Software Requirements Specification

(Smart Attendance System with Face Recognition using OpenCV)

Group Members:

1. **SHUMBUL ARIFA**

   181CO152 (181636) | 9901917660 | shumbul.181co152@nitk.edu.in

2. **KEERTI CHAUDHARY**

   181CO226 (181303) | 8281851472 | keerti2001.kc@gmail.com

Table of Contents

1. Introduction

We are going to make an Artificial Intelligence-based face recognition system that can have a high impact on different organizations. The uniqueness or independence of an individual is his face. By considering this fact our system will be super faster and accurate in marking the attendance of individual students. We are going to use face detection and recognition in this project. Face detection is used to locate the position of face region and face recognition is used for marking the attendance. The database will store the faces of students. when the face of the student matches with one of the faces stored in the database then the attendance is recorded.

1.1 Purpose

The old method for taking attendance is by calling the name or roll number of the student to record attendance. It was a time-consuming and less accurate process of marking attendance because as we know the data written in the paper can be lost or can be less accurate because students often mark each other's attendance proxy. As a result, to solve these problems and avoid errors, we suggest computerizing this process by providing a system that records and manages students' attendance automatically without needing lecturers' interference.

1.2 Project Perspective

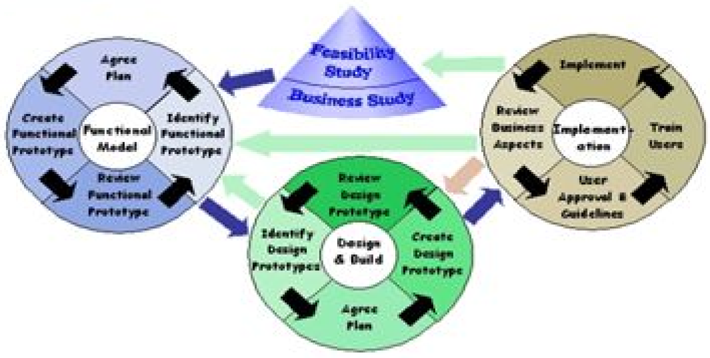
- We aim to improve and organize the process of tracking and managing the attendance of students and faculties in colleges.
- Reduce manual process errors by providing an automated and reliable attendance system that uses face recognition technology.
- Updating the time of attendance when a face is detected again.

1.3 Advantages

- The system stores the faces that are detected and automatically marks attendance.
- Convenient.
- Manipulate and recognize the faces in real-time using live video data.
- Multiple face detection.
- Multipurpose software.
- Can be used in different places, like classrooms, offices, etc.

2.Process Model

The agile project delivery framework is the approach that will be used for the development of the system in this project. **DSDM** is an agile methodology approach primarily used as a software development method. The development of this project is requiring user involvement to have timely visible results. Information gathered from the literature review shows researchers using different algorithms in face detection and recognition. However, as it is an ongoing research area, this project requires incremental implementation in smaller functionalities which will be put together at the end of a complete system. The project objectives specified in the proposal can only be achieved with the expertise of the user, as functionalities will be prioritized in order of importance alongside continuous user involvement. Unlike other approaches (Waterfall) where the stages of implantations are clearly defined, it is preferable to use the approach which will adapt easily to changes made during the implementation.



