**AUTOMATED ATTENDANCE SYSTEM USING FACIAL RECOGNITION**

**By**

**Haitham Eliyas – 20178065**

**Qusai Al Saidi – 20155833**

**Nooh Al Balushi – 20201477**

**Submitted in partial fulfilment of the requirements of the degree of Bachelor of Science in Computer Science**

**Summer I 2021**



**Department of Mathematics and Computer Science**

**Modern College of Business and Science**

# Declaration

We certify that:

1. The work has not been submitted to any other College/Institute/University for any degree or diploma.
2. We have followed the guidelines provided by the college in writing project report.
3. We have conformed to the norms and guidelines given in the Ethical Code of Conduct of the college.
4. Whenever we have used materials (data, theoretical analysis, and text) from other sources, we have given due credit to them by citing them in the text of the project report and giving their details in the references.
5. Whenever we have quoted written materials from other sources, we have put them under quotation marks and given due credit to the sources by citing them and giving required details in the references.

**Name:** Haitham Eliyas **Date:**

**ID:** 20178065

**Signature:**

**Name:** Qusai Al Saidi

**ID:** 20155833

**Signature:**

**Name:** Nooh Al Balushi

**ID:** 20201477

**Signature:**

# Certificate

Certified that this report titled **Automated Attendance System Using Facial Recognition** isa bona fide record of the project work done by Mr............................... under my supervision and guidance, towards partial fulfilment of the requirements for the award of the Degree of BS Computer Science/ Information System of the Modern College of Business and Science”.

**Student Name:** Haitham Eliyas

**ID:** 20178065

**Signature:**

**Date:**

**Student Name:** Qusai Al Saidi

**ID:** 20155833

**Signature:**

**Date:**

**Student Name:** Nooh Al Balushi

**ID:** 20201477

**Signature:**

**Date:**

**Supervisor Name:**

**Position:**

**Signature:**

**Date:**

**HOD Name:**

**Position:**

**Signature:**

**Date:**

# Acknowledgements

We have put a lot of effort into this project, but it would not have been possible without many individuals who supported and guided us when we needed them to. We sincerely thank everyone of those people.

We would like to express utmost deepest gratitude to Dr Joseph Mani, Head of Computer Science Department, for his continuous guidance and supervision despite his hectic schedule. We are grateful to Dr Vijaya Padmanabha for her advice and critiques through various stages of this project.

We are thankful for their timely feedback, ideas and encouragement which helped us in keeping the project on track. We acknowledge deep sense of gratitude and encouragement we received from Dr Vijaya Padmanabha to pursue project of this kind.

Furthermore, we would like to thank our friends and families for constant encouragement and support.

Haitham Eliyas

# Abstract

These days student attendance is an area of concern for educational institutes due to student’s attendance in classes affecting their academic performance. There are two traditional methodologies of attendance taking, manual and automated.

Manual attendance such as calling student’s name takes a long time, whereas automated attendance systems such as RFID can be easily spoofed by students scanning the cards of their classmates, and fingerprint scanners are expensive and can require many attempts before they correctly identify the fingerprint. Hence, there is a need to implement an attendance system which is not only faster than existing ones but also prevents buddy punching practices.

We thought of a new method that addresses all these problems, using facial recognition algorithms and a database of registered students to automate attendance taking. In this project, we implemented AFRAS (Automated Facial Recognition Attendance System) to make the process of attendance taking easier, faster, and less disruptive to the classroom environment. Every individual has a unique facial characteristic and AFRAS uses face recognition technology to instantly and automatically take count of the students who are present in the class, while only requiring the teacher to press a button and the students to face the camera. This system can be implemented in any field where attendance management is vital.

Our system needs to pass certain tests and performance benchmarks to make sure it is effective, these evaluation methods include detection speed, accuracy, number of false positives and negatives, and many more. Should our system pass these tests, it will prove to be an effective tool for accurate and quick attendance taking that suffers none of the fallbacks other systems face. If implemented correctly, it would increase productivity of any institute or organization.

TABLE OF CONTENTS

[Declaration ii](#_Toc89566585)

[Certificate iii](#_Toc89566586)

[Acknowledgements iv](#_Toc89566587)

[Abstract v](#_Toc89566588)

[1. Introduction 16](#_Toc89566589)

[1.1 Preamble 16](#_Toc89566590)

[1.2 Motivation 16](#_Toc89566591)

[1.3 Problem Statement 16](#_Toc89566592)

[1.4 Project Objective 17](#_Toc89566593)

[1.5 Scope of Study 17](#_Toc89566594)

[1.6 Research Methodology 18](#_Toc89566595)

[1.7 Face Detection and Face Recognition 18](#_Toc89566596)

[1.8 Definitions and Acronyms 18](#_Toc89566597)

[2. Literature Overview 19](#_Toc89566598)

[2.1 Examples of Facial Recognition Technologies 20](#_Toc89566599)

[2.2 Specific Difficulties 21](#_Toc89566600)

[3. System Study Analysis 22](#_Toc89566601)

[3.1 Existing System 22](#_Toc89566602)

[A. Fingerprint Based Attendance System 22](#_Toc89566603)

[B. RFID Based Attendance System 22](#_Toc89566604)

[C. Iris Based Attendance System 22](#_Toc89566605)

[3.2 Proposed System 22](#_Toc89566606)

[3.2.1 Proposed System Components 23](#_Toc89566607)

[3.2.2 Proposed System Outcome 23](#_Toc89566608)

[3.3 Implementation 23](#_Toc89566609)

[A. Image Capturing 23](#_Toc89566610)

[B. Face Detection 24](#_Toc89566611)

[C. Pre-Processing 24](#_Toc89566612)

[D. Training 24](#_Toc89566613)

[E. Attendance 24](#_Toc89566614)

[3.4 Feasibility Analysis 25](#_Toc89566615)

[A. Economic Feasibility 25](#_Toc89566616)

[B. Technical Feasibility 25](#_Toc89566617)

[C. Operation Feasibility 25](#_Toc89566618)

[3.5 Functionalities 25](#_Toc89566619)

[3.6 Algorithms 26](#_Toc89566620)

[A. Viola-Jones 26](#_Toc89566621)

[B. LBPH 31](#_Toc89566622)

[3.7 Evaluation Measures 34](#_Toc89566623)

[4. Requirement Analysis & Specification 36](#_Toc89566624)

[4.1 Introduction 36](#_Toc89566625)

[A. Purpose 36](#_Toc89566626)

[B. Intended Audience and Use 36](#_Toc89566627)

[C. Team Members 36](#_Toc89566628)

[D. Scope 36](#_Toc89566629)

[4.2 Overall Description 37](#_Toc89566630)

[A. Product Perspective 37](#_Toc89566631)

[B. Background 37](#_Toc89566632)

[C. Product Functions 37](#_Toc89566633)

[D. User Classes and Characteristics 38](#_Toc89566634)

[E. General Constraints 39](#_Toc89566635)

[F. Assumptions and Dependencies 39](#_Toc89566636)

[4.3 System Features and Requirements 39](#_Toc89566637)

[A. Functional Requirements 39](#_Toc89566638)

[C. External Interface Requirements 40](#_Toc89566639)

[D. Student Requirements 42](#_Toc89566640)

[E. Teacher Requirements 43](#_Toc89566641)

[F. Administrator Requirements 43](#_Toc89566642)

[G. System Modules 43](#_Toc89566643)

[H. Non-functional Requirements 44](#_Toc89566644)

[4.4 Tools and Technologies 45](#_Toc89566645)

[A. Tools 45](#_Toc89566646)

[B. Technologies 46](#_Toc89566647)

[5. Software Design 48](#_Toc89566648)

[5.1 Use Case Diagram 48](#_Toc89566649)

[A. Specification of Actors 49](#_Toc89566650)

[B. Specification of Use Cases 49](#_Toc89566651)

[5.2 Data Flow Diagrams 52](#_Toc89566652)

[5.3 Process Flowchart 56](#_Toc89566653)

[5.4 Architectural Design 57](#_Toc89566654)

[5.5 Sitemap 59](#_Toc89566655)

[5.6 User Interface 59](#_Toc89566656)

[5.7 Activity Diagram 63](#_Toc89566657)

[5.8 Database Design 64](#_Toc89566658)

[5.9 UML Class 65](#_Toc89566659)

[5.10 Registration Process 66](#_Toc89566660)

[5.11 Attendance Process 67](#_Toc89566661)

[6. System Implementation 68](#_Toc89566662)

[6.1 Module Description 68](#_Toc89566663)

[6.2 Implementation Requirements 68](#_Toc89566664)

[6.3 Technology Used 68](#_Toc89566665)

[6.4 Implementation of Face Encoding 69](#_Toc89566666)

[6.5 Code Details of Face Encoding 69](#_Toc89566667)

[6.6 Implementation of Face Detection and Recognition 71](#_Toc89566668)

[6.7 Implementation of Flask 73](#_Toc89566669)

[6.8 Implementation of Front End 75](#_Toc89566670)

[6.9 Structure of GUI 75](#_Toc89566671)

[7. System Testing 78](#_Toc89566672)

[7.1.1 Acceptance Testing 78](#_Toc89566673)

[7.1.2 Beta Testing 78](#_Toc89566674)

[7.1.3 Code Review 78](#_Toc89566675)

[7.1.4 Functional Testing 78](#_Toc89566676)

[7.1.5 Integration Testing 78](#_Toc89566677)

[7.1.6 Performance Testing 79](#_Toc89566678)

[7.1.7 Regression Testing 79](#_Toc89566679)

[7.1.8 Software Testing 79](#_Toc89566680)

[7.1.9 Stress Testing 79](#_Toc89566681)

[7.1.10 System Testing 79](#_Toc89566682)

[7.1.11 Unit Testing 79](#_Toc89566683)

[7.1.12 Usability Testing 80](#_Toc89566684)

