PROFESSIONAL TRAINING REPORT

at

Sathyabama Institute of Science and Technology

(Deemed to be University)

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Engineering degree in Computer Science and Engineering

By

ROSHAN K R
(Reg. No – 42111086)

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

SCHOOL OF COMPUTING

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

CATEGORY- 1 UNIVERSITY BY UGC

Accredited "A++" by NAAC I Approved by AICTE

JEPPIAAR NAGAR, RAJIV GANDHI SALAI CHENNAI - 600119

OCTOBER - 2024

SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Category - I University by UGC
Accredited "A++" by NAAC | Approved by AICTE
www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Professional Training-1 Report is the bonafide work of **ROSHAN K R (42111086)** who carried out the Project entitled **"SMART ATTENDANCE SYSTEM"** under my supervision from June 2024 to October 2024.

Internal Guide

Dr.B.Ankayarkanni, M.E., Ph.D.,

Head of the Department

Dr. L. LAKSHMANAN, M.E., Ph.D.,

Submitted for Interdisciplinary Viva Voce Examination held on _____

Internal Examiner **External Examiner**

DECLARATION

I, **ROSHAN K R (Reg. No-42111086),** hereby declare that the Professional Training-1 Report entitled **"SMART ATTENDANCE SYSTEM"** done by me under the guidance of **Dr.B.Ankayarkanni, M.E., Ph.D., Professor, CSE,** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in **Computer Science and Engineering**.

DATE:

PLACE: Chennai **SIGNATURE OF THE CANDIDATE**

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **BOARD OF MANAGEMENT** of **Sathyabama Institute of Science and Technology** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala, M.E., Ph. D.**, **Dean**, School of Computing, and **Dr. L. Lakshmanan, M.E., Ph.D., Head of the Department** of Computer Science and Engineering for providing me with necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr.B.Ankayarkanni, M.E., Ph.D.,** for her valuable guidance, suggestions, and constant encouragement paved the way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

TRAINING CERTIFICATE

Elite

NPTEL ONLINE CERTIFICATION

(Funded by the MoE, Govt. of India)

Skill India
कौशल भारत-कुशल भारत

This certificate is awarded to

ROSHAN K R

for successfully completing the course

Data Science for Engineers

with a consolidated score of **70** %

Online Assignments	19.17/25	Proctored Exam	51.28/75

Total number of candidates certified in this course: **3812**

Prof. Andrew Thangaraj
Chair
Centre for Outreach and Digital Education, IITM

Jul-Sep 2024
(8 week course)

Prof. Vignesh Muthuvijayan
NPTEL Coordinator
IIT Madras

Indian Institute of Technology Madras

FREE ONLINE EDUCATION swayam शिक्षित भारत, उन्नत भारत

Roll No: NPTEL24CS133S438300438 To verify the certificate No. of credits recommended: 2 or 3

ABSTRACT

The Smart Attendance System project utilizes the Haar Cascade Frontal Face Algorithm for face detection, a highly efficient machine learning-based approach used to detect human faces in real-time. This system works by capturing images of individuals' faces and creating a dataset that trains the model to recognize each person’s unique facial features. Once the dataset is created, it is stored for future reference and used for tracking attendance. When a live video feed is provided to the system, it scans the faces of individuals and cross-checks them against the dataset to identify the person. Upon successful identification, the system logs the name, along with the exact time and date, and records this data into an Excel file.

The process is fully automated, thereby eliminating the need for traditional, manual attendance methods such as sign-in sheets or roll calls. The system streamlines the entire attendance process, saving time and reducing errors. Additionally, using Python’s Pandas library, the stored attendance data in the Excel file can be easily loaded and analyzed. For instance, loading the data can be done using the command df = pd.read_excel('attendance.xlsx').

Further analysis such as checking the total number of entries per day or the number of absences can be done using df.groupby('Name').count() or filtering specific dates with df[df['Date'] == '2024-10-11']. This system not only ensures high accuracy in attendance tracking but also provides the ability to analyze attendance patterns effectively through data analytics..

TABLE OF CONTENT

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW AND ISSUES INVOLVED

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 factour 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. whenthe face of the student matches with one of the faces stored in the database then the attendance is recorded.