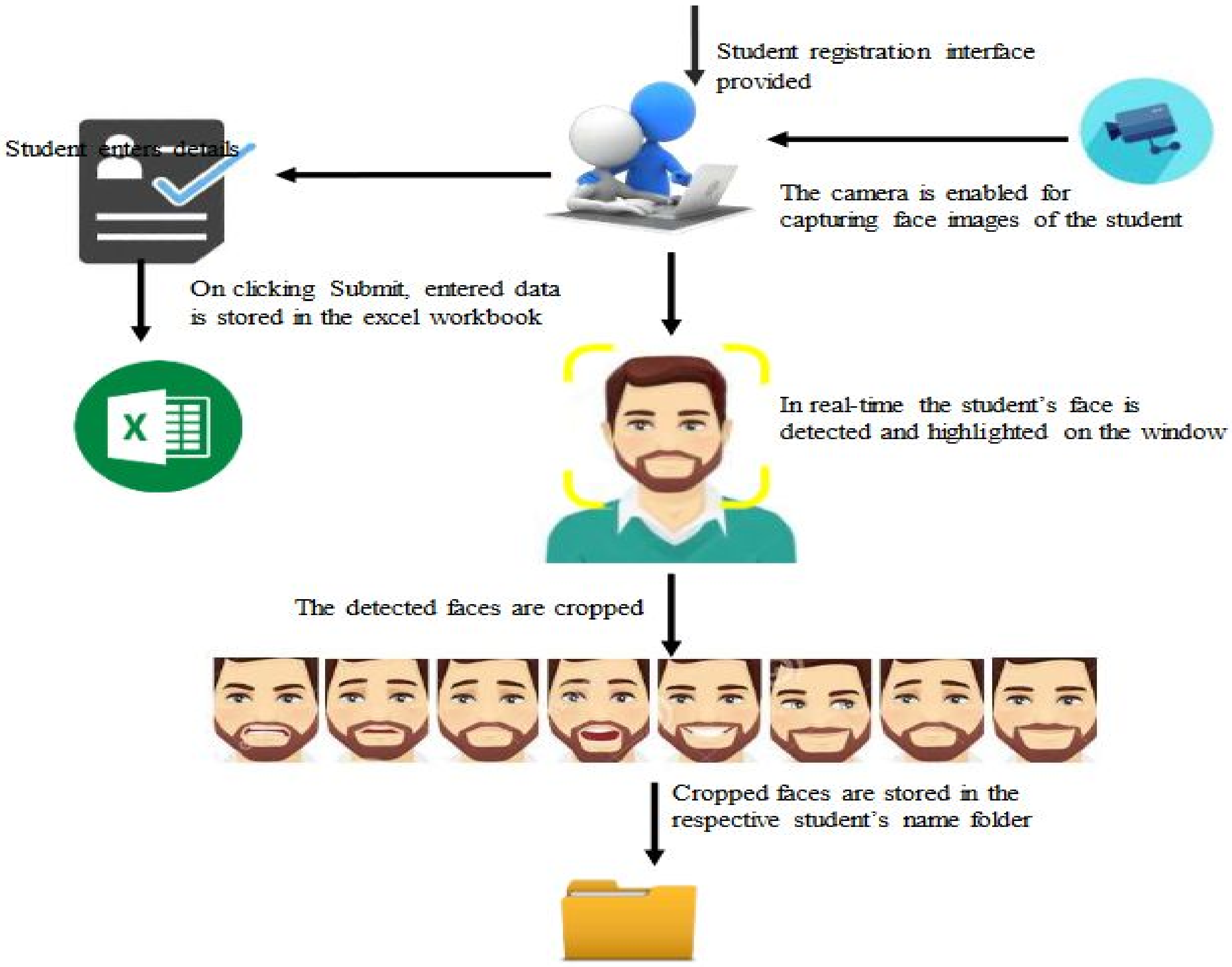
DSDM Development Process.

3. Requirements Elicitation

The requirements are the descriptions of the system services and constraints. In the requirement, elicitation process requirements are collected for software from customers, users, and stakeholders. In this paper, we discussed the concept of needs elicitation and process elicitation in software engineering. Needs elicitation are known through situational characteristics and sources of information. In situational characteristics, a service provider must know the types of stakeholders (domain expert, not domain expert, homogeneous, heterogeneous), the domain of the system being developed (existing system or new system), the scope of the system, know the environment, followed approach, know the problem, etc. In sources of information, a service provider knows competitors, technical literature, expert advice, surveys, etc. All this must prepare you for the elicitation process.