[7.1.13 Validation 80](#_Toc89566685)

[7.1.14 Verification 80](#_Toc89566686)

[7.2 Test Environment 80](#_Toc89566687)

[7.3 Test Cases 81](#_Toc89566688)

[7.4 Testing Result (Output) 84](#_Toc89566689)

[7.5 Deployment of System 87](#_Toc89566690)

[7.6 Maintenance 87](#_Toc89566691)

[8. Benefits of Afras 88](#_Toc89566692)

[8.1 Easy to Install 88](#_Toc89566693)

[8.2 Eliminates Manual Attendance 88](#_Toc89566694)

[8.3 Eliminates Proxy Attendance 88](#_Toc89566695)

[8.4 High Success Rate 88](#_Toc89566696)

[8.5 Saves Money 88](#_Toc89566697)

[8.6 Saves Time 88](#_Toc89566698)

[8.7 Secure 89](#_Toc89566699)

[8.8 Contact Less System 89](#_Toc89566700)

[9. Limitations 90](#_Toc89566701)

[10. Conclusion 91](#_Toc89566702)

[11. Future Enhancement 92](#_Toc89566703)

[12. References 93](#_Toc89566704)

List of Figures

[Figure 1.1 Diagram of General Framework of the System 16](#_Toc89566364)

[Figure 3.1 Proposed System Implementation 24](#_Toc89566365)

[Figure 3.2 Types of Haar-like features 27](#_Toc89566366)

[Figure 3.3 Haar Feature Invalid Example Figure 3.4 Haar Feature Valid Example 27](#_Toc89566367)

[Figure 3.5 Haar-Life Feature Working 28](#_Toc89566368)

[Figure 3.6 Conversion of Integral Image 29](#_Toc89566369)

[Figure 3.7 Strong Feature Weak Feature 30](#_Toc89566370)

[Figure 3.8 Cascade Classifier 30](#_Toc89566371)

[Figure 5.1 Functional Requirement Use Case Diagram 47](#_Toc89566372)

[Figure 5.2 DFD: Context Level 51](#_Toc89566373)

[Figure 5.3 DFD: Level 1 (1.0 Administrator Activities) 52](#_Toc89566374)

[Figure 5.4 DFD: Level 1 (2.1 Teacher Activities) 53](#_Toc89566375)

[Figure 5.5 DFD: Level 1 (1.0 Student Registration) 53](#_Toc89566376)

[Figure 5.6 DFD Level 0 54](#_Toc89566377)

[Figure 5.7 Process Flowchart 55](#_Toc89566378)

[Figure 5.8 Architectural Design 56](#_Toc89566379)

[Figure 5.9 Sitemap 58](#_Toc89566380)

[Figure 5.10 Homepage 58](#_Toc89566381)

[Figure 5.11 Registration Page 59](#_Toc89566382)

[Figure 5.12 Login Page 59](#_Toc89566383)

[Figure 5.13 Menu Page 60](#_Toc89566384)

[Figure 5.14 Student Attendance Record 60](#_Toc89566385)

[Figure 5.15 Attendance Report 61](#_Toc89566386)

[Figure 5.16 Activity Diagram 62](#_Toc89566387)

[Figure 5.17 ERD Diagram 63](#_Toc89566388)

[Figure 5.18 Relational Database 63](#_Toc89566389)

[Figure 5.19 UML Class 64](#_Toc89566390)

[Figure 5.20 Registration Process 65](#_Toc89566391)

[Figure 5.21 Attendance Process 66](#_Toc89566392)

[Figure 6.1 AFRAS Dashboard 74](#_Toc89566393)

[Figure 6.2 System in real time marking attendance 75](#_Toc89566394)

[Figure 6.3 Showing Recorded Students 76](#_Toc89566395)

[Figure 6.4 Attendance Report 76](#_Toc89566396)

[Figure 7.1 Successfully Ran the System 83](#_Toc89566397)

[Figure 7.2 Fully Functional Dashboard 83](#_Toc89566398)

[Figure 7.3 Encoding Complete 84](#_Toc89566399)

[Figure 7.4 Camera Turns on and Detects Faces Successfully 84](#_Toc89566400)

[Figure 7.5 Recognition Completed and Marked Successfully 84](#_Toc89566401)

[Figure 7.6 Upload Successful 85](#_Toc89566402)

[Figure 7.7 Images in Dataset 86](#_Toc89566403)

List of Tables

[Table 2.1 Factors which Cause Issues with Face Detection 19](#_Toc89566559)

[Table 3.1 Example 32](#_Toc89566560)

[Table 3.2 Example 32](#_Toc89566561)

[Table 3.3 Ground Truth Matrix 33](#_Toc89566562)

[Table 4.1Backend Hardware Requirements 40](#_Toc89566563)

[Table 4.2 Frontend Hardware Requirements 40](#_Toc89566564)

[Table 5.1 Admin UCD 48](#_Toc89566565)

[Table 5.2 Teacher UCD 48](#_Toc89566566)

[Table 5.3 Student UCD 48](#_Toc89566567)

[Table 5.4 Register UCD 48](#_Toc89566568)

[Table 5.5 Login UCD 49](#_Toc89566569)

[Table 5.6 Attendance Report UCD 49](#_Toc89566570)

[Table 5.7 CRUD Functions UCD 49](#_Toc89566571)

[Table 5.8 Activate System UCD 49](#_Toc89566572)

[Table 5.9 Create New Accounts UCD 50](#_Toc89566573)

[Table 5.10 Add Student Photo UCD 50](#_Toc89566574)

[Table 5.11 Detect Face UCD 50](#_Toc89566575)

[Table 5.12 Manual Attendance UCD 50](#_Toc89566576)

[Table 5.13 Logout UCD 50](#_Toc89566577)

[Table 7.1 Test Environment 79](#_Toc89566578)

[Table 7.2 Test Case 01 80](#_Toc89566579)

[Table 7.3 Test Case 02 80](#_Toc89566580)

[Table 7.4 Test Case 03 81](#_Toc89566581)

[Table 7.5 Test Case 04 81](#_Toc89566582)

[Table 7.6 Test Case 05 82](#_Toc89566583)

[Table 7.7 Test Case 06 82](#_Toc89566584)

# Introduction

## Preamble

We live in a society where technology is rapidly developing in various sectors and organizations. Attendance system is required in every organization and institute to monitor the presence of its employees and students. There are two methods of monitoring, manually and automated. Many institutes and organizations still use manual methods of attendance which wastes time and effort. Some do use automated methods such as biometrics or RFID (Radio Frequency Identification), but these come with their own constraints and issues. Therefore, we propose an automated facial recognition attendance system.

Face recognition technology has become popular in recent years. It has various applications today, such as unlocking smart phones and controlling access to sensitive places such as bank vaults.

## Motivation

We got this project idea from witnessing various instances where in college, students would ask others to swipe their cards for them and skip classes, they would often go unnoticed by the teachers, we also noticed that such an unreliable system forces teachers to manually take attendance regardless, which wastes precious time, therefore, we decided this could potentially be a good project to undertake and prevent such things from happening. Another reason is that the field of image processing, facial recognition, artificial technology has great scope, and it would help us develop new skills which may become useful in the near future.

## Problem Statement

The traditional methods such as a teacher calling out each student’s name or checking identification cards are troublesome; they are not only time consuming but also disruptive to the overall environment of the class. Sometimes other students act as proxies for their friends and in large classrooms it could be difficult for a lecturer to identify each student. These are just some of the issues with current systems. Our proposed automated system would eliminate these traditional methods.

## Project Objective

The objective of this project is to create an automated facial recognition attendance system which is faster than any other methods of attendance taking. We need to ensure the system can recognize faces accurately. We also need to develop a database for attendance management, and it should have enough memory to store records. The user interface needs to be user-friendly for admins and teachers to use.

To do this, the system must meet the following criteria:

1. Detect the face of someone from a video camera or an image, whether they are alone or in a crowd.
2. Extract facial characteristics in real-time using machine learning and artificial intelligence by analysing 80 node points on the human face.
3. Compare it with existing data and find a match.
4. Record attendance if a match has been found.

Figure 1.1 Diagram of General Framework of the System

## Scope of Study

With this project we would be studying:

* Machine Learning
* Face Detection Algorithms
* Face Recognition Algorithms
* Front-End and Back-End Web Development
* Computer Vision and Digital Image Processing
* Database Management

## Research Methodology

Our aim is to develop an automated attendance system using image processing method. We would need to develop an efficient face recognition algorithm that would be able to recognize each student’s face in an image of the whole class. We would also need to create a secure and user-friendly interface in the form of a website for teachers and administrators.

## Face Detection and Face Recognition

Face detection is an AI based technology used to identify human faces in digital images. There are many applications of face detection such as biometrics, security, and law enforcement to allow surveillance of people in real-time. Overtime, it has improved and plays a vital role in applications such as face recognition and face analysis.

Face recognition goes beyond detecting the presence of a human face, it can identify who the person is. With the help of computer application, a captured image is compared with images stored on a database. Facial recognition is widely used today for unlocking smart phones and mobile wallets.

## Definitions and Acronyms

* **AFRAS-**Automated Facial Recognition Attendance System
* **SRS-** Software Requirement Specification
* **Student-** User whose face is to be recognized.
* **Admin-** Person responsible for handling the system and enrol students’ image into the database.
* **GUI-** Graphical User Interface
* **IDE-** Integrated Development Environment, consists of editor and compiler used to write programs and compile them.
* **AI-** Artificial Intelligence

# Literature Overview

There are number of similar existing systems like our idea of automated facial recognition attendance systems in an institute using various algorithms and facial recognition techniques. We researched and read some of those peoples’ published works.

Reddy et al. (2018) took the idea of two systems, student attendance and feedback system and implemented them using machine learning. The system keeps track on student’s performance, attendance, and feedback for each course, therefore, students can not only can record their attendance, but also gain information about their marks from this system.