1.2 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 attendanceproxy. 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.

1.3 PROJECT PERSPECTIVE

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

1.4 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.

CHAPTER 2

ANALYSIS

2.1 PROCESS MODEL

The agile project delivery framework is the approach that will be used for the development of the system in thisproject. 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.

Fig .No – 2.1 DSDM Development Process

2.2 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 youfor the elicitation process.

2.2.1 FUNCTIONAL REQUIREMENTS

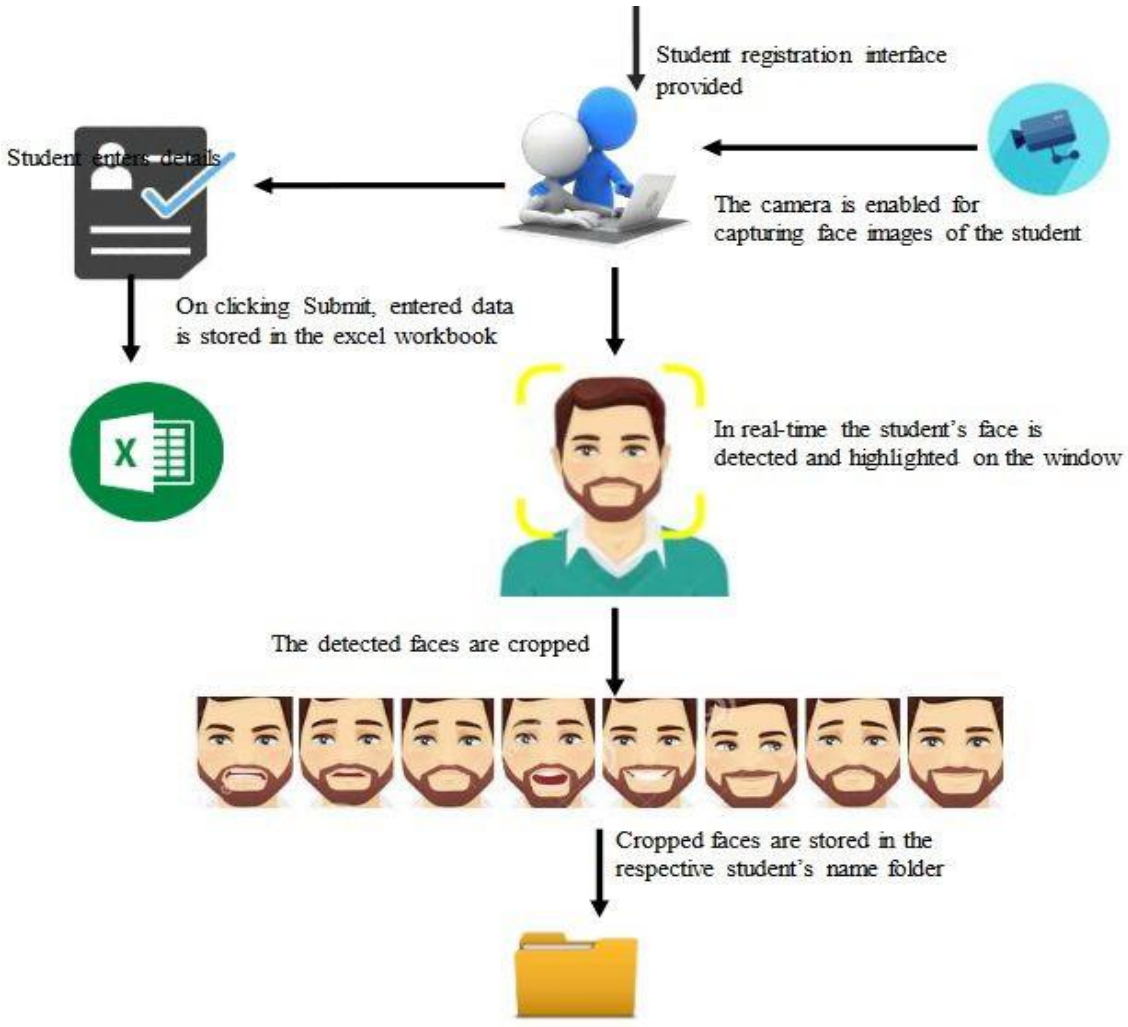

Fig 2.2 project lifecycle

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 for recognition

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

2.2.2 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

2.2.3 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.