3.1 Functional Requirements

- Capture face images via webcam or external USB camera
- Faces on an image must be detected.
- The faces must be detected in bounding boxes.
- Resize the cropped faces to match faces the size required for recognition
- Store the cropped faces to a folder
- Load faces on database
- Train faces for recognition
- Perform recognition for faces stored on the database.
- Display the name of the output image above the image in the plot area


Student registration interface provided
Student enters details
The camera is enabled for capturing face images of the student
On clicking Submit, entered data is stored in the excel workbook
In real-time the student's face is detected and highlighted on the window
The detected faces are cropped
Cropped faces are stored in the respective student's name folder

3.2 Non-Functional Requirements

- **Accuracy and Precision:** the system should perform its process in accuracy and Precision to avoid problems.
- **Modifiability:** the system should be easy to modify, any wrong should be correct.
- **Security:** the system should be secure and save student's privacy.
- **Usability:** the system should be easy to deal with and simple to understand.
- **Maintainability:** the maintenance group should be able to fix any problem that occurs suddenly.
- **Speed and Responsiveness:** Execution of operations should be fast

3.3 Tools and Technology Requirements

- Python
- Pycharm Libraries
  - **Numpy** - is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.
  - **Dlib** - is a general-purpose cross-platform software library written in the programming language C++. Its design is heavily influenced by ideas from a design by contract and component-based software engineering.
  - **CMake** - is a cross-platform free and open-source software tool for managing the build process of software using a compiler-independent method.

  - **Face_Recognition** - Recognize and manipulate faces from Python or the command line with the world's simplest face recognition library.
  - **OpenCV** - is a library of programming functions mainly aimed at real-time computer vision.
- Visual Studio

4. Requirements Analysis

DSDM is an agile software development methodology. It is an iterative, incremental approach that is largely based on the Rapid Application Development methodology. In a DSDM project where time has been fixed, it is vital to understand the relative importance of the work to be done to make progress and keep to deadlines. Prioritization can be applied to requirements/User Stories, tasks, products, use cases, acceptance criteria, and tests, although it is most commonly applied to requirements/ User Stories. (User Stories are a very effective way of defining requirements in an Agile style; see later chapter on Requirements and User Stories for more information.)

MoSCoW is a prioritization technique for helping to understand and manage priorities. The letters stand for:

- **Must-Have** - are the requirements that have been identified by the client that must be implemented for the final solution
- **Should Have** - features will be implemented if possible with this project time frame. Although these features are a priority, the system will still meet its aim and objective.
- **Could Have** - is a desirable feature for the system of this project but will only be implemented if time permits.
- **Won't Have** - is a feature that was identified during the meeting that will be implemented in the future as it is not much of an issue at the moment.