There is a significant difference between face recognition and face detection. Face detection is used to see specific regions of the face whereas face recognition identifies actual persons from the facial image. Two authors Chao [1] and S.Aanjanadevi et al. [2] presented issues with face detection and recognition illustrated in table 1.1.

|  |  |
| --- | --- |
| Background | **Different backgrounds and environment while using face recognition might alter the effectiveness of the system** |
| Lighting | Various lighting environments can affect detection of facial features. A face recognition tool might struggle to match two photos of the same person if one is taken in a well-lit environment and the other in a dark environment. |
| Posture | Different postures when capturing the image can cause distortion during recognition process, particularly when using recognition methods such as Eigen face and Fisher face. |
| Occlusion | Which means blockage, it can happen if a part of a face is obscured, this can occur due to the nature of the photo taken, or it can happen naturally if a person wears glasses or grows a beard or longer bangs. |
| Expression | Changes in expressions can also cause issues when using the facial recognition systems. |
| Extraction | While extracting the image, system would rotate, scale, and translate the image which could cause misrepresentation when comparing it to the original image. |
| Resolution | To perform an accurate face recognition, the image or video needs to be captured by a high-resolution camera, which are rare in security environments like CCTV and ATMs where facial recognition could otherwise be used to catch a thief or criminal or exonerate an innocent person. |
| Performance | The newest facial recognition methods use Convolutional Neural Networks, which are very complicated and not suitable for day to day or real time applications. |

Table 2.1 Factors which Cause Issues with Face Detection

Ara et al. proposed the use of Conventional Neural Networks to detect and capture images of student’s facial features. They trained their model with CNN and used Support Vector Machine to classify their trained images. Accuracy of their system was 95% [3].

Liu and Wang developed an application that uses mobile terminals and GPS to track student’s attendance. Their system was based on android application which didn’t just mark attendance but also had other functionalities such showing student’s timetable, grades etc [4].

Chintalapati and Raghunadh proposed an Automated Attendance System that uses Facial Recognition to mark the attendance of students as soon as they enter the class. Viola-Jones algorithm was used to detect faces. The system helped teachers monitor their students and saved time when compared to traditional methods [5].

Kawaguchi et al. proposed a Face recognition-based Lecture Attendance System which monitors students continuously and based on the data it estimates seating position and location of each student. However, an issue with this system was that students were not allowed to move during the class or change their seating position [6].

## Examples of Facial Recognition Technologies

* **Academia:** Researchers at The Chinese University of Hong Kong developed The GaussianFace algorithm in 2014, which achieved a facial identification score of 98.52%, which means it’s better at recognizing and matching faces than humans are who achieve a score of 97.53%.
* **Facebook:** DeepFace announced in 2014, can tell if the same person appears in multiple photos with an accuracy rate of 97.52%.
* **Google:** FaceNet in 2015, performed even better than DeepFace, using the Labelled Faces in the Wild (LFW) dataset, an artificial neural network (ANN) and a new algorithm, it has a much more impressive accuracy of 99.63%, much better than humans. It’s used in Google Photos to automatically sort, and tag photos based on who appears in them.
* **OpenFace:** An unofficial open-source version of FaceNet.
* **Amazon:** Amazon Rekognition, a cloud-based face recognition service promoted to law enforcement agencies. This impressive technology can take an image comprising hundreds of people and match them against a database.

## Specific Difficulties

* An MIT Study found that some facial recognition tools had high error rates when attempting to identify people of darker complexion.
* Amazon’s Recognition was alleged to have falsely identified 28 US Congress members as people who have been previously arrested for crimes.

# System Study Analysis

## Existing System

## 

### Fingerprint Based Attendance System

In a biometric system such as fingerprint scanning, devices need to be configured beforehand to detect each student’s fingerprint, but they might struggle to detect a student’s fingerprint if they have wet or dirty hands [7]. Additionally, incorrect acceptance and incorrect rejections are common issues with biometric systems.

### RFID Based Attendance System

In a Radio Frequency Identification System, students are given an RFID card which is used to mark attendance when it is placed near the card reader. Katara et al. believed that RFID systems are often installed due to their simplicity, however, the system has two major issues: Firstly, it is easy to cheat; Students can give their cards to their peers to scan and ensure their attendance is recorded [8]. Secondly, students often forget their cards, hence their attendance is not recorded which causes a hassle as teacher must then do it manually [7].

### Iris Based Attendance System

In this system, students would stand in front of a camera while it scans their eye. This system does not require any physical contact and is more accurate than facial recognition systems, however, this type of system is expensive as it uses special cameras and requires the person being scanned to be at a very close proximity to the camera which can be a difficult and slow process.

## Proposed System

We aim to build a platform independent automated attendance system where facial recognition algorithms can be used to record the attendance of students in educational institutes without any human interference. The way this system would work is by using a camera to capture a student’s face and compares it with facial images stored on the database. If the system finds a matching image, it will mark the student as present, if the software does not recognize the student, it will display an invalid input. The goal of this project is to make tracking attendance in institutes more efficient, convenient, and simple compared to traditional methods. The proposed system will have a user-friendly website and a secure database.

### Proposed System Components

The following are the components for the proposed system:

* Login
* Face Detection
* Face Recognition
* User Records
* Attendance Report
* Attendance Management System

The system will allow users to add, update, delete, and retrieve content from the system. The management system will be able to mark attendance automatically, print attendance records and record details of attendance.

### Proposed System Outcome

* Allow only authenticated users to use the system with login credentials.
* Take student attendance with facial recognition technology.
* Facial recognition will work by detecting faces through webcam/ video camera.
* Once recognition is complete, student will be marked present in the attendance report.
* Users will be able to access student records anytime.

## Implementation

### Image Capturing

Every student will first register and have images of their face taken using a video camera in the classroom. A camera will be placed in a class near the entrance so it can capture facial images of students efficiently and will be connected to the computer through a wireless medium, however in our prototype we will be using an in-built computer webcam.

### Face Detection

Student’s face will be detected if the camera is able to capture it. Upon successful face detection, the frontal face would be cropped and saved as a new image in the dataset. We will be using OpenCV because it comes with a dataset trainer and detector. We have chosen Viola-Jones algorithm for face detection as its fast and robust [9].

### Pre-Processing

Detected face is then extracted and pre-processed with histogram equalization. Histogram Normalization would improve contrast of the image to make it clearer.

### Training

For facial recognition to work, we need to train the dataset images. The dataset is the collection face images of each user. Every individual student’s images needs to be captured in different angles, expressions, lighting conditions and stored on the database.

### Attendance

To record the attendance, a tutor would activate the system at the start of a class, then a camera will begin capturing images of students. If a student’s face is captured, then it would compare it against the dataset. If there is a match, then the student’s name and ID would be shown and saved in a report along with recorded time of entrance.

Chart, diagram

Description automatically generated

Figure 3.1 Proposed System Implementation

## Feasibility Analysis

### Economic Feasibility

This system would be cost-effective because it would eliminate manual attendance systems where lot of paper is wasted. It would also be time effective because all attendance would be recorded online and be easily accessible when needed hence teachers don’t need to waste time looking for a particular attendance files or papers.

### Technical Feasibility

We have studied many research papers regarding facial recognition technology and found many algorithms and libraries which have successfully implemented them in attendance systems. Furthermore, there are many real-life applications where facial recognition and simulation is used such as Instagram filters and snapchat. The system we are developing is not a new system. There are many existing applications on it, and it has been a popular topic in research papers in recent years. Therefore, we think this system is feasible.

### Operation Feasibility

The system would be user-friendly and easy to learn hence users would not be required to go through any formal training to use the system.

## Functionalities

The users of this system are students, lecturers and admin. Lecturer would decide when to start the image capturing process, admins would be the ones who would manage and update the dataset, students would not get any permission to edit the system and only be allowed to view their attendance.

**Lecturer:**

* Start the process.
* View all student’s attendance.
* Manual attendance marking if needed.

**Students:**

* View attendance.

**Administrator:**

* View attendance.
* Fully control the system.
* Perform CRUD (Create, Retrieve, Update, Delete) functions.
* Train the system.
* Manage classes and schedules.
* Add students and teachers.

## Algorithms

### Viola-Jones

There are many methods available for face detection such as FloatBoost algorithm, S-AdaBoost algorithm, Bayes Classifier and Support Vector Machines, CNN [10]. But for face detection, we choose the Viola-Jones algorithm. Viola-Jones algorithm was founded in an article in 2004 by Paula Viola and Michael J. Jones named “Robust Real-Time Face Detection”. The algorithm is so successful that it was an obvious choice for face detection errands [9]. Apple used Viola-Jones algorithm between 2014-2015 and Snapchat still uses Viola-Jones algorithm for their application.

Although Viola-Jones algorithm is outdated and slow in training compared to other algorithm, its real-time processing is fast according to Mayank Chauhan et al. (2014). Viola-Jones has a detection rate of 97.41% hence it is one of the most accurate methods in image processing.

When Viola Jones was first introduced, people thought it was revolutionary as it was one of the first object detection framework that detected objects in real-time [11]. The algorithm has four phases which are selecting Haar-Like features, creating integral image, running Adaboost training and creating classifier cascades [9].

Viola-Jones inspired many face detection methods and one of them was Conventional Neural Network (CNN) which can detect faces in different positions and faster than Viola-Jones can, however it does not mean Viola-Jones is inferior to CNN. Both algorithms follow a series of steps, however the steps in CNN are much less structured than in Viola-Jones where they are set. CNN also needs large amounts of information which requires more storage than Viola-Jones. Even though CNN is faster than Viola-Jones it is expensive to implement, therefore we choose Viola-Jones [12].