2.3 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.

2.4 REQUIREMENTS SPECIFICATION

2.4.1 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.

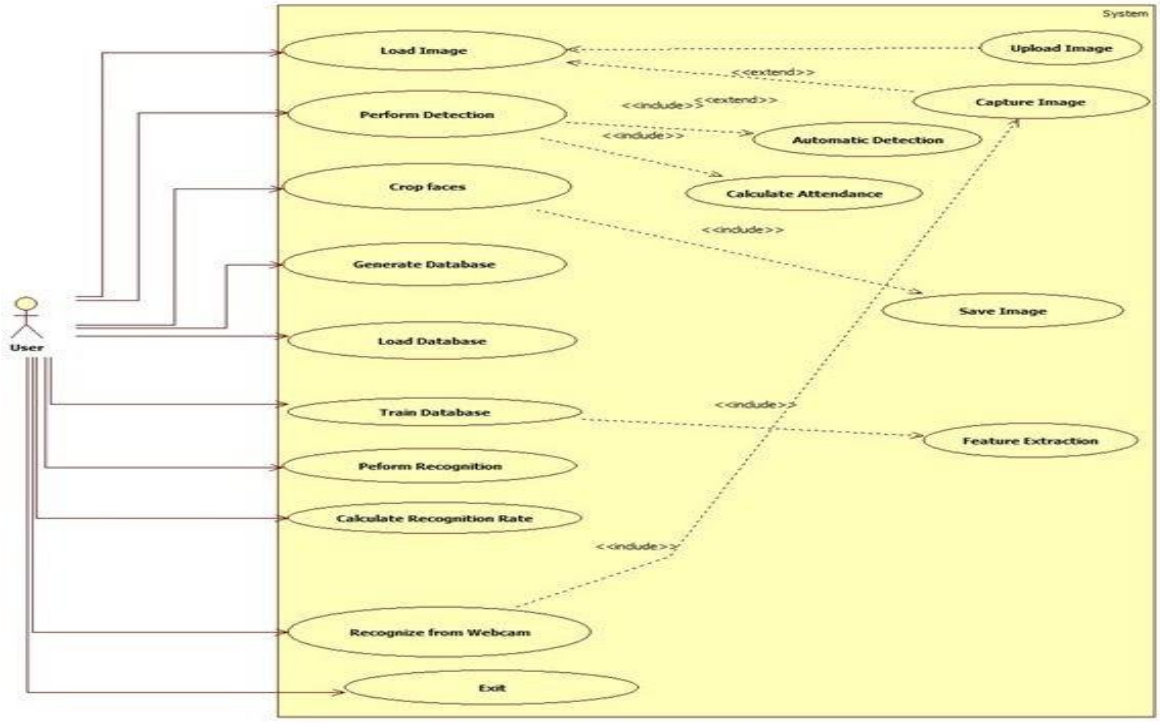

Fig 2.3 use case diagram

2.4.2 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.

2.4.3 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.

