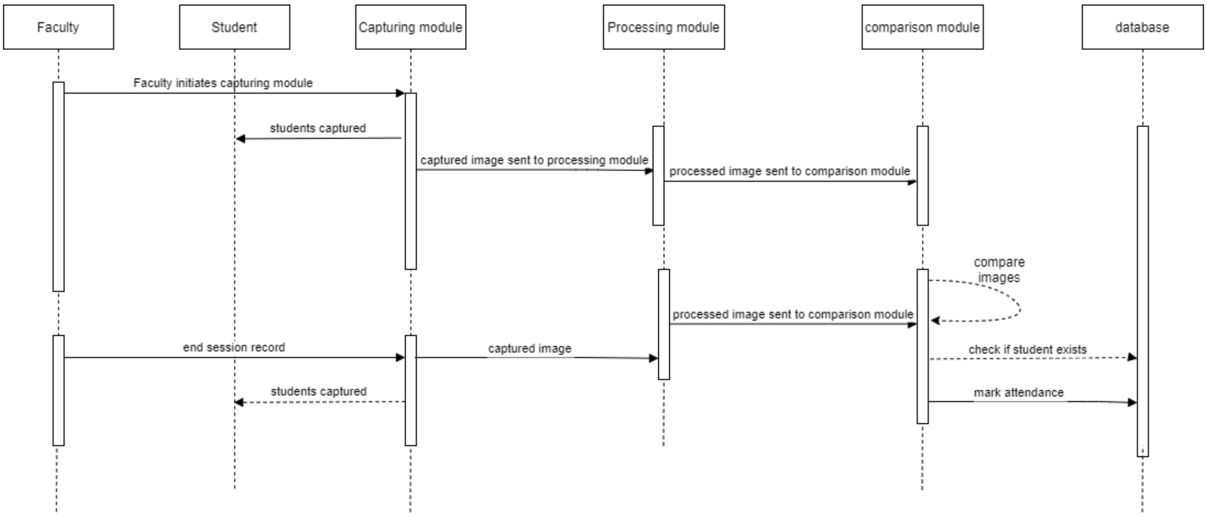


Figure 8 Sequence Diagram

Collaboration Diagram

A collaboration diagram, also known as a communication diagram, is an illustration of the relationships and interactions among software objects in the Unified Modelling Language (UML). These diagrams can be used to portray the dynamic behaviour of a particular use case and define the role of each object.

Collaboration diagrams are created by first identifying the structural elements required to carry out the functionality of an interaction. A model is then built using the relationships between those elements.

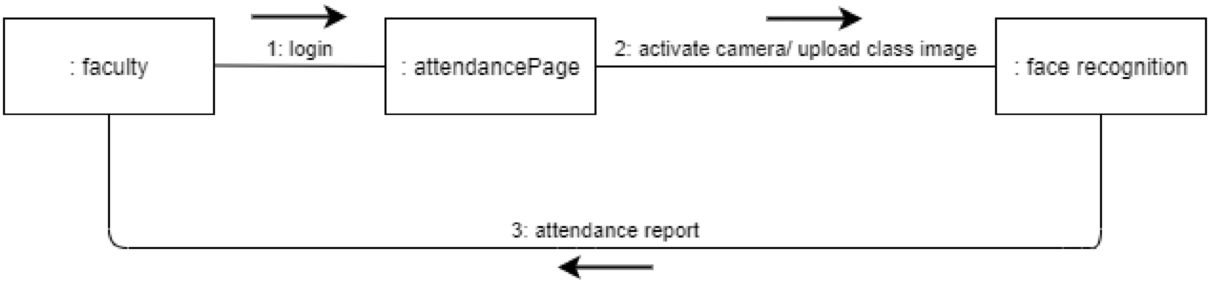


Figure 9 Collaboration Diagram

Activity Diagram

Activity diagrams illustrates the flow of control in a system and refer to the steps involved in the execution of a use case. We model the sequential and concurrent activities using activity diagrams. An activity diagram focuses on condition of flow and the sequence in which it happens.

It portrays the control flow from a start point to a finish point showing the various decision paths that exist while the activity is being executed. We can depict both sequential processing and concurrent processing of activities.