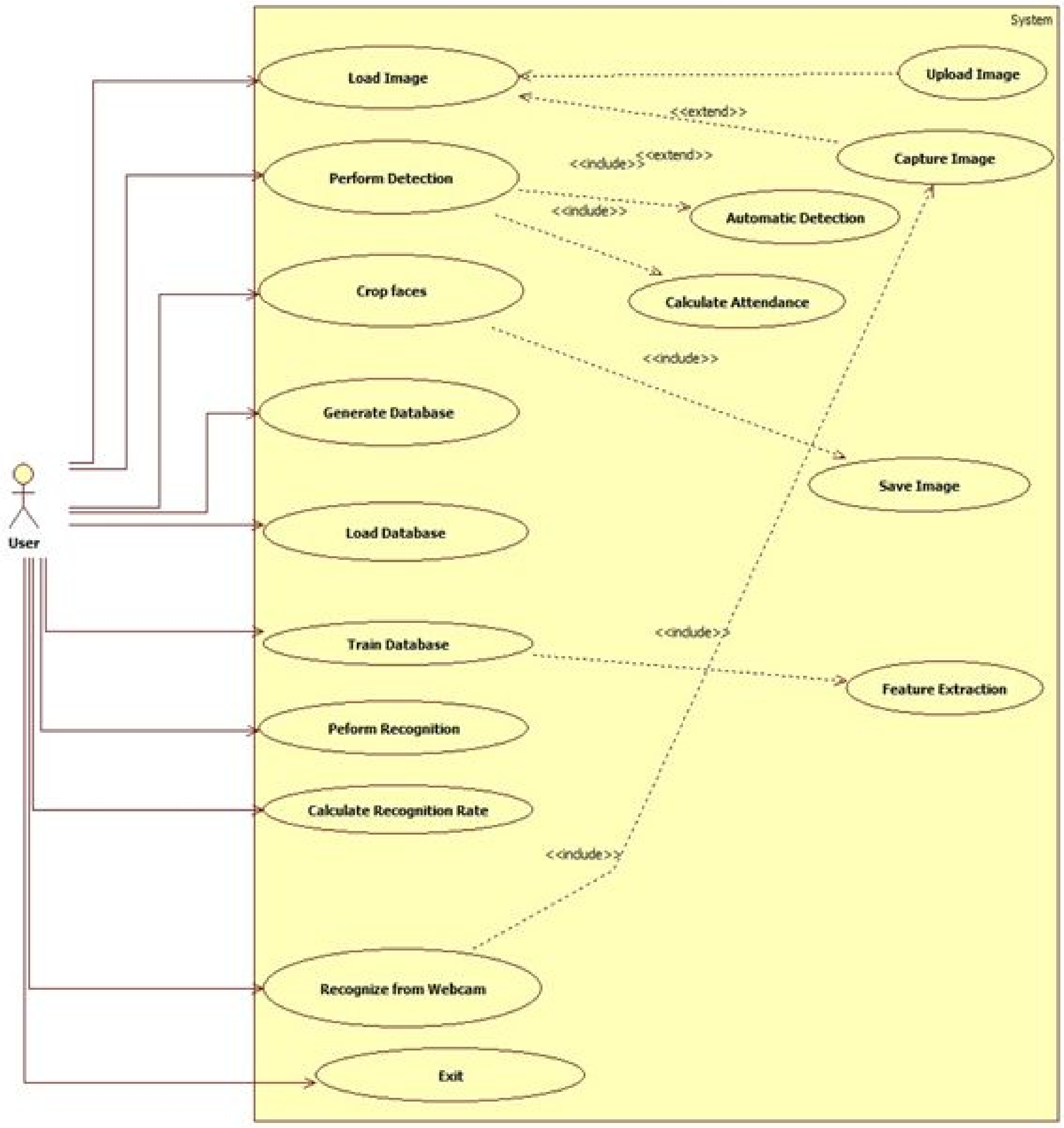
With regards to this project, the features in each category are:-

1. Must-Have :
   - The application must detect images by the use of bounding boxes.
   - Crop the total number of faces detected.
   - The application must resize faces to match the size of faces stored on the database.
   - Train Images for recognition.
   - Display the name of the output image above the image in the plot area.
   - Enter the details of the student in a database corresponding to each image.
2. Should Have :
   - Display the name of the input search image and the output image in the command window.
   - Determine the percentage Recognition of an image to that found on the database.
3. Could Have :
   - A robust app to perform face recognition for attendance.
   - Graphical user interface.
4. Will Not Have:
   - Login Authentication
   - Location tracking.

5. Requirements Specification

5.1 User Case Diagram

A UML use case diagram is the primary form of system/software requirements for a new software program underdeveloped. Use cases specify the expected behavior (what), and not the exact method of making it happen (how). A key concept of use case modeling is that it helps us design a system from the end user's perspective. It is an effective technique for communicating system behavior in the user's terms by specifying all externally visible system behavior. It only summarizes **some of the relationships** between use cases, actors, and systems. It does **not show the order** in which steps are performed to achieve the goals of each use case.



5.2 The User Story:

As a user, the client wants a system where they can load an image that will automatically detect the number of faces on the image. The system should have the option to capture an image using an inbuilt webcam on a laptop. As a user, the system should be able to crop the faces on an image after detection and store them on a folder/dataset that will be used for recognition purposes in the second phase of the system. The system should be able to automatically count the number of faces detected on the image.

As a user, the client requests the second phase of the system to be able to match faces stored on a dataset against input images which are either detected from the first phase or captured by an input device (camera). When the face is detected while taking attendance, it is recognized and details of name and time of attendance taken are entered in the database.