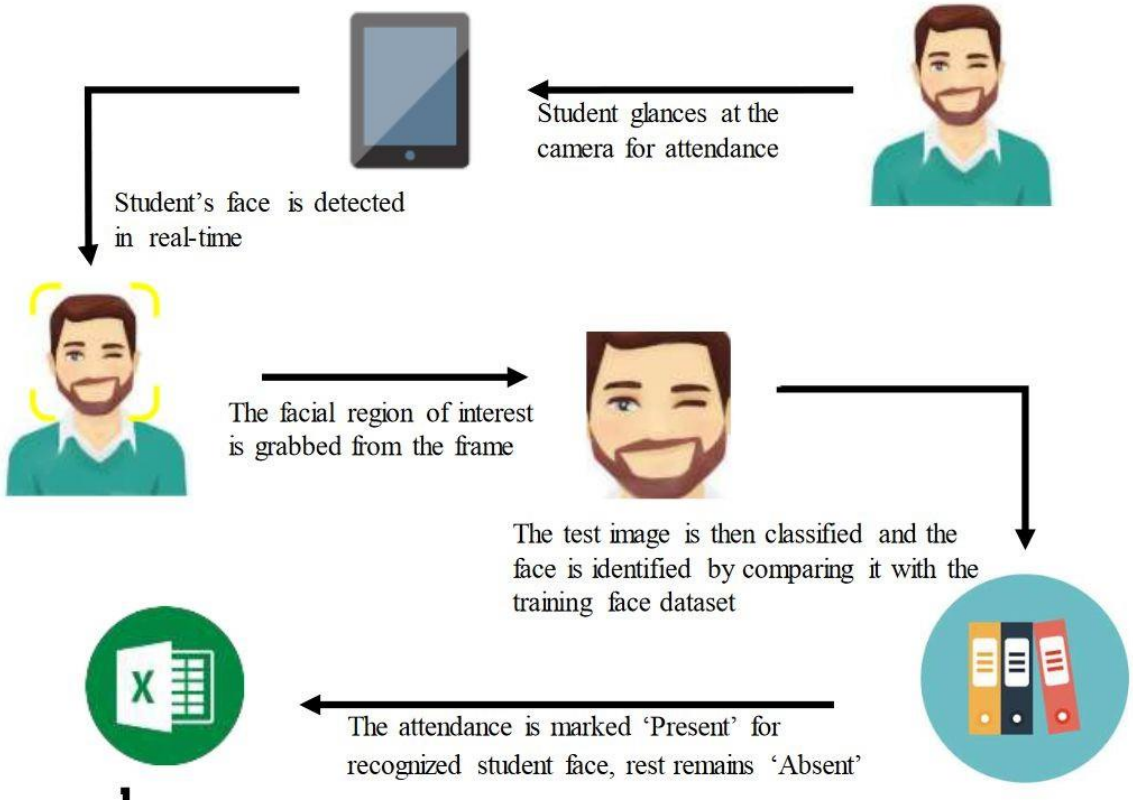

Fig 2.4 project flow-diagram

2.4.4 FACE DETECTION AND RECOGNITION FLOW DIAGRAM

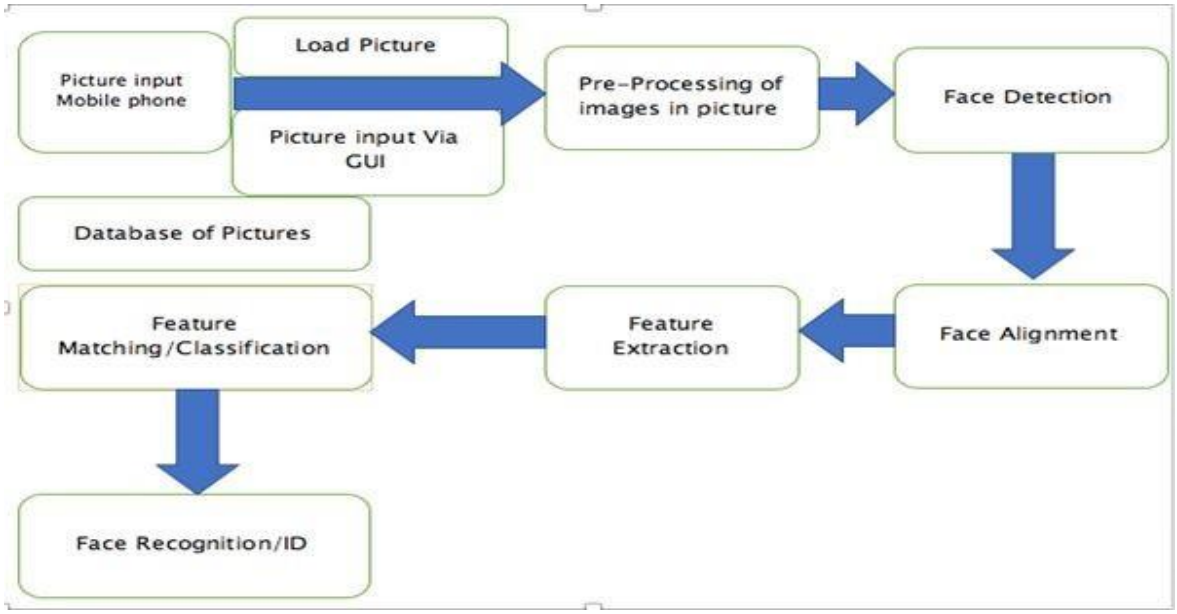

Fig 2.5 face detection flow diagram

CHAPTER 3
REQUIREMENT ENGINEERING

3.1 TESTING

Testing is the technique for assessing a framework or its parts to search out whether it fulfills the ideal requirements or not. In direct words, testing is executing a Frame work to recognize any holes, mistakes, or missing requirements. It is a fundamental advance in a product designing cycle to develop any product. Software Testing is fundamental since we as a whole commit errors. A portion of those errors are irrelevant, however, some of them are costly. We have to check everything and anything we produce since things can generally turn out wrong.

3.1.1 BLACK BOX TESTING

In black-box testing, the structure of the program isn't thought about. It considers the usefulness of the application as it was. It is likewise called practical testing. In this sort of testing, the analyzers focus on utilitarian testing, that is, on giving known information