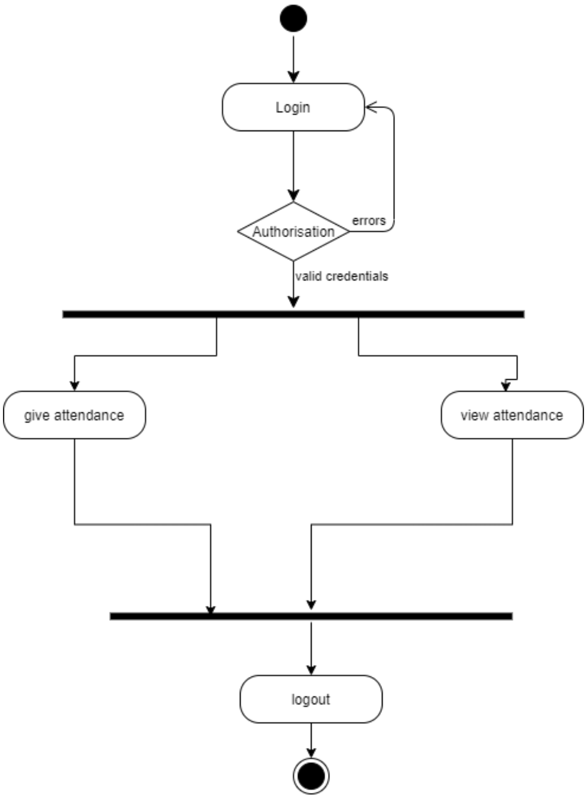


Figure 10 Student Activity Diagram

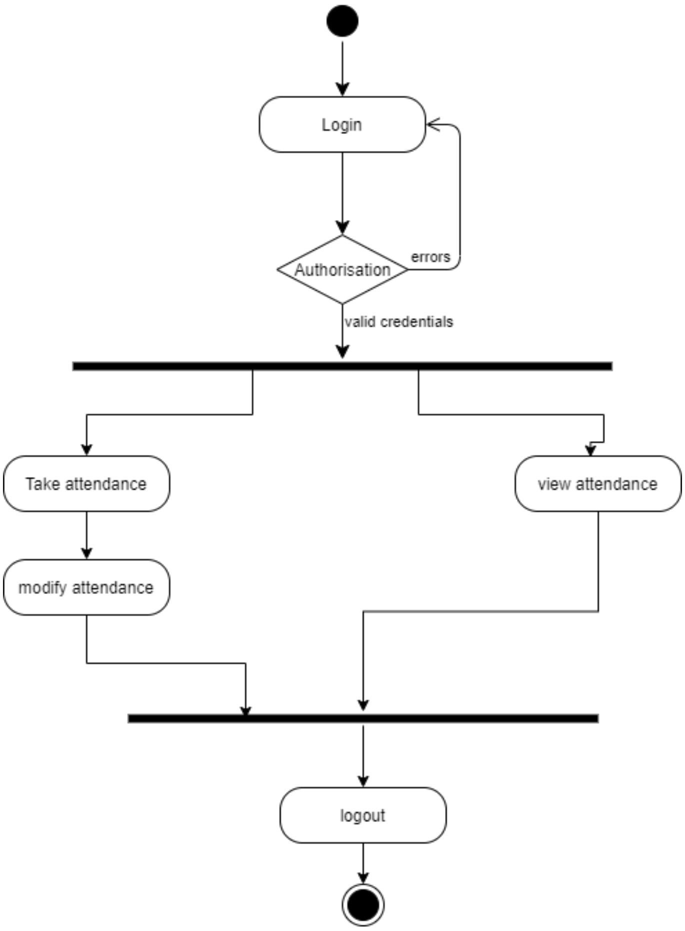


Figure 11 Faculty Activity Diagram

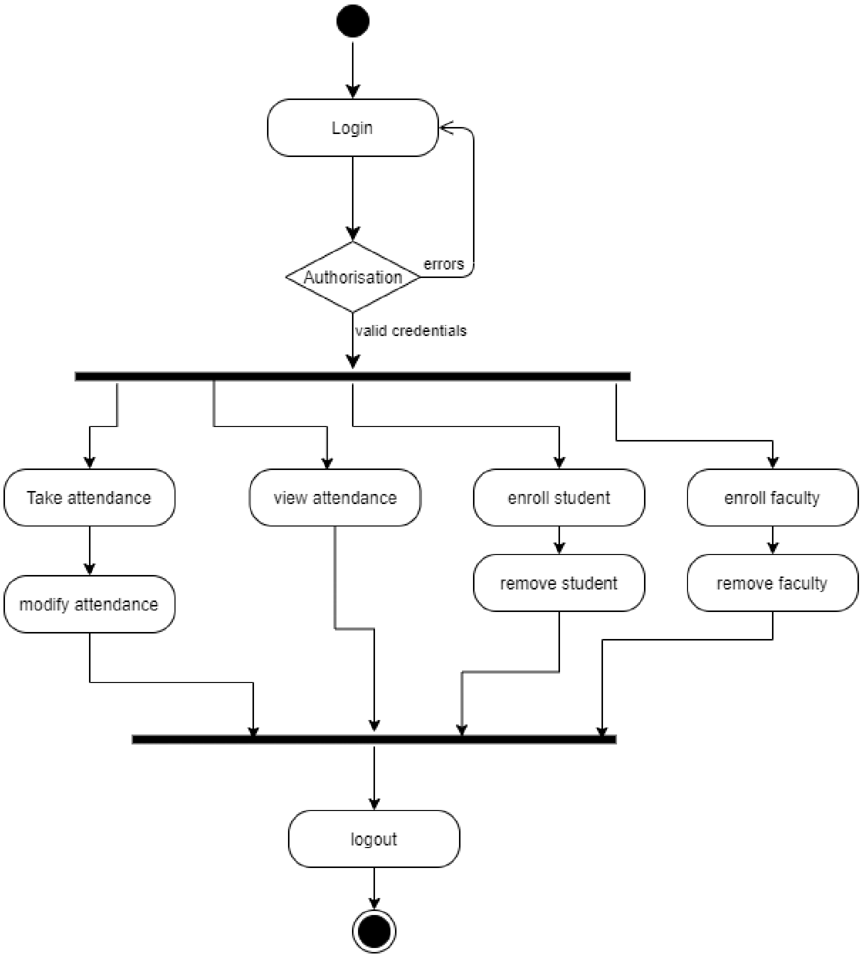


Figure 12 Admin Activity Diagram

State Chart Diagram

A state chart diagram is used to represent the condition of the system or part of the system at finite instances of time. It is a behavioural diagram, and it represents the behaviour using finite state transitions. It is used to state the events responsible for change in state. We do not show what processes cause those events. We use it to model the dynamic behaviour of the system.

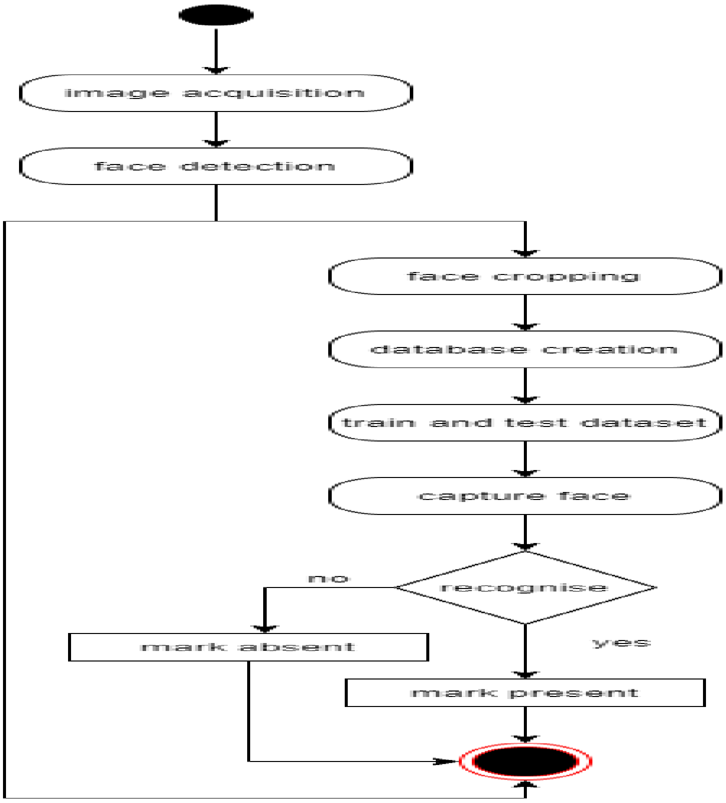


Figure 13 State Chart Diagram

Component Diagram

Component diagrams are used to visualise the organisation and relationships among components in a system. These diagrams are also used to make executable systems. It does not describe the functionality of the system, but it describes the components used to make those functionalities. Thus, from that point of view, component diagrams are used to visualise the physical components in a system. These components are libraries, packages, files, etc. they can also be described as a static implementation view of a system. Static implementation represents the organisation of the components at a particular moment.

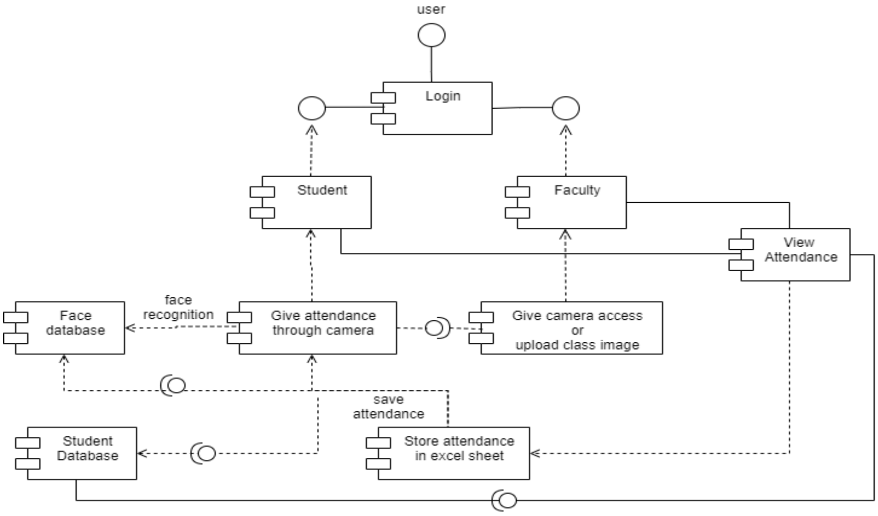

user
Login
Student
Faculty
View Attendance
Face database
face recognition
Give attendance through camera
Give camera access or upload class image
save attendance
Student Database
Store attendance in excel sheet

Figure 14 Component Diagram

Deployment Diagram

A deployment diagram is a UML diagram type that shows the execution architecture of a system, including nodes such as hardware or software execution environments, and the middleware connecting them.