The way the algorithm works is as follows:

1. Input desired image.
2. Haar feature would detect features of the face.
3. Integral image would be used to fasten processing time.
4. AdaBoost training would be used to locate, train the features and enhance processing time.
5. Lastly, cascading would be used classify whether window contains a face or not.

All the phases are explained in detail below.

1. **Haar:** Viola-Jones uses variety simple Haar-Like features to classify images. Haar-Like features are a set of rectangular digital image features which break up into multiple parts and are masked over an image. Rectangles are visualized in black and white.

OpenCV has built-in haar-like classifier, it is used to find human faces present in an image. All humans have darker and lighter regions on their faces, for example, the eye region is dark, but the forehand region is lighter, to find out which value is light or dark, we compare both regions by summing up their pixel values. Total number of pixels in darker region would be less than total number of pixels in the lighter region. With Haar-like feature we can interpret different regions of the human face [13].

Viola and Jones identified three haar-like features:

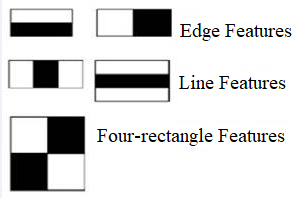


Figure 3.2 Types of Haar-like features

Within each rectangle submission of a pixel is calculated and then difference between black and white region is calculated.

For example, haar feature in fig 3.3 is useless as it does not overlap the face, however in fig 3.4 intersects the face showing eye regions so its useful.

A person with long hair

Description automatically generated with low confidence A picture containing text, person, person, outdoor

Description automatically generated

Figure 3.3 Haar Feature Invalid Example Figure 3.4 Haar Feature Valid Example

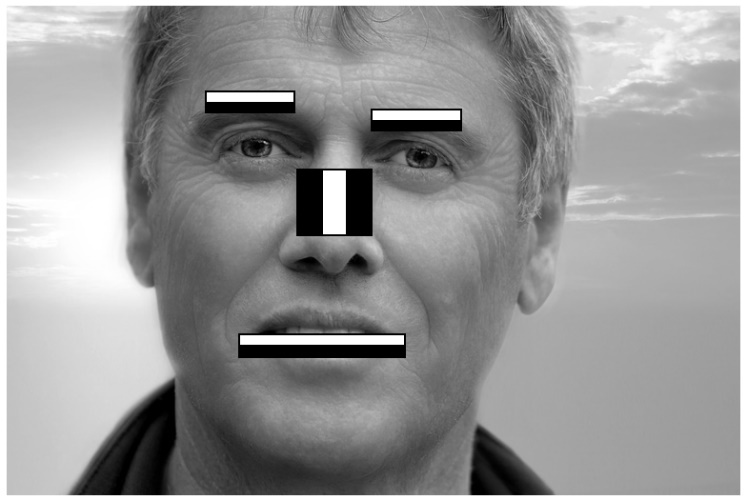


Figure 3.5 Haar-Life Feature Working

Edge and line features are used to detect edges and lines of faces. Whereas four-rectangle features are used to find diagonal features.

When haar detects a human face, it would automatically draw a rectangular box around it. It is an effective method of identifying objects and was proposed by Viola and Jones [14]. It is a machine learning method where a cascade function is trained by many positive and negative images, this is called the extraction process in the algorithm.

In 24x24 window, there are over 160,000 features, however not all of them are useful, hence addition and subtraction of those features is compulsory to reduce processing time.

1. **Integral Image:** A data structure which allows us to calculate the total number of pixel values in an image or a rectangular region of it. Viola-Jones created integral image to make the processing time faster. “An Integral Image is an intermediate representation of an image where the value for location (x, y) on the integral image equals the sum of the pixels above and to the left (inclusive) of the (x, y) location on the original image” [15]. The intermediate representation is essential as it allows for faster calculation time of rectangular regions.

Diagram, schematic

Description automatically generated

Figure 3.6 Conversion of Integral Image

1. **AdaBoost:** Adaptive Boosting is a machine learning algorithm which identifies the best features present in the detector window (24x24 rectangle) [16]. Viola and Jones algorithm detect 160,000 features, but only a few of these features would be useful. After features have been selected, they are tested with AdaBoost learning algorithm to find out all best features that need to be looked through. AdaBoost training assures region of faces that are examined are precise to obtain best accuracy. AdaBoost combines weak classifiers into one powerful classifier.Training is intensive because all the different possibilities and combinations need to be checked frame by frame.

The distinguishment between most useful features and useful features is done by strong feature weak feature classifier formula. In fig 3.7, you can see f1, f2, f3 which represent features and a1, a2, a3 which represent weights of the features. Features are known as weak classifiers and F(x) is known as strong classifier. By combining the weak classifiers, we would get one strong classifier. As, we keep adding the classifiers together, it gets stronger, and it is called an ensemble [17].

A picture containing text

Description automatically generated

Figure 3.7 Strong Feature Weak Feature

1. **Cascade Classifier:** A multi-stage classifier which performs detection quickly and accurately. Different stages are composed together to form a strong classifier from AdaBoost. Each stage determines whether a face is a real face or a non-face. If it is a non-face that it would be deleted and if it is a face than it would be passed on to the next stage for further testing. To achieve this, cascade classifier eliminates many numbers of false positives so that many numbers of correct detections can be made.

Diagram

Description automatically generated

Figure 3.8 Cascade Classifier

### LBPH

Local Binary Patterns Histograms (LBPH) is one the simplest and popular face recognition algorithm, which comes with the OpenCV library. LBP is a “simple yet very efficient texture operator which labels the pixels of an image by thresholding the neighbourhood of each pixel and considers the result as a binary number.” [18].

LBP are visual descriptors for what a computer might see, it was first described in 1994 and has proven to be good at texture classification. It was also later combined with Histogram of oriented gradients (HOG), another descriptor, which improved its performance considerably.

The images taken would be trained using the LBPH algorithm. When facial images have been extracted, they would be converted to greyscale, this face recognition algorithm would find characteristics which best describe the image [19]. LBPH has been chosen because it takes less load on the computer, it is fast and can be used in real-time environments.

1. **Process:**

LBP creates a feature vector with this process:

1. Convert image to greyscale.
2. Divide image into cells, for example, 32x32 pixels per cell.
3. Pick a pixel, and compare its value (How bright or dark it is) with its eight neighbours (All directions as well as the pixels on the diagonal)
4. If the value of the centre pixel is greater than or equals a neighbour, write 1, otherwise write 0, do this for all 8 pixels in a circle, you will end up with an 8-digit binary number.
5. Repeat steps 3 and 4 for each cell and pixel.
6. You will now have a byte for each pixel which tells you how bright that pixel is relative to its neighbours.
7. Computer the histogram, or the frequency of each number occurring (For example, how many times is the pixel above the centre pixel is brighter than it, etc.). The histogram would have a dimension of 256.
8. Normalize the histogram (Optional).
9. Concatenate all histograms, leaving you with one vector for the whole image.

We can also label this vector with a binary byte based on expression, for example, a person smiling might have a label of 10010111, the same person having a neutral face will have a different label. We can now use these histogram vectors and their labels to train our facial recognition algorithm [20].

1. **Example:**

A certain pixel with 8 neighbours could have the following pixel values:

|  |  |  |
| --- | --- | --- |
| 32 | 36 | 6 |
| 20 | **12** | 5 |
| 14 | 9 | 4 |

Table 3.1 Example

Here the centre pixel has a pixel value (brightness) of 12, if we wish to create an LBP, we must compare 12 with each neighbour, and replace the value of the pixel with a 1 or 0 depending on its relative luminance:

|  |  |  |
| --- | --- | --- |
| 0 | 0 | 1 |
| 0 |  | 1 |
| 0 | 1 | 1 |

Table 3.2 Example

For this LBP, let us start from the top pixel and go in a clockwise motion, the binary number it gives us will be 01111000, in decimal this will be 120.

Now we assign that value of 120 to the centre pixel and repeat the process for each pixel in the image [21].

1. **Features:**

An important feature of LBP is that it’s luminance invariant, for example, if you take an image in a dark environment, then turn on the light and take the same image again, the value of each pixel will be different, but the LBP histogram will be the same, that’s because LBP measures the **relative** luminance of neighbouring pixels, and they will all change by the same amount. If say the pixel above the centre pixel in a cell is brighter than the centre pixel, it will always be brighter regardless of the light level, and the same applies for all pixels and most cases.

Another important feature of LBP is that it can be used to easily identify edges, which is essential in face recognition and detection. Using the above example in table 2, we see that there is a sequence of 0’s followed by a sequence of 1’s, these sequences can tell you exactly where an edge is.

## Evaluation Measures

**Ground Truth Notion:**

Determine how true each positive and negative detection is by using the confusion matrix:

|  |  |  |
| --- | --- | --- |
| Ground Truth \ Detection | Positive | Negative |
| Relevant | True Positive (TP) | False Negative (FN) |
| Nonrelevant | False Positive (FP) | True Negative (TN) |

Table 3.3 Ground Truth Matrix

This is used to determine if the system can accurately differentiate between a face and a nonface.

* **True Positive:** System says two faces match – They are of the same person.
* **False Positive:** System says two faces match – They are of different people.
* **False Negative:** System says two faces do not match – They are of the same person.
* **True Negative:** System says two faces do not match – They are of different people.

False positives and negatives are also applicable when the system detects a face in an image when there is no face there [22].

**Precision:** Also known as reliability or repeatability: A measure of how accurately the system matches a face to a person given a certain number of images [23].

**Recall**: Proportion of positive matches that were properly identified [23].

**Accuracy:** System would be measured by number of correct predictions it makes over total number of predictions made [22].

Using just these three equations will provide a good metric of how well the face recognition system performs. Though there are many others such as fallout, F-measure, error rate, effectiveness, etc [22].