and checking if the realized yield is acquired. This strategy is by and large followed while completing acknowledgment testing when the end-client isn't a product engineer however just a client.

3.1.2 WHITE-BOX TESTING

White-box Testing is a kind of testing technique where the tester (a developer testing the application) knows about the framework internals. Its usage is straightforward. The goal is to guarantee that each line of the code is tried. The tester recognizes all consistent, plan, and typographical mistakes. The tester likewise needs to approve the interior structure of the thing viable alongside the yield.

3.2 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.

3.3 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.

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.

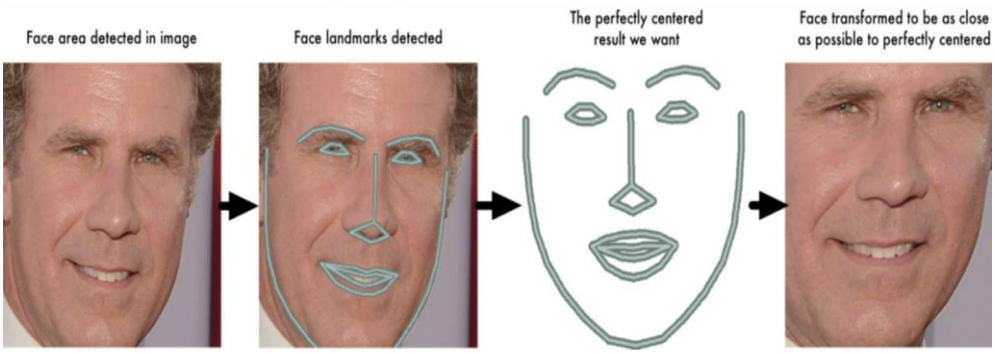

Fig:3.1 - Haar cascade algorithm working principle

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.

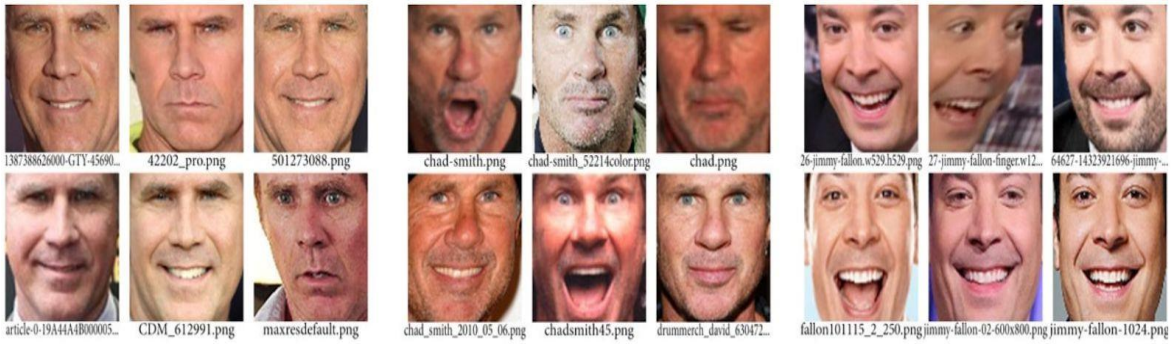

Fig:3.2 - **Face datasets for model training**

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.