Deployment diagrams are typically used to visualise the physical hardware and software of a system. Using it we can understand how the system will be physically deployed on the hardware.

Deployment diagrams help model the hardware topology of a system compared to other UML diagram types which mostly outline the logical components of a system.

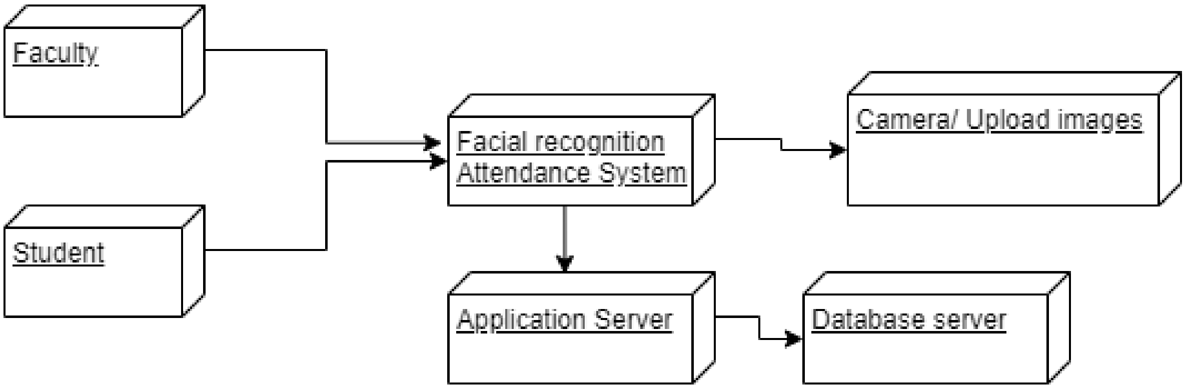
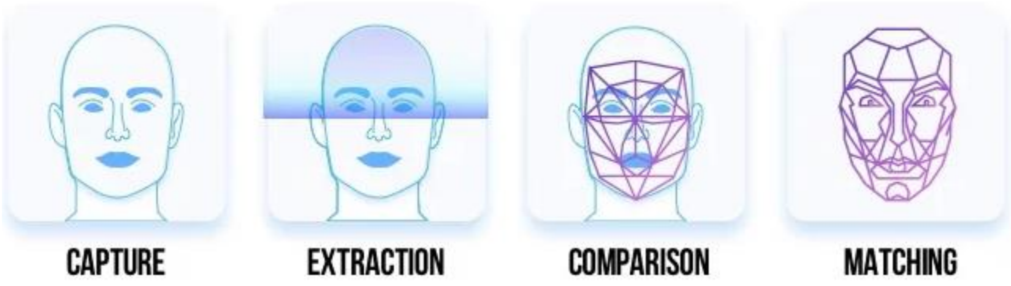


Figure 15 Deployment Diagram

IMPLEMENTATION

Once the designs are deemed to be viable, technical implementation begins. Implementing the project was the toughest part as all the coding was done in this phase. The complete automated attendance system can be divided into four categories: face detection, feature extraction, feature matching and marking attendance.



Database Creation

For the database, a set of at least 10 pictures with various sizes, colour, lighting, angles are taken and stored in a folder under their name.

Face Detection

The face detection method is used to find the faces present in the image, extract the faces, display, and create a compressed file to use it further for feature extraction.

Methods used in Face Detection:

- ***HAAR CASACADE FACE DETECTION:***
  This method has a simple architecture that works nearly real-time on CPU. Also, it can detect images at different scales. But the major drawback is that it gives false results as well as it does not work on non-frontal images. [7]
- ***DLIB (HOG) FACE DETECTION:***
  It is the fastest method on CPU which can work on frontal and slightly no-frontal images. But it is incapable of detecting small images and handling occlusions. Also, it often excludes some part of the chin and forehead while detection. [8]
- ***DLIB (CNN) FACE DETECTION:***
  It works very fast on GPU and is capable to work for various face orientations in images. It can also handle occlusions. But the major disadvantage is that it is trained on a minimum face size of 80*80 so it can not detect small faces in images. It is also very slow on the CPU. [8]
- ***MTCNN:***
  This method gives the most accurate results out of all the four methods. It works for faces having various orientations in images and can detect faces across various scales. It can even handle occlusions. It does not hold any major drawback as such but is comparatively slower than HOG and Haar Cascade Method. [9]

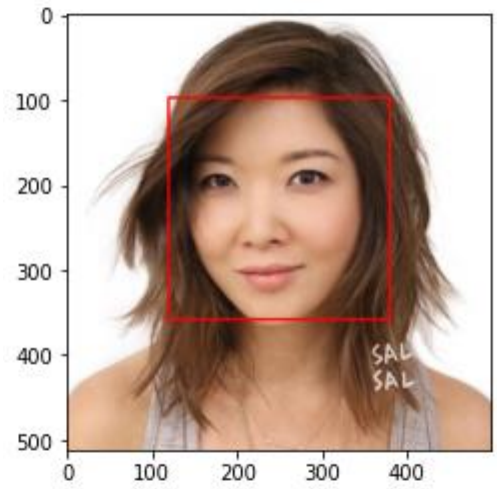


Figure 16 Haar Cascade

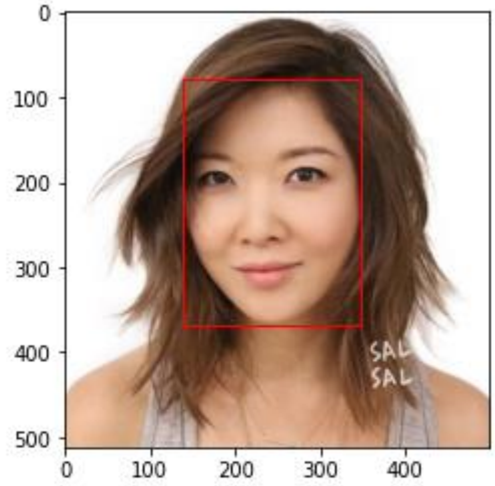


Figure 17 MTCNN

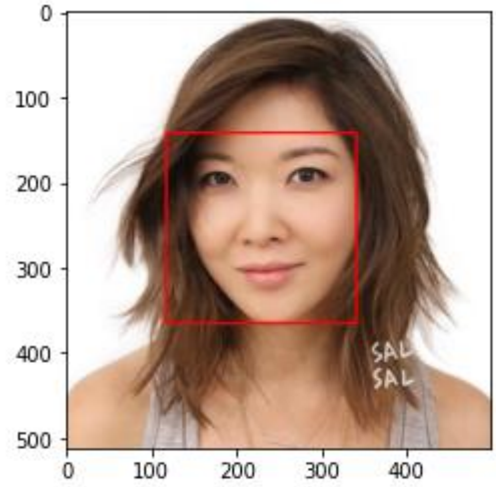


Figure 18 DLIB(HOG)