# Requirement Analysis & Specification

## Introduction

### Purpose

This SRS describes what the system will do and what outcome users could expect from it. The purpose of this documentation is to define objectives and get work on the project started. The intended audience for this document are the project developers who would use it as a guide to ensure successful implementation and functioning of the final product. Another intended audience would be the course supervisors and their respective peers.

### Intended Audience and Use

This document has been created for developers, administrators, users, testers, documentation writers, and the project advisor(s). This SRS describes how users would be able to interact with the system as well as its limitations and restrictions.

### Team Members

The project team consists of three members responsible for building this project:

1. Haitham Eliyas
2. Qusai Al Saidi
3. Nooh Al Balushi

### Scope

The scope of this project is to build a modern sophisticated attendance system using facial recognition technology. This project also includes web development and database management. It will provide an automated attendance system that is reliable, efficient, easy to use, and able to prevent any of the usual disturbances caused by traditional methods of recording attendance.

The system should be able to detect the faces of students in different environmental conditions, extract unique facial characteristics amidst other characteristics such as glasses, beards etc., automate the attendance system without any human interference and send an alert to the teacher if a student has missed more than three classes. With this system, education institutes would have a secured system with authentication.

## Overall Description

### 

### Product Perspective

AFRAS is a web-based system. The system will interface with two other systems: A database where student’s information will be stored, and browsers used by teachers. This system will automate attendance process, ensure accurate time records, and keep everything well organized and secure.

### Background

The purpose of this AFRAS is to replace existing forms of attendance systems such as RFID, biometrics, and manual marking. It will help institutions to record student’s attendance in real-time and is fool proof. The system will also contain other features which would make it efficient and a good replacement for traditional methods of attendance.

### Product Functions

AFRAS will have two modes of access:

1. Administrator
2. User
   1. Student
   2. Teacher

*Admin* *will be able to perform following functions:*

* Login
* Register students into the system.
* Train the model by adding student image into the dataset.
* Create accounts for employees and grant access rights.
* View attendance report for all students.

*Teachers* *will be able to perform following functions:*

* Login
* Start the attendance process.
* View attendance report for all students.

*Students will be able to perform following functions:*

* Scan their faces.

### User Classes and Characteristics

* **Administrator:** Admin would be the person who has full control and access rights to AFRAS. They will be responsible for registering students into the system and create accounts for the teachers. They will also be responsible for managing all courses, as well as making sure students are enrolled in the correct classes. They can also change the attendance sheet if required, for example, if AFRAS fails to recognize a student due to unknown reasons, or if a student misses a class but provides a valid excuse, etc. If any teacher needs assistance in using the system, the admin will be the one to help.
* **Teachers:** Teachers will be given limited access rights to the system. They will be able to login into the system so they can activate recording of attendance when class starts. They will be able to check attendance sheets but will not be able to modify or delete any data.
* **Students:** They would be the ones mainly using the facial recognition system daily. Students will not be given access to the system directly but will be required to register and have pictures of their facial images taken for them to be able to use the system when attending classes.
* **Developers:** Developers will personally be accountable for success and failure of this system.Their responsibilities include planning, designing, executing, testing documenting, and finally, implementing the system.

### General Constraints

Constraints for this project are its schedule and scope. The project needs to be delivered according to the given timeline along with all documentation. The database needs to be able to handle a large number of students and train using all the images captured. The face recognition module needs to identify each student individually and record their attendance into the database.

### Assumptions and Dependencies

To use this system initially, students are expected to register themselves and get their pictures taken. Administrator will be provided with all this information, and they will be responsible for maintaining the system. Users who are given administrator rights need to be cautious when modifying or deleting any information intentionally or unintentionally as this could lead to inconsistency in the database. Teachers who will be operating the system should have the basic computer skills necessary to start the system when a class begins.

The detector module should crop only the face area and exclude other parts. It is assumed that after a successful detection; the module will compare the images stored on the database. So, we can assume recognition module works in recognizing the users and records them onto the database.

The whole recognition and detection process is dependent on the CascadeClassifer() which is core part of recognition and detection process. Haar Cascade is used to detect objects from its origin and in this case, it would detect frontal faces of users.

## System Features and Requirements

Below are descriptions of requirements for system services and constraints.

### Functional Requirements

Functional requirements are tasks and services system must provide.

* Administrator should be able to perform all CRUD (Create, Retrieve, Update, Delete) functions with the system.
* Administrator must enter data correctly and be able to easily manage all student records.
* Secure the system therefore only authorized users can access it.
* For the prototype, a laptop webcam will be used but the final system must be connected to a wireless external camera to be used in a classroom environment.
* Face recognition must have acceptable accuracy.
* Teachers who will be accessing the system must login into the website before using it.
* Teachers should be able to easily track student attendance.
* System should be capable of handling invalid inputs.
* Information needs to be entered correctly and must be managed properly.

### External Interface Requirements

#### User Interfaces

User interface of the system will be user-friendly and simple. There will not be any unnecessary elements, and clear language will be used. The system will be easily accessible and understandable. System will have consistent GUI elements such as screens, buttons, and pages. This will make users feel comfortable and able to learn how to use the system quickly. An appropriate colour scheme will be used so the system is aesthetically pleasing and easy to look at.

#### Hardware Interfaces

The hardware requirements for a computer to run the back end (Database, facial recognition processing) portion of the system:

|  |  |  |
| --- | --- | --- |
| Component | Minimum | Recommended |
| Processor | Dual Core | Hex Core |
| RAM | 2 GB | 16 GB |
| Disk | 64 GB | 512 GB |
| Network | 512 KB/s | 3 MB/s |
| Software | MySQL server | |
| Operating System | Windows Server, Linux, Unix | |

Table 4.1Backend Hardware Requirements

The hardware requirements for a computer to run the front-end (Image capture and transfer, website access) portion of the system:

|  |  |  |
| --- | --- | --- |
| Component | Minimum | Recommended |
| Processor | Dual Core | Quad Core |
| RAM | 2 GB | 8 GB |
| Disk | - | 512 GB |
| Network | 512 KB/s | 3 MB/s |
| Camera | 4 Mega-Pixel | 16 Mega-Pixel |
| Operating System | Any | |

Table 4.2 Frontend Hardware Requirements

A wireless camera will be required to perform high resolution image capture in class with necessary drivers installed on the operating system. However, in our prototype we will be using computer’s in-built webcam.

#### Software Interfaces

* **Server-Side:** AFRAS will be connected to Microsoft MySQL server database, where all student information will be stored.
* **Client-Side:** Website would be accessed through default web browser such as Google Chrome.

#### Communication Interface

* **Communication Standard:** HTTPS
* **Network Server:** Localhost
* **Web Browser:** Google Chrome

#### Performance Requirements

* The system needs to load within seconds and be usable.
* System needs to perform its task quickly and accurately.
* Database needs to be in normalized format to prevent data redundancy and to increase performance.
* System should be capable of handling large amounts of data.

#### Safety Requirements

To maintain safety of the system, a data backup will be regularly created and maintained.

#### Security Requirements

* Admin should give limited access rights to users.
* Keep logs on people who use the system.

### Student Requirements

* At time of registration students must enter all details correctly.
* Students needs to get 10 or more face images captured in different directions.
* Students must face the camera when using attendance system.

### Teacher Requirements

* Teachers must login into the system before using it.
* Course needs to be selected before attendance process.
* When students are recognized, attendance report will be created.

### Administrator Requirements

* Admin must login first before registering students into the system.
* Admin needs to ensure all student details are correctly entered.
* Admin has rights to alter the system in any way.
* Admin has rights to manage or update records on the database.

### System Modules

#### Realtime Webcam

Webcam would capture student’s facial features in real time to mark their attendance or for registration.

#### Face Detection

The system decides whether there is a face or not using Viola-Jones algorithm and if face has been found it would crop it from rest of the body.

#### Face Recognition

System would use LBPH algorithm on detected faces and compare it with stored images on the database. If match has been found, respective student’s attendance would be marked and saved.

#### Website

Website will have the following pages:

* 1. **Login:** To access the website and the system, teacher or admin will need to login into the website and be given access according to their access rights.
  2. **Homepage:** Once login credentials are verified, user will be introduced to this page.
  3. **Facial Recognition System:** Teachers access this page to activate facial recognition system so students can scan their faces.
  4. **Attendance Report:** This page will give teachers weekly attendance reports for each class. It gives teacher information about how many times a student has attended and how many classes they have missed.
  5. **Help:** Website will provide a help link on each HTML page to explain how to use that page.

### Non-functional Requirements

Following are characteristics on which this system shall be judged:

* **Accuracy:** System shall perform all tasks accurately to avoid any errors.
* **Availability:** This system is critical toany institute hence it should be available 99.9% of times.
* **Correctness:** Access to the system shall be given only to authorised users, for example, teachers will be given rights only to start attendance process and view attendance report. Only the admins should have complete access rights to the system.
* **Flexibility:** System must be flexible to any changes.
* **Maintainability:** Backups for system would be provided.
* **Responsiveness:** Execution of any task should be fast.
* **Reliability:** System shall be available 24/7 and be highly robust.
* **Security:** The system should only be controlled by administrators and authorized users. Admin is only person who can create accounts for teachers and provide them with a username and password. Only authorized personnel can access the system using login credentials.
* **Simplicity:** The GUI needs to be simple and intuitive.
* **Time:** The project needs to be completed within given time frame.
* **Usability:** System must be user-friendly where attendance taking, tracking of records, and updating can be done with ease without any errors.

## Tools and Technologies

### 

### Tools

The tools and software that will be used to develop this system are:

* **Visual Studio 2019:** Visual Studio is IDE which will be used to create our website using HTML, CSS and JavaScript.
* **VS Code:** Visual Studio Code combines the simplicity of a source code editor with powerful developer tooling, like IntelliSense code completion, support for development operations like debugging, task running, and version control. It aims to provide just the tools a developer needs for a quick code-build-debug cycle. We would be using this to build our face recognition algorithm and integrating it with the website.
* **Microsoft Word 2019:** MSWord processor will be used to write all documentation.
* **Microsoft Project 2019:** MS Project is a project management software, and we will be creating our Gantt chart with it to keep track of our work.
* **Microsoft Visio 2019:** Microsoft Visio 2019: MS Visio would be used to create software diagrams such as data flow diagrams, UML model diagram, use case diagram etc.