CHAPTER 4

RESULTS AND DISCUSSION

4.1 FUTURE SCOPE

In future work, we need to improve face detection effectiveness with the help of the interaction among our system, the scholars, and the faculty. On the other hand, our system is improved by desegregation video-streaming service and lecture archiving system, to give additional profound applications in the field of distance education, course management system (CMS), and support for college development (FD). We will improve this technique thus as we tend to run this system with additional than two students on a bench and permitting them to vary their positions.

- Can improve security by adding administrator login.
- Can use Neural Network for high accuracy.
- Can be used in a big factory or employee attendance.
- Can build on a fully web-based system.

4.2 RESULTS AND OUTPUT

This system is capable of detecting the faces from the captured image from HD Video to investigate and discover the face. Face detection determines when a picture, a face is found and it's being done by scanning the various image scales and extracting the precise patterns to discover the face. The entire range of pictures saved within the folder can actuate the numeric attendance throughout the class. The images within the folder are going to be matched by the recognition part to faces already held on and trained on the database images. This may facilitate confirmation of attendance for a specific student. It'll conjointly modify the lecturer to require full management of the class by calling out every student by name once it's needed. The random sequence of input images when tested will most likely be the sequence from the output image. Although it does not change the overall percentage of recognition, it was not possible to tell the user at what percentage they could decide a face is a face, in other words, it hardly fails to recognize a face. The only way out was to carry out a test on all five input images and at least with three matches, the user can confirm a face, based on the output match displayed side by side. The performance of the system has impacted the reliability of the system. Because it is still an ongoing research area, the system will not be available for use at the end of the project. However, it can be used for research purposes by the supervisor and

experimented with in a lecture room before approval.