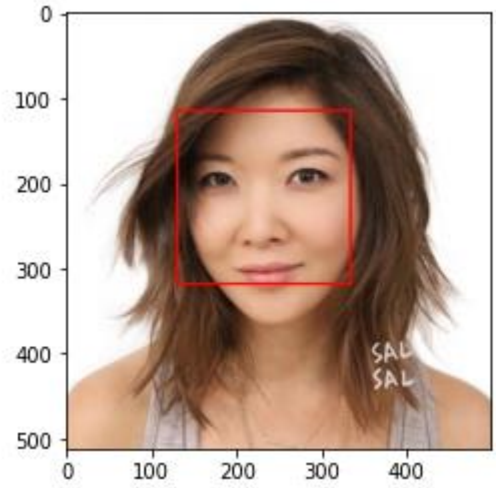
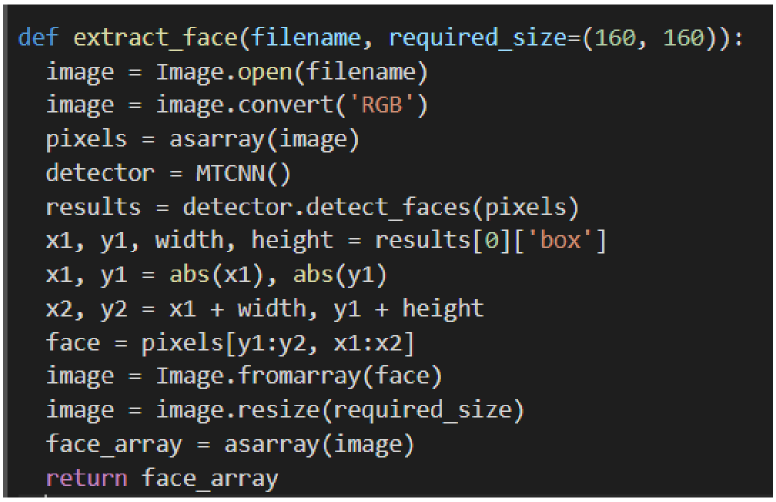


Figure 19 DLIB(CNN)

We used the MTCNN library to create a face detector and extract faces for our use with FaceNet face detector models in subsequent sections.



```
def extract_face(filename, required_size=(160, 160)):
  image = Image.open(filename)
  image = image.convert('RGB')
  pixels = asarray(image)
  detector = MTCNN()
  results = detector.detect_faces(pixels)
  x1, y1, width, height = results[0]['box']
  x1, y1 = abs(x1), abs(y1)
  x2, y2 = x1 + width, y1 + height
  face = pixels[y1:y2, x1:x2]
  image = Image.fromarray(face)
  image = image.resize(required_size)
  face_array = asarray(image)
  return face_array
```

Figure 20 Face Detection Code

Face Extraction

Face extraction is the basic and most important initializing step for face recognition. It extracts the biological components of face. These biological components are the features of face which differ from person to person. There are various methods which extract various combination of features, commonly known as nodal point. No two people can have all the nodal points like each other except for identical twins. Each person's face has about 80 nodal points which, when measured by facial recognition software, create a faceprint based on a numeric code representing a face in a database. Some features measured by the software are depth of the sockets, distance between the eyes, shape of the cheekbones, width of nose and length of the jawline.

Facial feature extraction has two approaches:

- **Shallow Approach**
    1. ***PCA***
       PCA is used to reduce the dimensionality of the data. In PCA, the original features of dataset will be converted into a linear combination of uncorrelated variables (features). These combinations are known as Principal Components. PCA increases algorithm performance and improves visualisation. PCA results in loss of information if the number of Principle Components is not selected wisely. [10] [11]
    2. ***LDA***
       It is a dimensionality reduction technique used to classify different classes based on the features of the supervised data. The major drawback of LDA is the so-called Small Sample Size Problem and non-linearity problem. SSS problem occurs when the sample is quite small as compared to the dimension of the data.[11]
    3. ***Cosine Similarity***
       The measure of cosine angle between two vectors is known as the cosine similarity between two vectors, the closer the cosine value to 1 and greater will be the possibility of a match. One vector among the two vectors is the test data (detected face) and the other is the vector of the training dataset. But it gives false results for sparse numeric data. [12]
    4. ***HOG***
       It only uses magnitude values of pixels without including the neighbouring values which lead to the extraction of improper features during image rotation.
    5. ***SIFT***
       SIFT is relatively alike to a sparse descriptor. Sparse Descriptor is a technique that initially detects the key points in the image and then generates descriptors at these points. SIFT consists of scale rotation and affine transformation properties as well. But it requires a long-running time as compared to other systems.
- **Deep Approach**
    1. ***VGG***
       VGG uses various architectures such as VGGFace1, VGGFace2 by Keras. The basic difference among these models is the number of layers included in its architecture that varies from model to model. These models have quite a good accuracy.[14]
    2. ***Face Recognition API***
       FaceRecognition API is easier to use. It has a much easier architecture to implement with some inbuilt libraries required for feature recognition. You need to upload a picture and call the FaceRecognition API. The API then simulates the browser using

the user's information to call the recognition points. It works well in real-time and holds good accuracy. [13]

3. ***FaceNet Keras***
   FaceNet Keras is a one-shot learning model. It fetches 128 vector embeddings as a feature extractor. It is even preferable in cases where we have a scarcity of datasets. It consists of good accuracy even for such situations.[15]

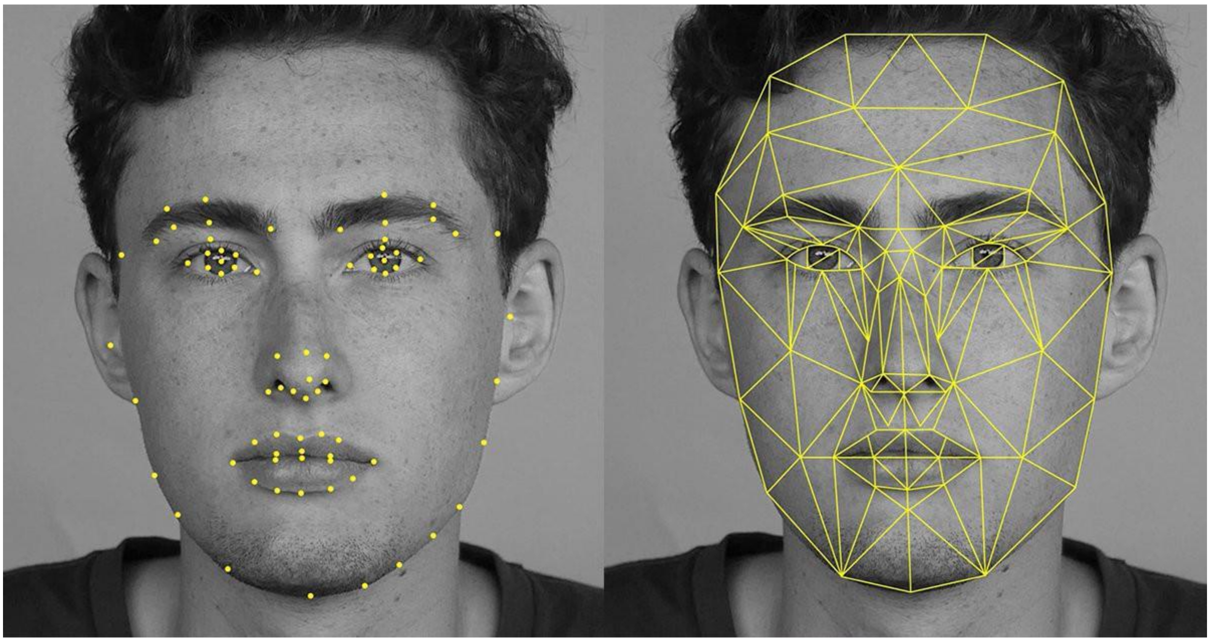
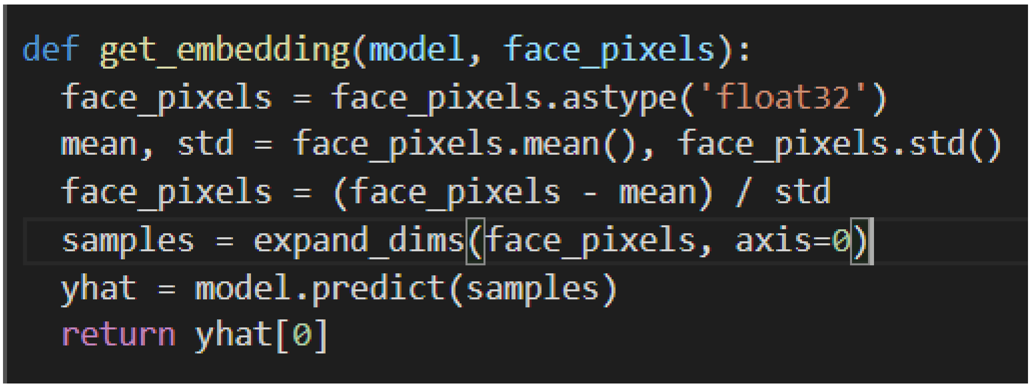