### Technologies

* **Python 3:** Python is a general-purpose programming language.
* **OpenCV:** Library of programming functions mainly aimed at real-time computer vision. We used OpenCV to capture the video stream, extract frames from the video, convert the image format, resize the image, and display the video back.
* **Python IDLE:** Integrated Development and Learning Environment is an IDE for python used to run and edit python script.
* **Dlib:** Dlib is a modern C++ toolkit containing machine learning algorithms and tools for creating complex software in C++ to solve real world problems. Package required for face\_recognition.
* **face\_recognition:** Recognize and manipulate faces from Python or from the command line with the world's simplest face recognition library. The main library for encoding, detecting, and recognizing faces.
* **Numpy:** 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. We used Argmin from Numpy to get the smallest face distance within a certain threshold in our comparison.
* **os:** This module provides a portable way of using operating system dependent functionality. A very comprehensive Python library that allows us to create and manipulate directories and files.
* **pickle:** The pickle module implements binary protocols for serializing and de-serializing a Python object structure. “Pickling” is the process whereby a Python object hierarchy is converted into a byte stream, and “unpickling” is the inverse operation, whereby a byte stream (from a binary file or bytes-like object) is converted back into an object hierarchy. We used this package to save and load our encodings array.
* **flask:** Flask is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions.
* **Jinja2:** A web template engine for the Python programming language. We used this to make our HTML pages display variables passed by our Python program.
* **pathlib:** This module offers classes representing filesystem paths with semantics appropriate for different operating systems. We used this to safely create a directory if it doesn't already exist.
* **HTML:** Hypertext Markup Language will be used to give structure to our website.
* **CSS**: Cascading Style Sheets is a language used to describe how HTML elements (colours, layout, font) should be illustrated.
* **JavaScript:** JavaScript is scripting language that will help us create interactive and dynamic content.
* **Bootstrap:** Opensource CSS framework used to develop front-end framework for websites.

# Software Design

## Use Case Diagram

Below in fig 5.1 is the functional requirements use case diagram which illustrates expected behaviour of the system. The purpose of UCA is to help developers visualize the functional requirements of the system by visualizing the “actors” (Users who would be using the system) and processes involved.

Diagram

Description automatically generated

Figure 5.1 Functional Requirement Use Case Diagram

### Specification of Actors

#### Admin

|  |  |
| --- | --- |
| Admin | |
| Element | **Details** |
| Description | Individual responsible for the system |
| Examples | System admin would be the one who would login into the system, register new students, capture their images, train the system, perform CRUD (create, retrieve, update, delete) functions. |

Table 5.1 Admin UCD

#### Teacher

|  |  |
| --- | --- |
| Employee | |
| Element | **Details** |
| Description | Teacher would be responsible for taking attendance. |
| Examples | Teacher would login, activate the system and view attendance reports. |

Table 5.2 Teacher UCD

#### Student

|  |  |
| --- | --- |
| Student | |
| Element | **Details** |
| Description | Individuals who would use facial recognition system. |
| Examples | Students would stand in front of camera, get their facial features detected and recognized. |

Table 5.3 Student UCD

### Specification of Use Cases

#### Register

|  |  |
| --- | --- |
| Register | |
| Element | **Details** |
| Actor | Admin |
| Trigger | Admin activates the camera so student’s face can be detected |
| Pre-Conditions | System menu is displayed, and student is not registered yet |
| Post-Conditions | Student is registered and his/her facial images are recognized. |
| Normal Course | Admin collects data and images from the student and stores them on database. |
| Alternative Course | Student already has been registered |

Table 5.4 Register UCD

#### Login

|  |  |
| --- | --- |
| Login | |
| Element | **Details** |
| Actor | Admin, Teacher |
| Trigger | Admin or Teacher decide to login into the system |
| Pre-Conditions | Not logged into the system |
| Post-Conditions | Login is successful and menu is displayed |
| Normal Course | 1. They go to the website and the login page is displayed. 2. They enter their username and password and click login. 3. Login is successful and the menu is displayed. |
| Alternative Course | Access is denied. |

Table 5.5 Login UCD

#### Attendance Report

|  |  |
| --- | --- |
| View Attendance Report | |
| Element | **Details** |
| Actor | Admin, Teacher |
| Trigger | Student face is recognized by the system. |
| Pre-Conditions | Image is captured and processed into the system. |
| Post-Conditions | If face exists in the system, then student is recorded as present and if not then absent. |

Table 5.6 Attendance Report UCD

#### CRUD Functions

|  |  |
| --- | --- |
| Perform CRUD Functions | |
| Element | **Details** |
| Actor | Admin |
| Trigger | Admin initialises any CRUD (Create, Retrieve, Update, Delete) operation. |
| Pre-Conditions | Admin logins into the system. |
| Post-Conditions | Records/Reports are modified. |

Table 5.7 CRUD Functions UCD

#### Activate System

|  |  |
| --- | --- |
| Activate System | |
| Element | **Details** |
| Actor | Admin, Teacher, Student |
| Trigger | Admin or Teacher activate the system so it can capture image of student |
| Pre-Conditions | 1. Student has been registered ex. student has not been recognized yet. 2. Student has not been registered ex. student has not been enrolled yet. |
| Post-Conditions | Student has been registered and they have been recognized. |
| Normal Course | Admin collects student’s facial images and personal data. |

Table 5.8 Activate System UCD

#### Create New Accounts

|  |  |
| --- | --- |
| Create New Accounts | |
| Element | **Details** |
| Actor | Admin |
| Description | Admin wants to create new account for teacher. |
| Pre-Conditions | Admin logins into the system. |
| Post-Conditions | New accounts created. |

Table 5.9 Create New Accounts UCD

#### Add Student Photo

|  |  |
| --- | --- |
| Add Student Photo | |
| Element | **Details** |
| Actor | Admin |
| Description | Admin wants to create new account for teacher. |
| Pre-Conditions | Admin logins into the system and webcam needs to be activated. |
| Post-Conditions | Image is uploaded on database and record has been created. |

Table 5.10 Add Student Photo UCD

#### Detect Face

|  |  |
| --- | --- |
| Detect Face | |
| Element | **Details** |
| Actor | Admin, Teacher |
| Description | User needs to select face detection option and system will start detecting faces from webcam. |
| Pre-Conditions | User logins into the system. |
| Post-Conditions | Face detected. |

Table 5.11 Detect Face UCD

#### Manual Attendance

|  |  |
| --- | --- |
| Manual Attendance | |
| Element | **Details** |
| Actor | Admin |
| Description | Admin can mark attendance manually upon request from a teacher. |
| Pre-Conditions | User logins into the system. |
| Post-Conditions | Attendance marked manually. |

Table 5.12 Manual Attendance UCD

#### Logout

|  |  |
| --- | --- |
| Logout | |
| Element | **Details** |
| Actor | Admin, Teacher |
| Description | User can logout of the system. |
| Pre-Conditions | User logins into the system. |
| Post-Conditions | User logged out of the system. |

Table 5.13 Logout UCD

## Data Flow Diagrams

Diagram, schematic

Description automatically generated

Figure 5.2 DFD: Context Level

Schematic

Description automatically generated

Figure 5.3 DFD: Level 1 (1.0 Administrator Activities)

Diagram

Description automatically generated

Figure 5.4 DFD: Level 1 (2.1 Teacher Activities)

Diagram

Description automatically generated

Figure 5.5 DFD: Level 1 (1.0 Student Registration)

Diagram, schematic

Description automatically generated

Figure 5.6 DFD Level 0

## Process Flowchart

Diagram

Description automatically generated

Figure 5.7 Process Flowchart

## Architectural Design

The system architecture consists of three layers, application, system, and database layer.

Diagram

Description automatically generated

Figure 5.8 Architectural Design

#### Application Layer

This is the capturing phase where the system is used to capture frames using a web application and store files on the database. This layer is secure, where authentication is provided to prevent unauthorised access to the system. Web application would be used to capture images and view the attendance reports.

#### System Layer

System layer is where detection and recognition modules are done. Viola-Jones algorithm is used to detect faces from the frames. First, integral image is generated, which assigns numbers to pixels by finding the total of the values. Furthermore, haar-like feature is generated to detect objects from the frames. Images which are extracted are sent to a classifier that is used to detect faces from the objects. LBHP algorithm is used for recognizing the images.

#### Database Layer

Centralized system is the database layer where the students’ records and attendance are stored. During attendance, student’s captured image is compared to one stored on database. Once recognition is completed successfully, attendance is recorded and updated in the database. Attendance report is available on the web application at any time.