4.2.1 TEST CASE 1 – DATASET CREATION

INPUT

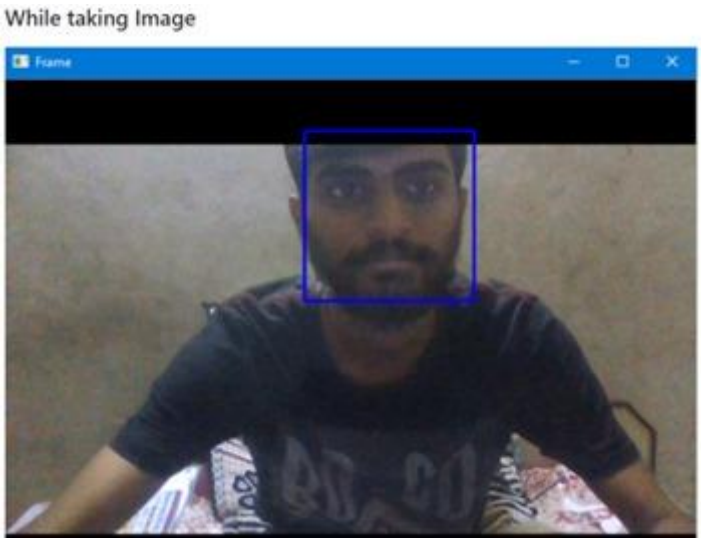

Fig 4.1 – Images for dataset

OUTPUT

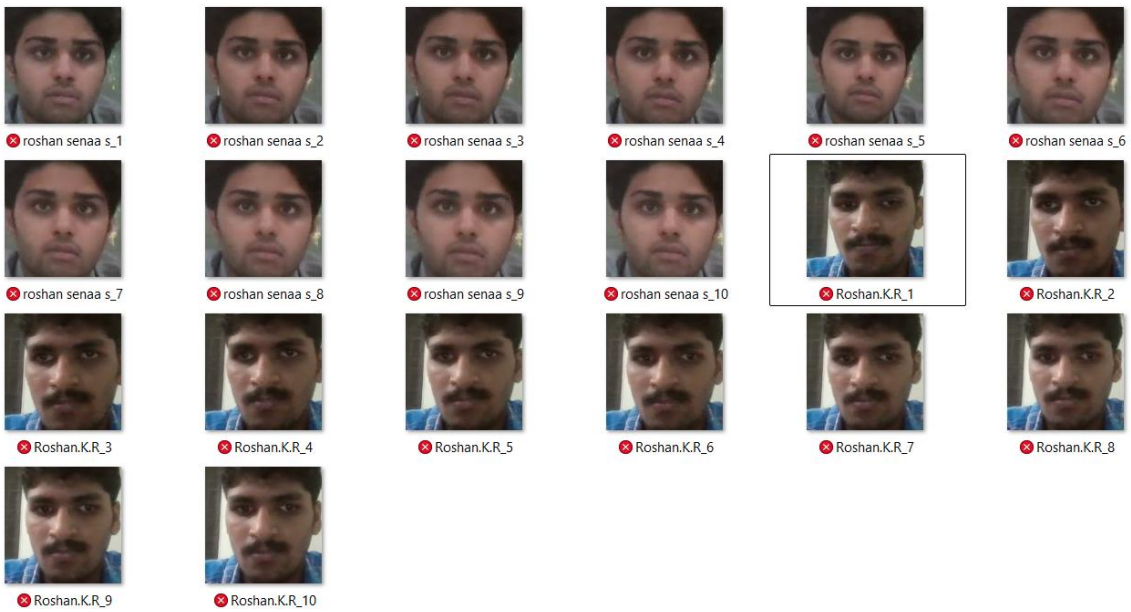

Fig 4.2 – Face dataset for model Training

4.2.2 TEST CASE 2 – FACE RECOGNITION AND ATTENDANCE LOG

INPUT

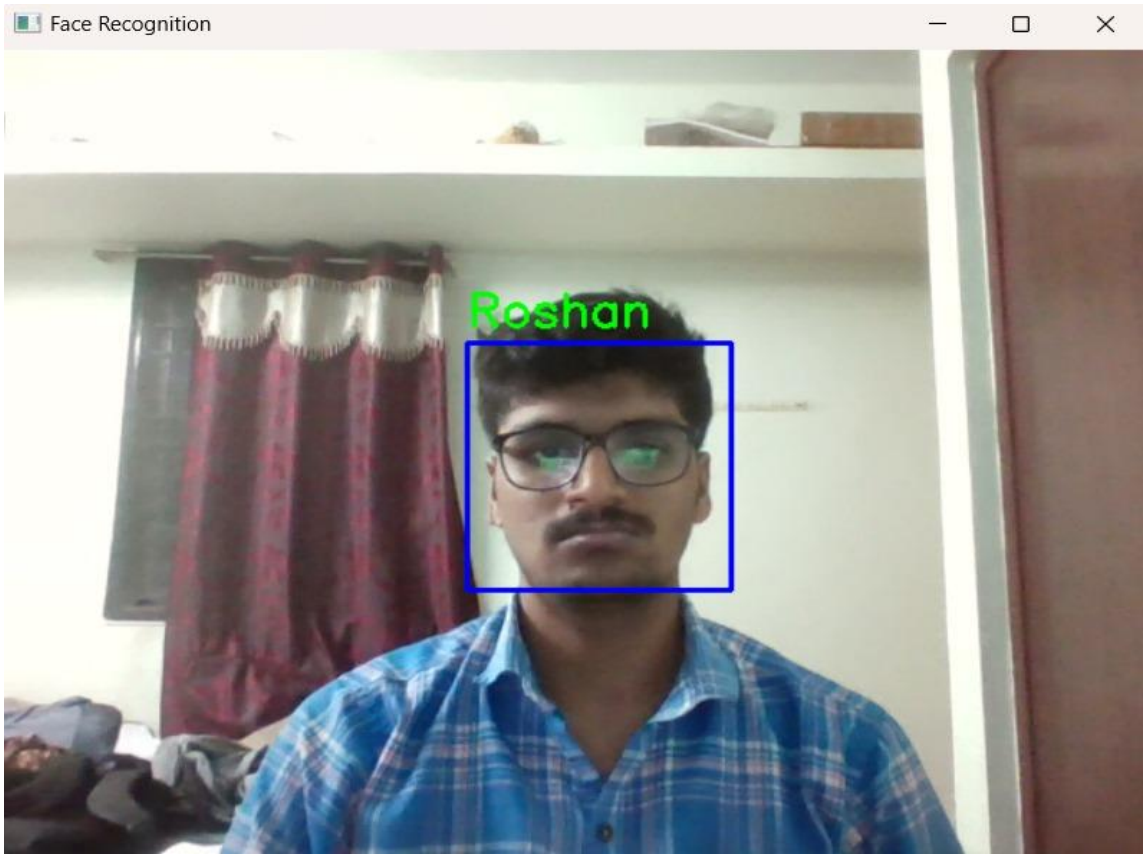

Fig 4.3 – Face Recognition

OUTPUT

	A	B	C	D
1	**Name**	**Date**	**Time**	
2	Roshan	2024-10-2	23:42:14	
3	steve jobs	2024-10-2	23:42:14	
4	bill gates	2024-10-2	23:44:19	
5	Roshan	2024-10-2	09:21:43	
6	steve jobs	2024-10-2	09:21:48	
7	roshan ser	2024-10-2	12:34:19	
8	rohith kum	2024-10-2	12:34:22	
9	bill gates	2024-10-2	12:34:22	
10	srilekha	2024-10-2	18:00:13	
11				
12				

Fig 4.4 – Attendance Logging

CHAPTER 5
CONCLUSION

5.1 CONCLUSION

The entire project has been developed from the requirements to a complete system alongside evaluation and testing. The system developed has achieved its aim and objectives. More careful analysis is required on a project intrinsically. The ways used may be combined with others to attain nice results. Completely different ways are enforced within the past in keeping with the literature review. The conclusion to set the parameters of this part of the system based on a very small class size was due to the failures obtained from the recognition part of the system. The size of the image is very important in face recognition as every pixel counts. The algorithm we've used 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. At the beginning of every year, the photographs of the latest people can be taken and kept by the organization. Each person will have the right to be told regarding the employment of their faces for a face recognition attendance system. This should be in line with government laws on moral problems and information protection laws and rights. The individuals can consent to their pictures used for the aim of attendance.