5.3 Specific Requirements

Face Detection and Recognition

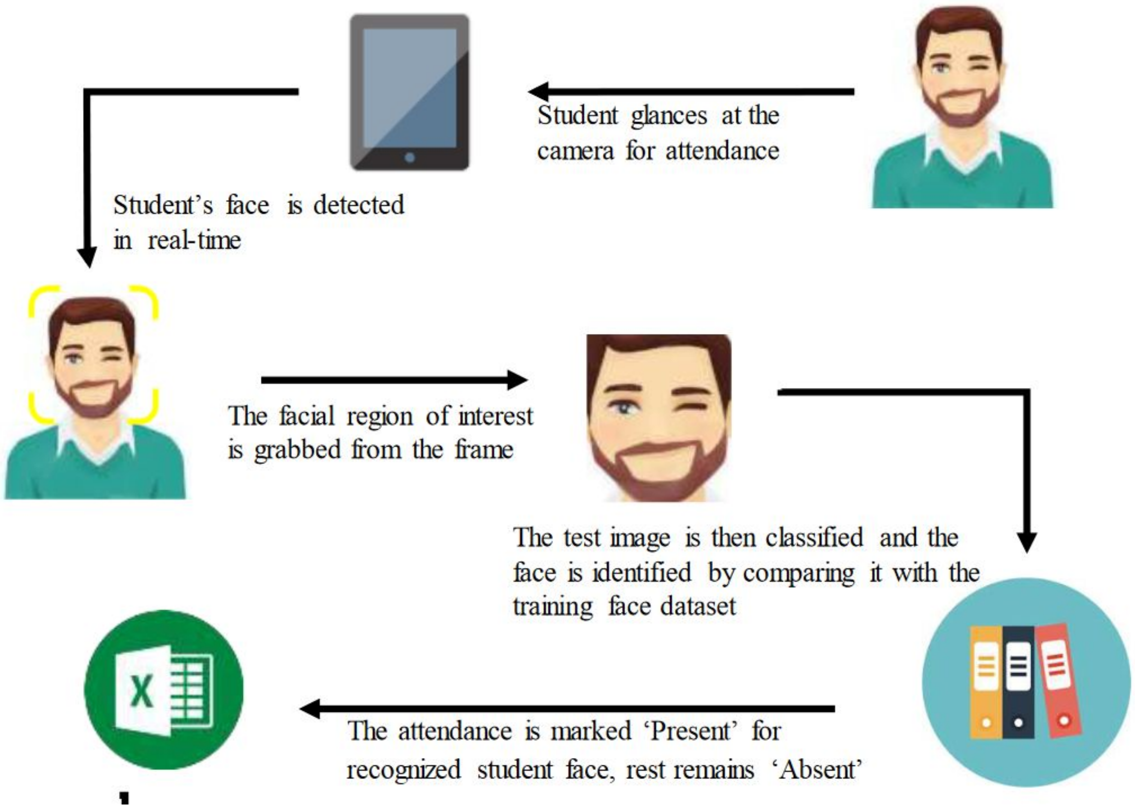
- This module will have a GUI interface used to train the system by taking pictures of each student and assigning names.
- It will show an image box and a button to start the camera
- The main menu has a present button to mark the attendance of the detected face in the camera.
- The main menu will also have delete, update, and close buttons to alter the information.