## Sitemap

Graphical user interface

Description automatically generated with medium confidence

Figure 5.9 Sitemap

## User Interface

Graphical user interface

Description automatically generated

Figure 5.10 Homepage

Graphical user interface, application

Description automatically generated

Figure 5.11 Registration Page

Graphical user interface

Description automatically generated

Figure 5.12 Login Page

Graphical user interface, text, application

Description automatically generated

Figure 5.13 Menu Page

A screenshot of a person

Description automatically generated with low confidence

Figure 5.14 Student Attendance Record

Table

Description automatically generated

Figure 5.15 Attendance Report

## Activity Diagram

Diagram

Description automatically generated

Figure 5.16 Activity Diagram

## Database Design

#### Entity Relationship Diagram

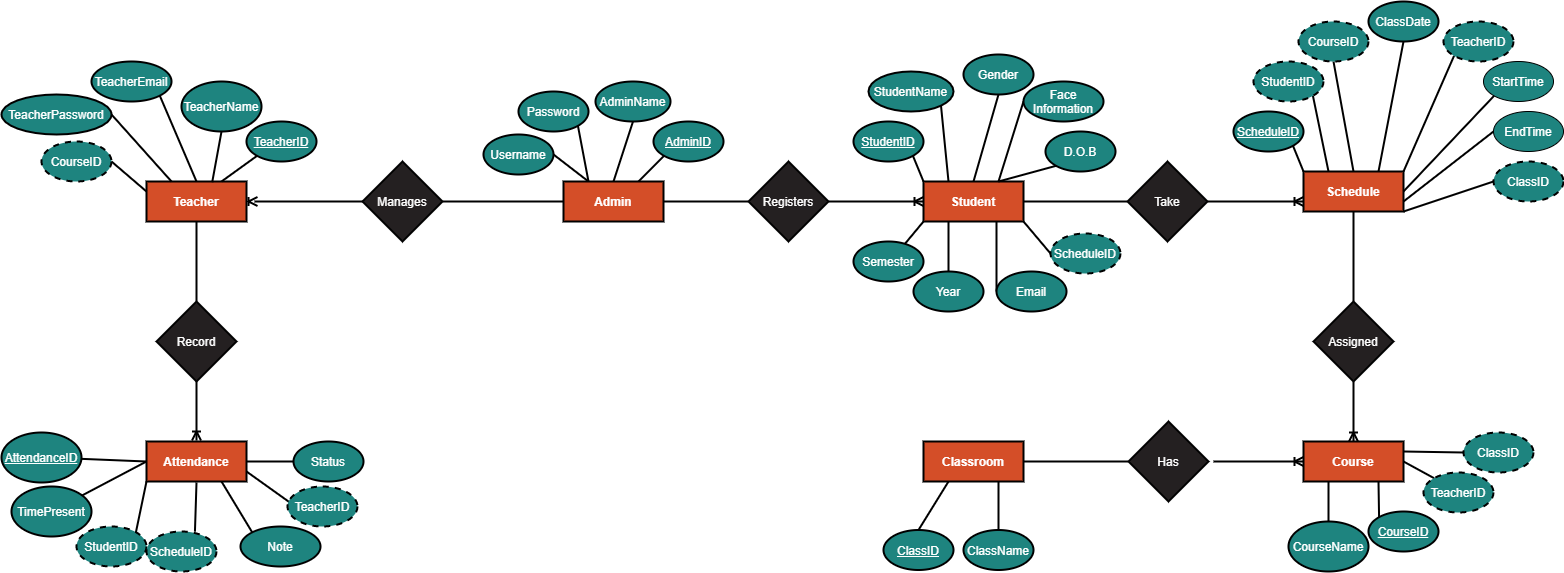


Figure 5.17 ERD Diagram

#### Relational Database

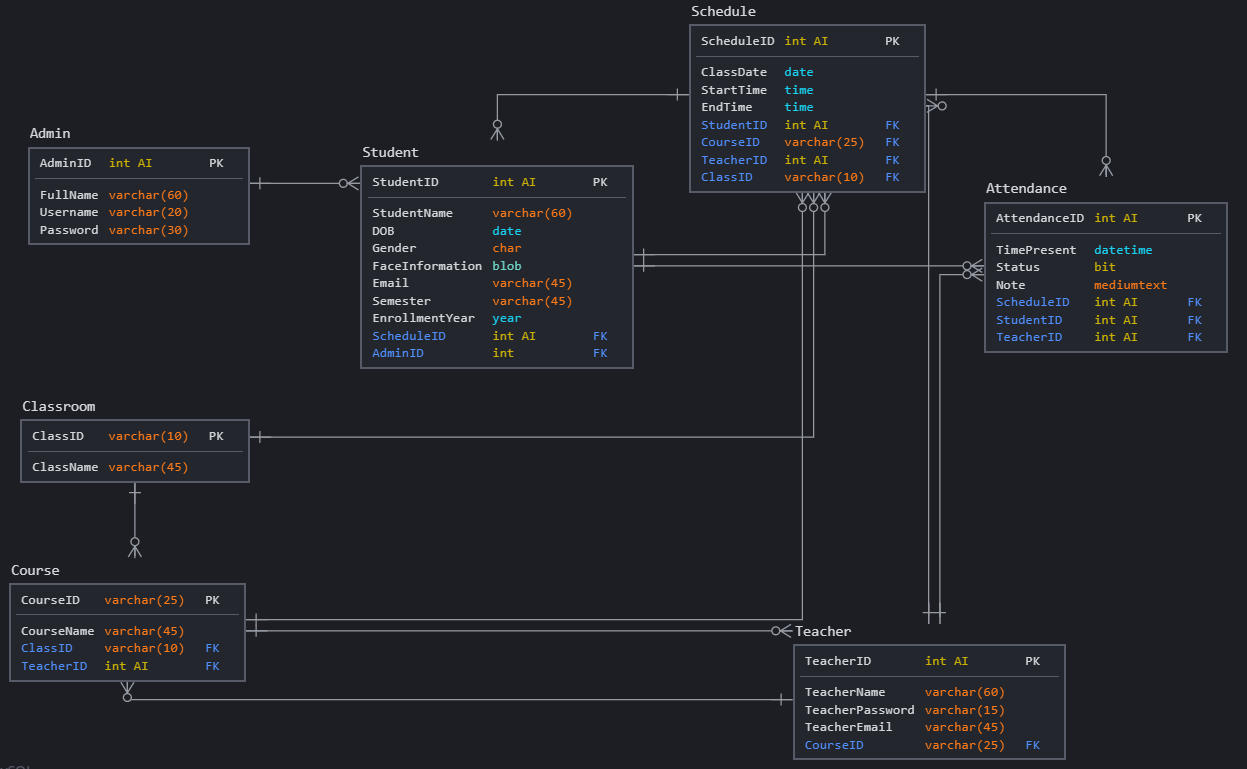


Figure 5.18 Relational Database

## UML Class

Timeline

Description automatically generated

Figure 5.19 UML Class

## Registration Process

Graphical user interface, application

Description automatically generated

Figure 5.20 Registration Process

## Attendance Process

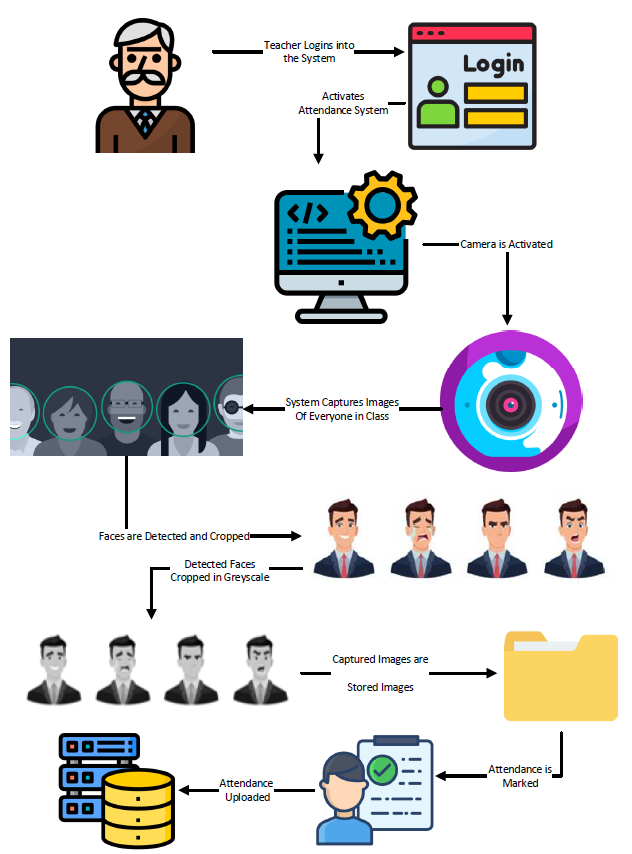


Figure 5.21 Attendance Process

# System Implementation

The proposed system has been implemented in Python programming language with the help of flask we can run Python script on webpage. Flask helped us host local webserver and provide communication with HTML pages.

## Module Description

The features of AFRAS are divided into two modules:

#### Manage Attendance

This module manages student’s attendance. Using this, teachers can mark student’s attendance. Admin can download view how many students have attended.

#### Manage Students

This feature deals with features related to student’s profile. With this feature admin can add student’s photo to the system and train it in order for facial recognition to work.

## Implementation Requirements

This section will detail how the project was implemented. We'll discuss the technology we used, give code examples and descriptions, as well as discuss how the project evolved.

## Technology Used

The project initially started on PyCharm, but we later switched to Visual Studio Code due to the simpler layout, we used Python to write the code, first initializing a virtual environment using the virtualenv package to make sure the project and its dependencies were isolated, than installing all the required packages for the project.

The project has two essential packages: OpenCV and face-recognition. OpenCV is a computer vision package we used to capture the video or images and perform detection on them, and display it back to the user. face-recognition is the package used to perform the face detection and face recognition.

At first the project was done completely as a python project, then we used another package, Flask, to display it as a webpage. Flask is a web framework API that separates the program methods into routes accessible through a URL, which allows the program to display and be interacted with through a web browser.

We had intentions of publishing the program to a web hosting service, either Heroku or Google App Engine, unfortunately, these web services cannot use OpenCV and do not allow us to easily capture the user's webcam, so the program will run locally in its present iteration.

## Implementation of Face Encoding

To use face recognition, the program must have data on the faces it already knows, in this case, the student images, the program can detect all the student images in the specified folder then return their encodings, a 128-dimensional array carrying all the relevant data of the face.