REFERENCES

1. Deshmukh, B.J., & Sudhir, M.K. (2014). Efficient attendance management: a face recognition approach. *International Journal of Advanced Engineering and Technology*, 7(3), 974.
2. Baskar, J., Karthik, S., & Jithin, S. (2020). Facial recognition for attendance and management. *International Journal of Recent Technology and Engineering*, 8(6), 1431-1435.
3. Husein, N.A., & Abbas, A.M. (2019). A novel approach to facial recognition in attendance systems. *Journal of Computer Science and Technology*, 34(4), 790-799.
4. Ramasamy, M., & Muthusamy, R. (2021). Smart attendance management system using face recognition and IoT. *International Journal of Computer Applications*, 175(18), 20-26.
5. Sahu, A.K., & Gupta, P. (2021). Implementation of smart attendance management system using face recognition. *Journal of King Saud University - Computer and Information Sciences*. DOI: 10.1016/j.jksuci.2021.03.009.
6. Bhatti, K.L., & Jilani, M.F. (2018). Smart attendance system using facial recognition and deep learning. *EAI Endorsed Transactions on Creative Technologies*, 5(17), e4.
7. Rani, A., & Chaurasia, V. (2021). Real-time face recognition for attendance system. *International Journal of Advanced Computer Science and Applications*, 12(5), 139-143.
8. Jain, P., & Kumar, A. (2020). Face detection and recognition based attendance management system. *International Journal of Recent Technology and Engineering*, 9(3), 25-29.
9. Singh, V., & Kumar, M. (2020). Automatic attendance management using face recognition. *International Journal of Engineering Research and Technology*, 9(4), 57-61.
10. Verma, P., & Gupta, S. (2021). Face recognition-based attendance system for classrooms. *International Journal of Computer Applications*, 177(26), 17-22