Figure 21 128 Face Embeddings

The FaceNet models is used as part of the classifier itself, in other words, we used it to pre-process a face to create a face embedding that is stored and used as input to the classifier model. The below function transforms the dataset into 93 face embeddings, each comprised of a 128-element vector.



```
def get_embedding(model, face_pixels):
  face_pixels = face_pixels.astype('float32')
  mean, std = face_pixels.mean(), face_pixels.std()
  face_pixels = (face_pixels - mean) / std
  samples = expand_dims(face_pixels, axis=0)
  yhat = model.predict(samples)
  return yhat[0]
```

Figure 22 Feature Extraction using FaceNet Keras Code

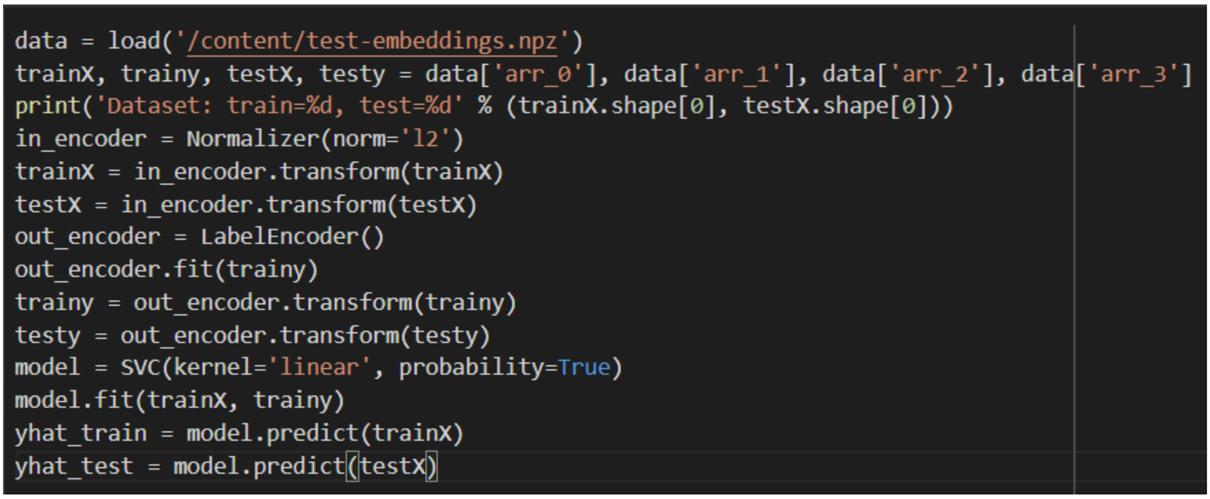
Face Classification

Face classification is a geometry-based or template-based algorithm used to classify the features of the test data among different classes of facial features in the training data. These template-based classifications are possible using various statistical approaches.

The well-known methods used in feature classification can be given as:

- ***Euclidean Distance***
  It is a distance-based feature classification method that calculates the distance between the facial nodes and the face which has the minimum difference between these distance values is the match. But it is suitable for the datasets having a smaller number of classes and lower dimensionality features.
- ***Cosine Similarity***
  In cosine similarity, the solution that we obtain after calculating the cosine of an angle is brought into concern. Here, we would compare the differences between these results. The more the value is closer to 1, the greater is the probability of the match. But it may give a false result if the test data features are incomplete (i.e., if the resultant value is 0 then the features do not match, and if nearly all the features match, then the value is 1). [12]
- ***SVM*** [16]
  SVM (Support vector machine) creates an optimal hyperplane to classify the classes of training dataset based on the different features of the face. The dimensionality of the hyperplane is one less than the number of features. Different kernels can be applied to see what features are used by the classifier to remove the features if required. This can help to improve speed.
- ***KNN***
  KNN (K-Nearest Neighbour) is all about the number of neighbours i.e., the k value. In KNN, if k=3 then we check that the data is close to which 3 data points. Thereafter, it is decided that closest data points belong to which class. Now, the test data is predicted to be in this class KNN has curse of dimensionality problem which can be solved by applying PCA before using KNN classifier.
- ***ANN***
  ANN (Artificial Neural Network) uses a very detailed algorithm for face recognition. It classifies the local texture using multi-layer perceptron for face alignment. It uses geometric feature based and independent component analysis for feature extraction and multi artificial neural network for feature matching. This is the best method to use.

Since we were working on normalised face embedding inputs, we have used the Linear Support Vector Machine. This is because the method is very effective at separating the face embedding vectors.