Update Attendance

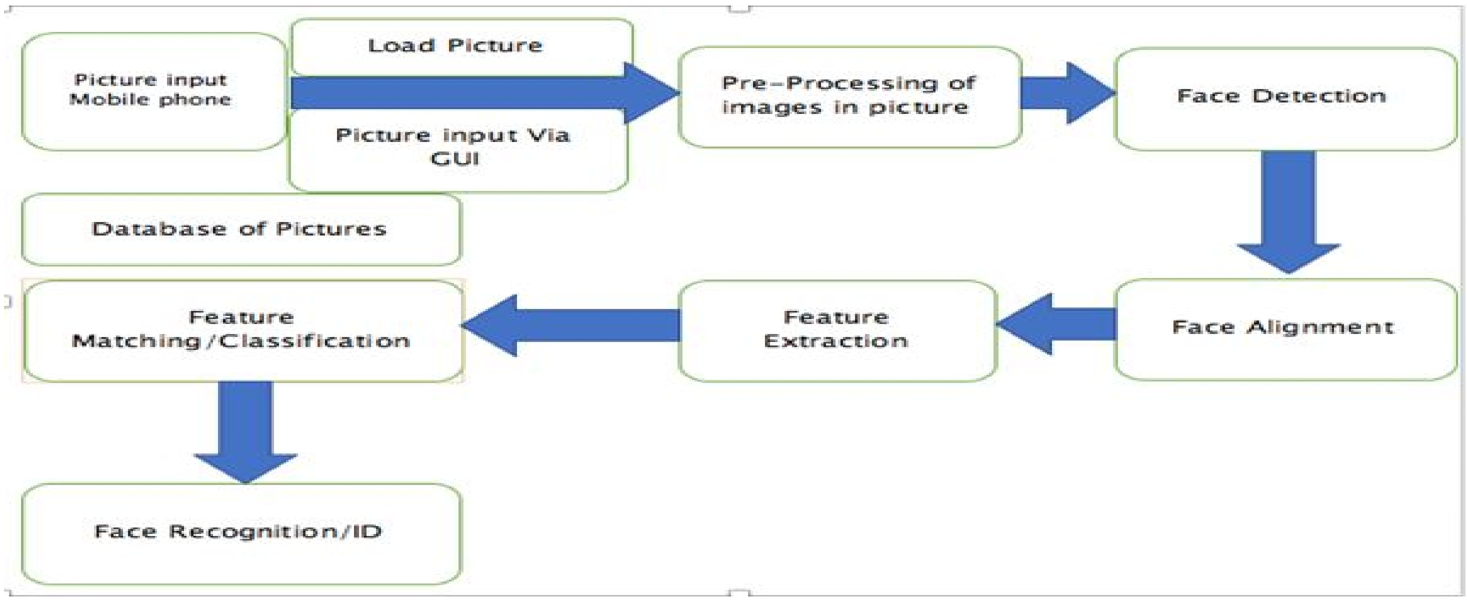
- When a student's image is re-entered while taking attendance the time of attendance is updated.
- It will also show a picture box and an update button.

Display Records

- This module will be used to display the information of students including their attendance status.
- It will also have a search bar to search for students on the list.


Student glances at the camera for attendance
Student's face is detected in real-time
The facial region of interest is grabbed from the frame
The test image is then classified and the face is identified by comparing it with the training face dataset
The attendance is marked 'Present' for recognized student face, rest remains 'Absent'

5.4 Face Detection and Recognition Flow Diagram



6. Requirements Validation

6.1 Testing

Test planning, test set creation, and testing are very important to validate the developed functionality. Test planning was independent of coding and was done in parallel to the coding stage. Equally important tests reporting, effective defect reporting, defect tracking, and defect resolution. The use of automated tools as well as well-established processes for these ensured that bugs were caught at the earliest possible stage and resolved cost-effectively. Unit testing, integration testing, functionality testing, system testing, and performance testing are some of the levels of testing which was performed at its time. Each level required its expertise, planning, and execution. The testing plan includes planning for several functions like:

1) Login feature
2) Registration of students
3) Face detection
4) Capturing 100 images automatically
5) Face recognition from the registered database
6) Unknown face detection
7) Attendance entry in Excel sheet
8) Reports generation
9) Email notification

6.2 Prototyping:

This System is capable of detecting the faces from the captured image from HD Video to analyze and detect the face. Face detection determines wherein an image, a face is located and it is being done by scanning the different image scales and extracting the exact patterns to detect the face.

The Prototype is built with a function from OpenCV that creates a search window that slides through an image and checks whether a certain region of an image looks like a face or not. These features and a large set of very weak classifiers use a single feature to define a certain image as face or non-face. Each feature is described by the template and its coordinate relative to the search window which is the origin of the size of the feature. The search window quickly scans the first classifier on the cascade

6.3 Verification test plan of the attendance taking process:

No.	Testcase Name	Description	Test Type	Test Values	Test Output	Status (pass/ fail)
1	Test input Timetable ID	This test is performed to check whether the Timetable entered is valid or not	Input value has more than 7 digits.	‘12345678’	Invalid entered input! Timetable ID must be 7 digits long.	Pass
2			Input value has less than 7 digits.	‘123456’	Invalid entered input! Timetable ID must be 7 digits long.	Pass
3	Test input Timetable ID	This test is performed to check whether the Timetable entered is valid or not	Input value doesn’t exist in the database.	‘9999999’	Timetable ID 9999999 doesn’t exist in the database.	Pass
4			Input values that have 7 numerical digits which existed in the database.	‘1000001’	No error page.	pass
5	Test input data	This test is performed to check whether the user selects a date before starting the attendance taking process.	Doesn’t select any date.	-	Error! Please select a date before submitting it.	pass
6			Selected a date	‘9/4/2018’	No error page.	pass