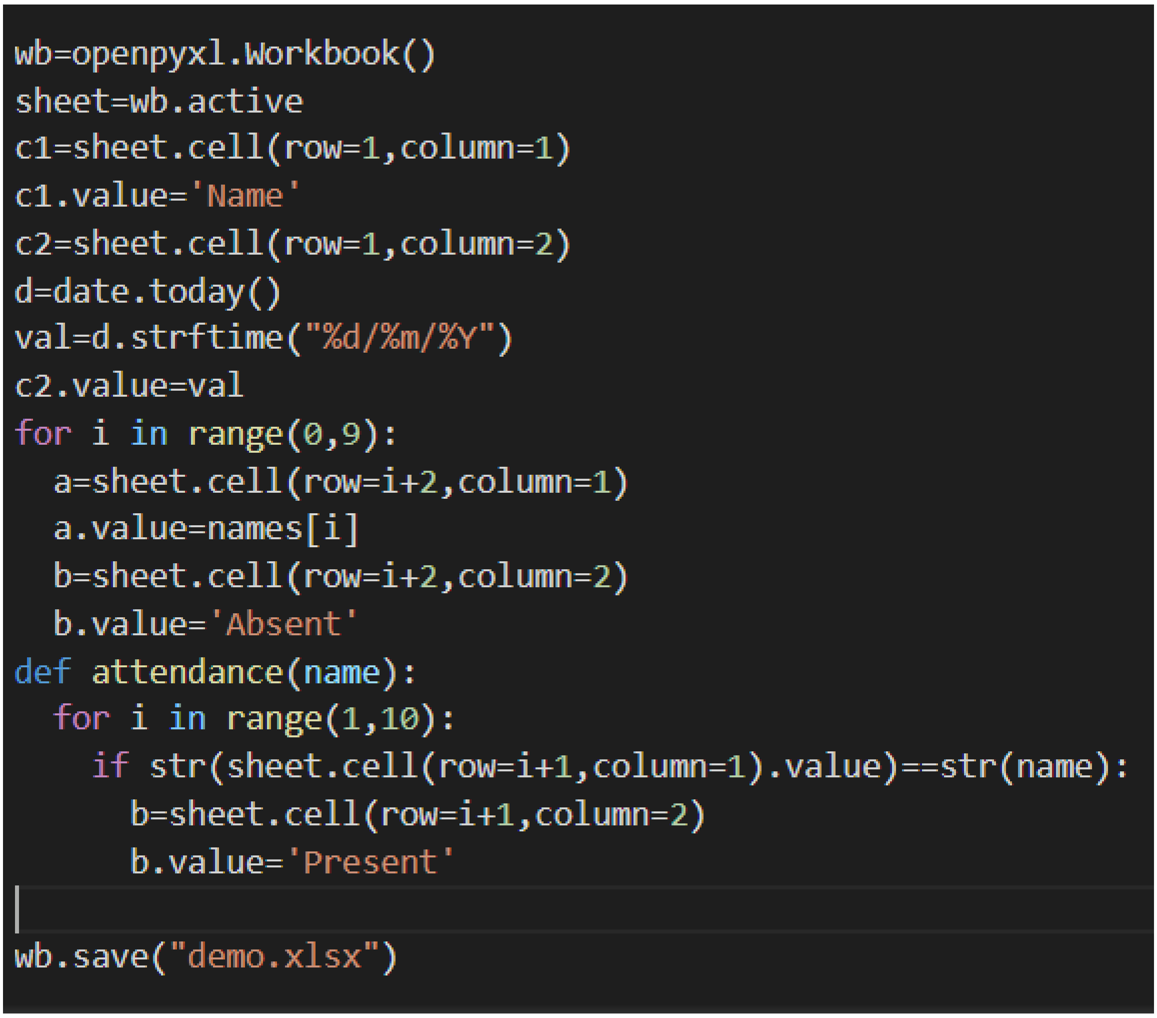


```
data = load('/content/test-embeddings.npz')
trainX, trainy, testX, testy = data['arr_0'], data['arr_1'], data['arr_2'], data['arr_3']
print('Dataset: train=%d, test=%d' % (trainX.shape[0], testX.shape[0]))
in_encoder = Normalizer(norm='l2')
trainX = in_encoder.transform(trainX)
testX = in_encoder.transform(testX)
out_encoder = LabelEncoder()
out_encoder.fit(trainy)
trainy = out_encoder.transform(trainy)
testy = out_encoder.transform(testy)
model = SVC(kernel='linear', probability=True)
model.fit(trainX, trainy)
yhat_train = model.predict(trainX)
yhat_test = model.predict(testX)
```

Figure 23 Feature Classification Code

Attendance Marking

The attendance is marked in an excel sheet which updates automatically with a different date added to a new column in the sheet.



```python
wb=openpyxl.Workbook()
sheet=wb.active
c1=sheet.cell(row=1,column=1)
c1.value='Name'
c2=sheet.cell(row=1,column=2)
d=date.today()
val=d.strftime("%d/%m/%Y")
c2.value=val
for i in range(0,9):
  a=sheet.cell(row=i+2,column=1)
  a.value=names[i]
  b=sheet.cell(row=i+2,column=2)
  b.value='Absent'
def attendance(name):
  for i in range(1,10):
    if str(sheet.cell(row=i+1,column=1).value)==str(name):
      b=sheet.cell(row=i+1,column=2)
      b.value='Present'

wb.save("demo.xlsx")
```

Figure 24 Attendance Marking Code

RESULT

This section discusses the result of automated attendance system. The system is tested using a computer with the specification as follows, a processor of Intel® Core™ i5-8250U CPU @1.60GHz 1.80GHz and a ram size of minimum 4GB.

Facial Recognition Test Result

To find out the accuracy level of face recognition used in this system, a photo of a student whose face has been trained is taken. The size of the photo taken is 160 X 160. This is done to make the face recognition process accomplish faster, because the more significant the image size, the time needed to recognise the student's face becomes longer.

Running the face detection function, a figure is created the faces detected in one of the student's directory/folder. Below we can observe that each face was correctly detected and that we have a range of lighting, skin tones, and orientations in the detected faces.

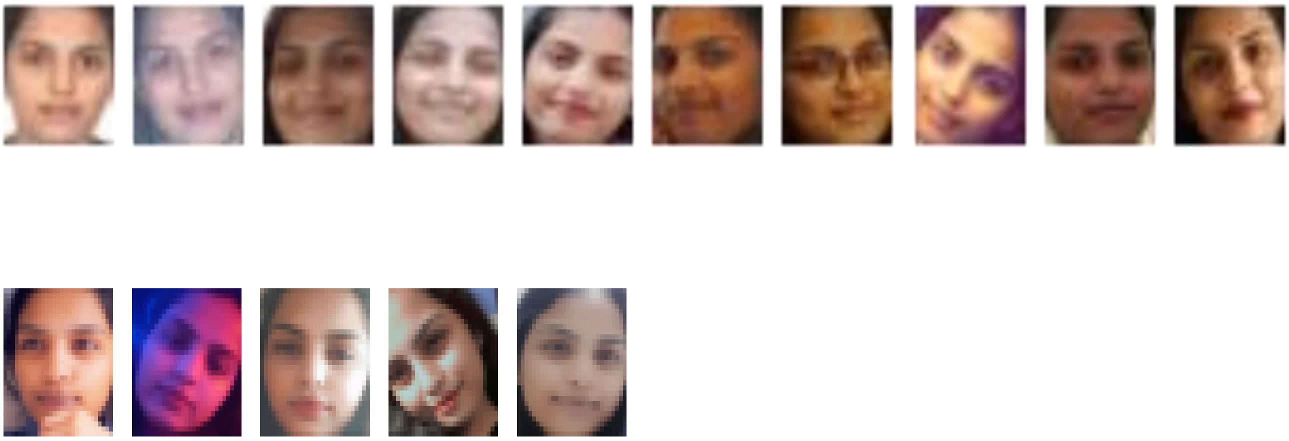
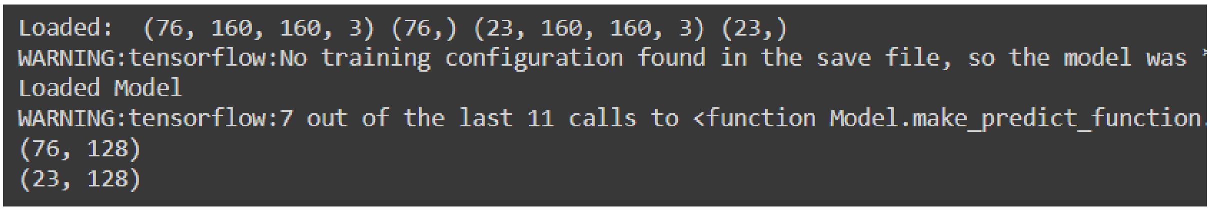


Figure 25 Plot of 15 photos of a student detected from the training dataset.