6.4 Test cases

ID	TC01
TITLE	Student registration form
PREREQUISITE	Roll no Name Semester Email
TEST ACTION	1. Start assistance application 2. Student registration
EXPECTED RESULT	The registration form should be filled up according to the validation set up and any incorrect detail should be popped as a warning message to the user.

ID	TC02
TITLE	Storing details of the Registration form
PREREQUISITE	Registered from filled up correctly
TEST ACTION	1. Start assistance application 2. Student registration
EXPECTED RESULT	Should check, whether the excel sheet, the name of a sheet according to the semester, and the subject are present or not and store information into that specific sheet or create a new one of that name and store it respectively.

ID	TC03
TITLE	Registration of faces
PREREQUISITE	Front camera
TEST ACTION	1. Start assistance application 2. Student registration form 3. Student face registration
EXPECTED RESULT	After connecting the front camera to the system, the face should be detected and automatically capture 100 images of a particular student and close automatically with a message of Registration over.

ID	TC04
TITLE	Encoding registered faces
PREREQUISITE	Training set
TEST ACTION	1. Start assistance application 2. Encode Database
EXPECTED RESULT	Get the folder names in training-dir as labels To be encoded in numerical form for Machine learning and if a file with the same name already exists, back up the old file.

ID	TC05
TITLE	Live face recognition
PREREQUISITE	Encoding and Training should be done before a live prediction
TEST ACTION	1. Start assistance application 2. Take Attendance
EXPECTED RESULT	The camera should be connected and turned on automatically and the name should be printed on the rectangular bounding box.

ID	TC06
TITLE	Attendance entry in Excel sheet
PREREQUISITE	Live prediction should be on
TEST ACTION	1. Start assistance application 2. Take Attendance
EXPECTED RESULT	Check whether today’s date is available or not, if not enter today’s date in a column next to yesterday’s date and match name which are already printed at the time of registration with the matched faces labels and mark present besides the name and under today’s date.

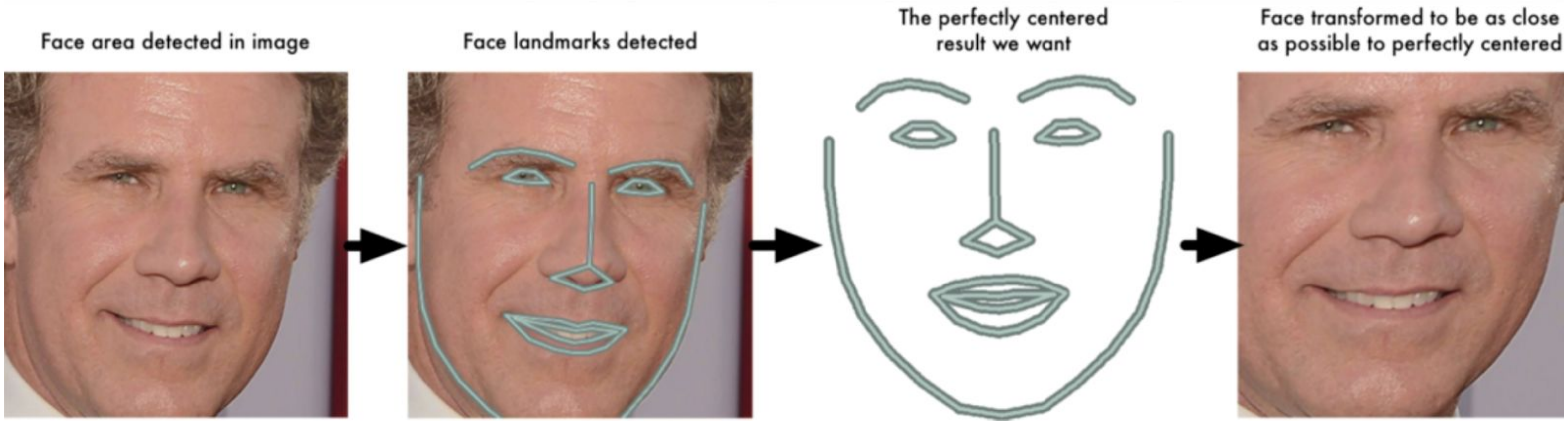
7. Requirements Management

Sometimes, a face detection algorithm may get more than one result even if there is only one face in the frame. In this case, post image processing is used for extracting the exact face coordinates with OpenCV. If the system output provides more than one rectangle, which indicates the position of the face, the distance of center points of these rectangles has been calculated. If this distance is smaller than a pre-set threshold, the average of these rectangles will be computed and set as the final position of the detected face. In simple words, the steps involved in face detection for each of the faces found in an image are as follows:

1. Find face in an image
2. Analyze facial features
3. Compare against known faces
4. Make a prediction

This would be the final image of a person from a photo as recognized by the software for face detection after recognizing a known person. The requirements analysis of any project has laid the foundation to take the project forward through to the design and implementation phases. Meeting with the client has been very useful in gathering functional and non-functional requirements. MoSCoW and Use Cases have been a strong tool to identify how the client wants the system to work. However, because aspects in this project are part of ongoing research, there will be changes during the implementation to achieve more which could lead to more contribution in the future.

The algorithm would be analyzed with images of different sizes and these images would be showing students in a classroom setting with natural sitting positions showing faces of different sizes. The basic idea is, no matter how the face is turned, we should be able to center the eyes and mouth in roughly the same position in the image, thus recognizing the person and marking their attendance.


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

What we need is a way to extract a few basic measurements from each face. Then we could measure our unknown face the same way and find the known face with the closest measurements. For example, we might measure the size of each ear, the spacing between the eyes, the length of the nose, etc. To be more specific, given below is how the attendance would be taken when multiple faces are detected by our smart attendance system, by saving the faces of different people in the database thus marking their attendance, for one person at a time.

1387388626000-GTY-45690...
42202_pro.png
501273088.png
chad-smith.png
chad-smith_52214color.png
chad.png
26-jimmy-fallon.w529.h529.png
27-jimmy-fallon-finger.w12...
64627-14323921696-jimmy-...
article-0-19A44A4B000005...
CDM_612991.png
maxresdefault.png
chad_smith_2010_05_06.png
chadsmith45.png
drummerch_david_630472...
fallon101115_2_250.png
jimmy-fallon-02-600x800.png
jimmy-fallon-1024.png

An integral image with low-quality scaling of this algorithm will lead to a loss in features that can directly affect the performance of the system.

Window Size: The window size consists of the MinSize and MaxSize. This sets the size a face detection can be. For this system, to maximize the accuracy, I have decided to use MinSize only which sets the minimum size for a face to include every tiny face at the back of the class, as the faces are of different sizes. The MinSize [height width] is greater than or equal to [20 20] for this system. Other sizes have been tested during the implementation and iteration testing of the system before settling for this size.

ScaleFactor: The ScaleFactor determines the scale for the detection resolution between successive increments of the window size during scanning. This parameter will help to decrease the number of false positives. (decrease in false positive is as a result of the increase in scale factor).

MergeThreshold: This parameter will control the number of face detections and declare the final detection of a face area after the combination and rejection of multiple detections around the object. The accuracy of the system depends on the level of MergeThreshold. The higher the value of the MergeThreshold level, the lower the accuracy and vice versa.

8. References

[1] https://www.sciencebuddies.org/science-fair-projects/engineering-design-process/design-requirements-emples

[2] http://www.appliedss.co.uk/InceptionPhase.aspx

[3] https://www.reqview.com/doc/iso-iec-ieee-29148-srs-example

[3] https://krazytech.com/projects/sample-software-requirements-specificationsrs-report-airline-database

[4] https://www.geeksforgeeks.org/software-engineering-requirements-engineering-process/

[5] https://www.geeksforgeeks.org/analysis-modelling-in-software-engineering/