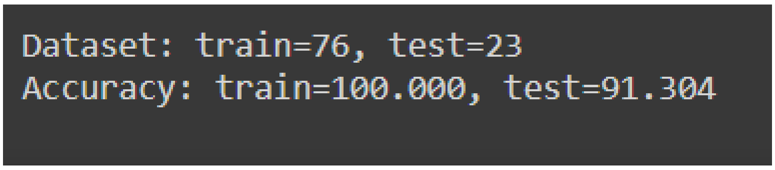
To get face embeddings, FaceNet Keras model is loaded correctly. The train dataset was then transformed into 76 face embeddings, each comprised of a 128-element vector. The 28 examples in the test dataset were also suitably converted to face embeddings.



```
Loaded:  (76, 160, 160, 3) (76,) (23, 160, 160, 3) (23,)
WARNING:tensorflow:No training configuration found in the save file, so the model was
Loaded Model
WARNING:tensorflow:7 out of the last 11 calls to <function Model.make_predict_function
(76, 128)
(23, 128)
```

Figure 26 Face embeddings

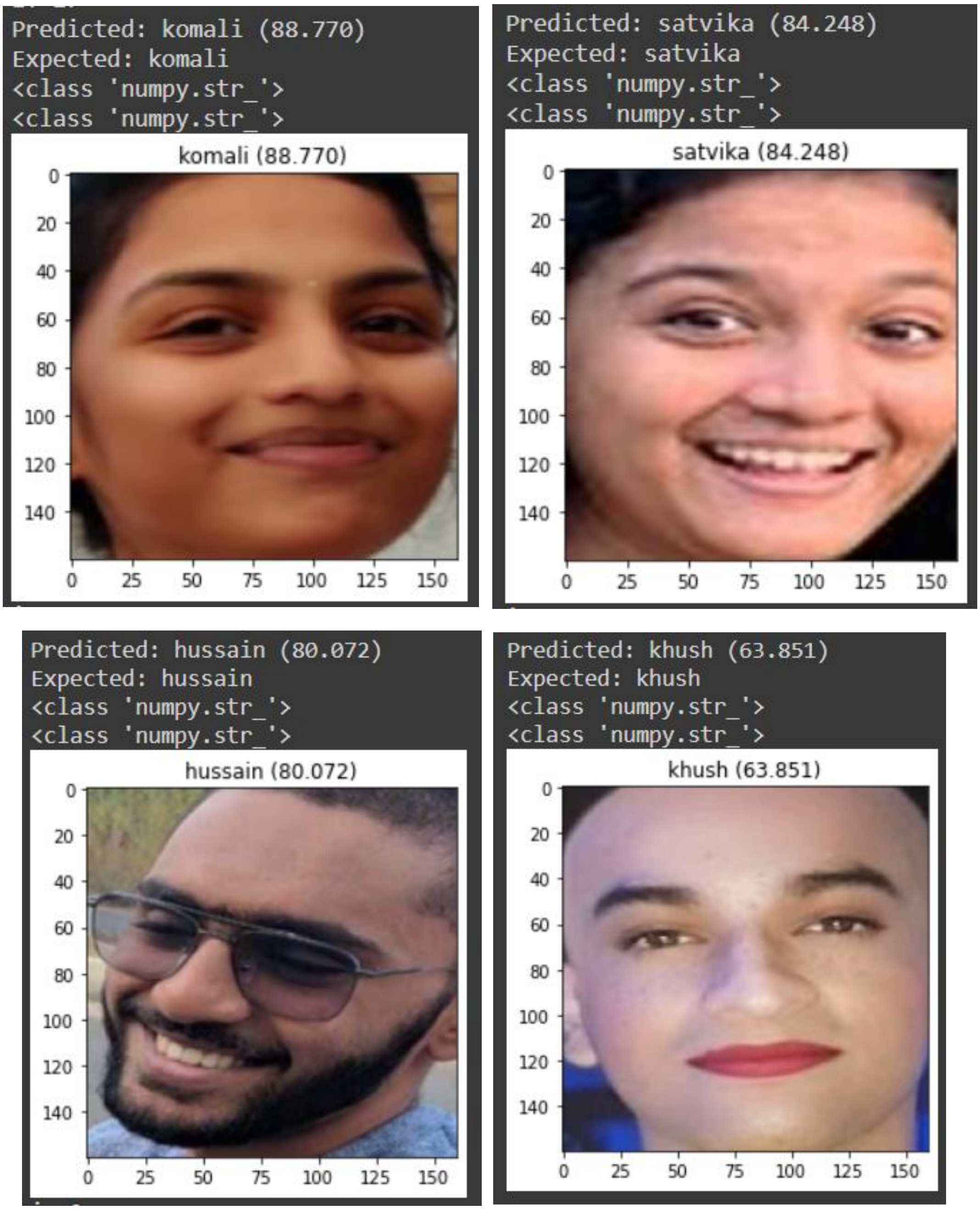
The model is then evaluated on the train and test dataset, showing perfect classification accuracy of 91.304 on test data set. This is not surprising given the size of the dataset and the power of the face detection and face recognition models used. The accuracy maybe affected by illumination of light leading to the results.



```
Dataset: train=76, test=23
Accuracy: train=100.000, test=91.304
```

Figure 27 Accuracy of the model

Below are the results of the predicted values, there are few cases where the prediction has been wrong. This might be because of a small training dataset and difference in the illumination of the light, and certain facial features.



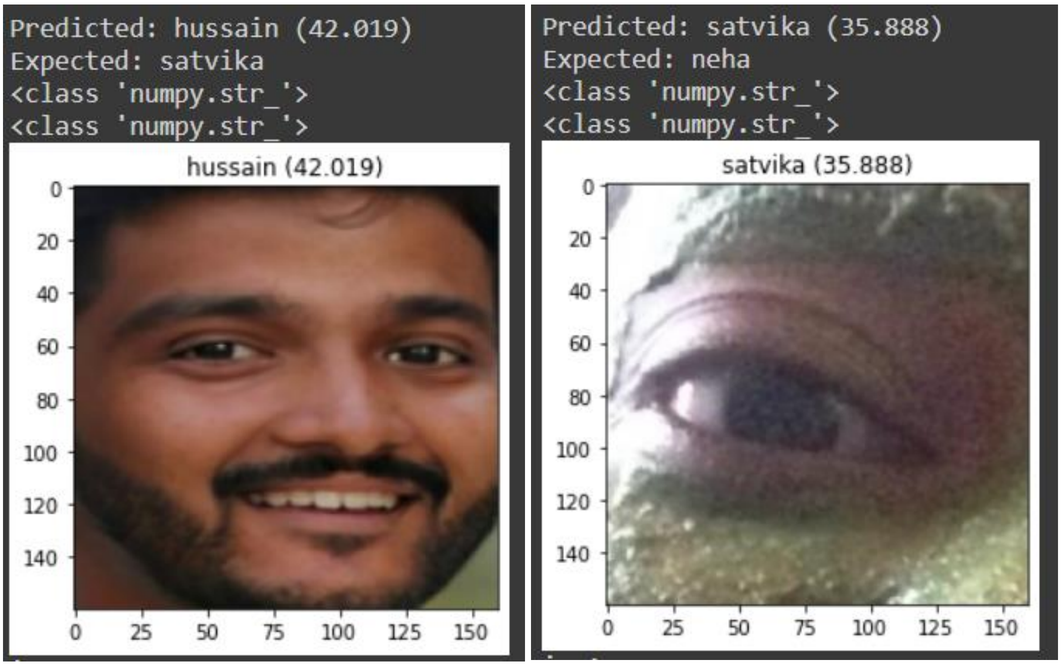
```
Predicted: komali (88.770)
Expected: komali
<class 'numpy.str_'>
<class 'numpy.str_'>
```

```
Predicted: satvika (84.248)
Expected: satvika
<class 'numpy.str_'>
<class 'numpy.str_'>
```

```
Predicted: hussain (80.072)
Expected: hussain
<class 'numpy.str_'>
<class 'numpy.str_'>
```

```
Predicted: khush (63.851)
Expected: khush
<class 'numpy.str_'>
<class 'numpy.str_'>
```

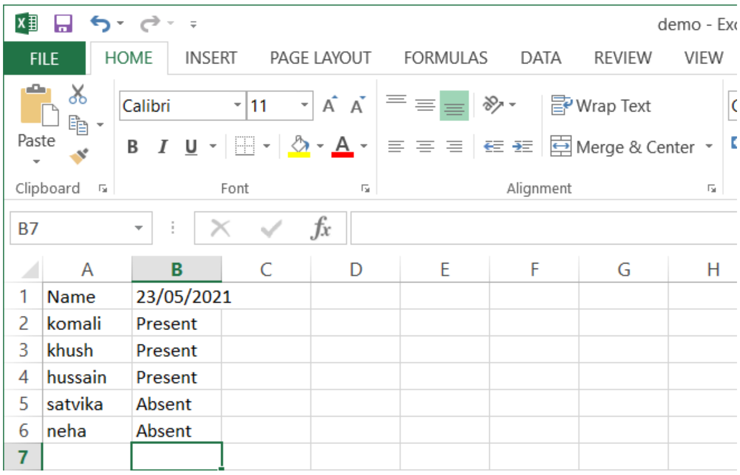
Figure 28 Detected and correctly identified by SVM



```
Predicted: hussain (42.019)
Expected: satvika
<class 'numpy.str_'>
<class 'numpy.str_'>
```

```
Predicted: satvika (35.888)
Expected: neha
<class 'numpy.str_'>
<class 'numpy.str_'>
```

Figure 29 Faces detected and wrongly identified.



	A	B	C	D	E	F	G	H
1	Name	23/05/2021						
2	komali	Present						
3	khush	Present						
4	hussain	Present						
5	satvika	Absent						
6	neha	Absent						
7								

Figure 30 Attendance Marked.

CONCLUSION

Before the development of this project, there were many loopholes in the process of taking attendance using the traditional methods which caused troubles to most of the institutions. Automated Attendance System using MTCNN and FaceNet is expected to be able to replace the old manual attendance process, which is currently used. The system has also proved to be time saving, secured and conquer the defects by merely saving resources but also reduce human intervention in the whole process. This standalone system detects the person which was already given in the dataset to track and an embedding being created was successfully detected with an accuracy of 91.304%. In real time scenarios, MTCNN and FaceNet algorithms outperforms other algorithms with better recognition rate and low false positive rate. SVM has also proved to be a better classifier when compared to others.

Future Work

For future work, the plan is to use cloud-based face recognition to speed up the face recognition process. The future work is to also improve the recognition rate of algorithms when there are unintentional changes in a person like having a mask or change in features due to surgery. The system should also be able to detect identical twins distinctly. Currently, the system to not curated to detect multiple faces from a single image which is yet to be achieved soon. The system should be able to take input from camera and be developed into a web application. At the same time, there are many places worth further research and discussion in terms of improving detection accuracy and recognition speed.

REFERENCES

[1] https://ieeexplore.ieee.org/document/6420657

[2] https://ieeexplore.ieee.org/document/7828654

[3]http://ijariie.com/AdminUploadPdf/STUDENT_ATTENDANCE_SYSTEM_USING_IRIS_DETECTION_ijariie4615.pdf

[4] https://ieeexplore.ieee.org/document/5356360?reload=true

[5] https://ieeexplore.ieee.org/document/7916753

[6] https://ieeexplore.ieee.org/document/9036819

[7]https://towardsdatascience.com/face-detection-with-haar-cascade-727f68dafd08#:~:text=So%20what%20is%20Haar%20Cascade,Simple%20Features%E2%80%9D%20published%20in%202001.

[8] https://www.pyimagesearch.com/2021/04/19/face-detection-with-dlib-hog-and-cnn/

[9] https://medium.com/@iselagradilla94/multi-task-cascaded-convolutional-networks-mtcnn-for-face-detection-and-facial-landmark-alignment-7c21e8007923#:~:text=Multi%2Dtask%20Cascaded,eyes%2C%20nose%2C%20and%20mouth.

[10]https://link.springer.com/chapter/10.1007/978-3-540-30126-4_90#:~:text=A%20hybrid%20technique%20based%20on,extracted%20by%20RSST%20color%20segmentation.

[11]https://medium.com/@cxu24/common-methods-for-feature-extraction-pca-and-lda-7b1f5679e3bf#:~:text=The%20most%20common%20linear%20methods,variation%20reflects%20differences%20in%20data.

[12]https://www.researchgate.net/publication/220745463_Cosine_Similarity_Metric_Learning_for_Face_Verification

[13] https://developers.google.com/ml-kit/vision/face-detection

[14]https://machinelearningmastery.com/how-to-perform-face-recognition-with-vggface2-convolutional-neural-network-in-keras/

[15] https://medium.com/analytics-vidhya/introduction-to-facenet-a-unified-embedding-for-face-recognition-and-clustering-dbdac8e6f02

[16] https://medium.com/analytics-vidhya/introduction-to-facenet-a-unified-embedding-for-face-recognition-and-clustering-dbdac8e6f02

https://arxiv.org/abs/1503.03832

APPENDIX

Operating Environment

The system will run on a 32-bit or 64-bit Windows and Mac X operating system. The other requirements such as a processor of Intel® Core™ i5-8250U CPU @1.60GHz 1.80GHz and a ram size of minimum 4GB.

Design and Implementation Constraints

The system must be programmed in an object-oriented language, in this case we will be using Python (3.6 or above). The database being used currently is SQLite and packages such as OpenCV, TensorFlow and Keras must be imported. Visual Studio Code is used for IDE. The files in which the information regarding securities and images should be secured against malicious deformations. Data should not become corrupted in case of system crash or power failure.

User Documentation

There should be minimal help needed to run this system. Should there be a need for help, there will be a help menu that will guide the user through using the system. This document will also serve as for all help documentation that follows.

Assumptions and Dependencies

One assumption that could affect the functioning of the system is if the packages mentioned in the Design and Implementation are not installed properly on the system. Another assumption that could affect the design is that the user will have adequate internet connection; this could affect the speed with which the interface communicates with the database. Lastly, this system will be written for users with a basic understanding of how computers work. Users with less computer experience may have a harder time.

Hardware Interfaces

The hardware requirement at the use end is simple with an i5 processor and a ram size of minimum 4GB and the application can also run on the hardware that can run on a basic simple browser, although the hardware should be good enough during the peak times for the servers.

Software Interfaces

The application should support all major web browsers that will make it convenient for the user to access our system with ease. The backend i.e., the database and face recognition models and services will be used to a great extent and hence it will be quiet efficiently designed.

Communication Interfaces

An ethernet port is used for network connection with computer through ethernet. The requirements associated with any communications functions required by this product, including email, web browser, network server communications protocols, electronic forms, and so on. Communication standards that will be used, such as FTP or HTTP. Communication security or encryption issues will handle by using java scripts.

Non-Functional Requirements

Performance Requirements

A RAM of size of minimum 4GB is a must for the smooth functioning of the system. The users must get response within seconds i.e., the response time of function should be minimum. Separate business logic at server side from the student and faculty interface ensures good performance. The system would exhibit high performance because it would be well optimised. The business logic would be clearly separate from the user interface.

Safety Requirements

The use of the student's photos shall not be used anywhere else and should be stored securely in the database and before taking the pictures of the students, their consent shall be taken.

Security Requirements

The professor shall be authenticated to get upload the attendance and download the attendance report.

Maintainability Requirements

Weekly backups of database is done. Access permission of the database is not given to people other than the admin so that database is not corrupted.

Software Quality Requirements

The system is scalable to hundreds of students and is also highly portable.