Most examples of the face-recognition library encode images for comparison every time the program runs, but the encoding process is slow, so in our implementation, we decided to perform the face encodings in a separate method and save the encodings values to the file, so that we only need to run it when adding or removing new students. Since the encodings are returned in an array, they can't be simply saved in a .txt file, so we used Python's pickle library to save and load the array.

We also have an uploader method which takes an image uploaded to the website, records the image name, and saves the image with the correct name in the /Students folder.

## Code Details of Face Encoding

**We first have to point the program to where the student images are stored, as well as extract the names of the students from the images, this is the method we use:**

1. pathlib.Path('./Students').mkdir(exist\_ok=True)
2. def StudentImagesPath():
3. global StudentImages, StudentNames
4. path = './Students'
5. StudentImages = []
6. StudentNames = []
7. StudentList = os.listdir(path)
8. for Student in StudentList:
9. Img = cv2.imread(f'{path}/{Student}')
10. StudentImages.append(Img)
11. StudentNames.append(os.path.splitext(Student)[0]) # Get just the names, ignore the .jpg

In this method we scan the /Students folder for images (To avoid errors, we create the folder if it doesn't already exist), store the images in StudentImages, and store the names in StudentNames, we also make these variables global so that they may be read from the other methods in our program where they are needed.

**The method we use for encoding the faces from the StudentImages is as follows:**

1. def encode(images):
2. enlist=[]
3. with open('encodings', 'wb') as file:
4. for img in images:
5. img = cv2.cvtColor(img, cv2.COLOR\_BGR2RGB) # Convert image to RGB
6. en = face\_recognition.face\_encodings(img)[0] # Encode the first face in the image
7. enlist.append(en) # Append the face data to the list
8. pickle.dump(enlist, file) # Save list to file
9. return(len(enlist))

We call this method using encode(StudentImages), passing the StudentImages to it. First we create an empty array called enlist to store the face encodings, than we open (or create if it doesn't exist) a file called encodings, we use openCV to convert the image from BGR to RGB so that face-recognition could recognize it, than we call the face\_recognition.face\_encodings(img)[0] method, this will save the face encodings of the face in the index 0 (The first face detected) of the image in variable en, than append the variable to the enlist array. We use pickle.dump to save the array into a file.

**This is the method we use to add student images to the program:**

1. UPLOAD\_FOLDER = './Students'
2. ALLOWED\_EXTENSIONS = {'jpg'}
4. def allowed\_file(filename):
5. return '.' in filename and \
6. filename.rsplit('.', 1)[1].lower() in ALLOWED\_EXTENSIONS
8. def uploader():
9. pic = request.files['pic']
10. if pic and allowed\_file(pic.filename):
11. filename = secure\_filename(pic.filename)
12. filename=filename.replace("\_", " ")
13. pic.save(os.path.join(app.config['UPLOAD\_FOLDER'], filename))
14. filename=os.path.splitext(filename)[0]
15. StudentImagesPath()

The first 2 lines specify the path the images should be stored, as well as the allowable file types respectively, in this case jpg files. In the uploader method, we use request.files to take the images from the user, and use secure\_filename to get the name of the image without allowing for the possibility of the image name to execute as code. By default, if you upload a file called "John Doe.jpg" it will be titled "John\_Doe.jpg" by secure-filename, so we use filename.replace to replace the underscore with a proper space. We then store the image in the specified folder and use split text to take the filename without the extension. Since a new student has been added, we need to run the StudentImagePath method again to update the lists with all the students.

## Implementation of Face Detection and Recognition

We must first capture a webcam stream with OpenCV using cv2.VideoCapture(0). The face-recognition library comes with a method called face\_recognition.face\_locations(image), we pass a frame of the video stream to this method and it "Returns an array of bounding boxes of human faces in a image", we then use another method face\_recognition.face\_encodings(Image, FaceLocation), we pass the image and the location of every face in that image and it returns a 128-dimensional array of the faces in each location of each image passed to it.

The next step is to perform the comparison between our captured frame and the students we have registered. For every face encoding we capture, we compare it to the encodings found in our encodings file, and we get the face\_distance which tells us how similar the faces are, if the face\_distance is sufficiently low, we count that as a match and add the name associated with the face to a list which we pass to our Attendance method to store the list in a .csv file.

The face recognition algorithm is unacceptably slow with our hardware, so we implemented two changes to improve performance: The first step in using the algorithm is shrinking the images using OpenCV, the images are shrunk down to 1/4th of their original size which improves the performance of the algorithm by 4 times while still maintaining acceptable accuracy, the second change is running the algorithm only every 5 frames, which improved the performance of the algorithm another 5 times, leading to a total speed increase of 20 times while maintaining almost the same level of accuracy.

**Code details of face detection and recognition:**

1. This is the method we use for video capture, face detection, and recognition:
3. def Capture():
4. StudentImagesPath()
5. if os.path.exists("Attendance.csv"):
6. os.remove("Attendance.csv")
7. cap = cv2.VideoCapture(0) # Video or image source, 0 for webcam stream,
8. # "file path" to open image or video files
10. # Open previously saved face encodings
11. if (os.path.exists('encodings')):
12. with open('encodings', 'rb') as file:
13. encodings = pickle.load(file)
14. else: return 'Error, encodings file does not exist'
15. f=0
16. namelist=[]
17. while True:
18. success, img = cap.read()
19. k = cv2.waitKey(1) & 0xFF
20. if k == ord('q'):
21. break
22. if img is None:
23. break
24. else:
25. imgS = cv2.resize(img, (0, 0), None, fx=0.25, fy=0.25) # Shrink image to improve performance
26. # Run face recognition every n'th frame to improve performance
27. if f%5==0:
28. # Run face rec on shrunk image
29. FacesCurFrame = face\_recognition.face\_locations(imgS)
30. EncodesCurFrame = face\_recognition.face\_encodings(imgS, FacesCurFrame)
31. # Find the encoded image with the smallest distance to the image in current frame
32. for EncodeFace, FaceLoc in zip(EncodesCurFrame, FacesCurFrame):
33. matches = face\_recognition.compare\_faces(encodings, EncodeFace)
34. FaceDis = face\_recognition.face\_distance(encodings, EncodeFace)
35. matchIndex = numpy.argmin(FaceDis)
36. if matches[matchIndex]:
37. name = StudentNames[matchIndex]
38. namelist.append(name)
39. cv2.imshow('Vid', img)
40. cv2.waitKey(1)
41. f+=1
42. cap.release()
43. cv2.destroyAllWindows()
44. nameset=set(namelist)
45. for name in nameset:
46. attendance(name)

To account for the fact that users might run face detection without performing uploading or encoding first, we run the StudentImagesPath method again, we then remove the previous Attendance.csv file if it exists so that we can create a new one.

We use the cv2.VideoCapture method to capture the video stream and store it in the variable cap. We again use pickle to open the encodings file if it exists (And return an error if it doesn't), and store the values in the encodings variable.

We use while True to continuously execute a loop, and store the image from cap into img, the following few lines of code break the loop when either the button 'q' is pressed on the keyboard or when the image doesn't exist. We use cv2.resize to shrink the image to 25% of its original size, and using f%5==0 and incrementing f by 1 every loop iteration, we run the face detection and recognition on the shrunk image every 5 frames. First we take the locations of all faces in the frame using face\_locations, than feed them to face\_encodings to get the encodings of all faces in frame. We use face-recognition's compare\_faces and face\_distance to get the difference between the faces in frame and each face in our Students folder, using numpy.argmin we write the smallest face distance to matchIndex, and append the name matching that face to namelist.

Assuming that a person stands infront of the camera, each frame where the algorithm detects a match, their name will be added to namelist, to avoid repetition, we convert namelist to a set called nameset, then for every name in that nameset we run it through the attendance method:

1. def attendance(name):
2. with open('Attendance.csv','a+') as file:
3. file.writelines(f'{name}\n')

This method opens (Or creates if it doesn't exist) a file called Attendance.csv in append mode and uses file.writelines to append the name followed by a line.

The last method we have in the code allows the user to download this attendance file:

1. def downloadFile ():
2. path = "./Attendance.csv"
3. return send\_file(path, as\_attachment=True)

It simply sends the Attendance.csv file as an attachment so that the name remains the same.

## Implementation of Flask

Originally, the program was just as it was described above, separated into several methods that the user can call by inputting certain integers on the terminal. To transform the program into a webpage, we had to use the flask package. Flask is a web framework API, it separates the program into routes, each with a method, accessible by a certain URL. Flask allows you to separate each method into a route, and to call that method, you simply use the URL of the main program followed by the route.

**Code details of Flask:**

First, we must initialize the main flask app and choose where the program starts:

1. app = Flask(\_\_name\_\_)
3. @app.route('/')
4. def start():
5. return render\_template('start.html')

The app route '/' indicates the main page of the program, which is accessible by going to the browser and typing "localhost:5000", which leads to the start() method, so when we initially run the app, it will call the start method. This method simply renders the start.html page found in the templates folder.

The start.html file can contain anything you want, and runs like any other html file when you start the program, but to get the ability to access different methods in the Python code, we must use routes. A simple example in our Python code is the download method, it's written like this:

1. @app.route('/download')
2. def downloadFile ():
3. path = "./Attendance.csv"
4. return send\_file(path, as\_attachment=True)

The @app.route in the beginning indicates that this method is routed through the URL localhost:5000/download, once you run flask, you can simply type that address into the URL bar to download the attendance file, or, we can add a link to that URL in the start.html page like so:

1. <a href="/download">Download file</a>

Our start page will now have a single link, and clicking that link will route to /download which in turn calls the downloadFile method.

Using flask you can link to several methods from the start page, and end each method with a render\_template, allowing the program to run completely through a web interface using simple links.

## Implementation of Front End

HyperText Markup Language was used to define and structure the website. To develop the UI and UX of website, Bootstrap, CSS and JavaScript styling were used. These assets were stored in individual static folder. We used cascading style sheets to describe presentation of website written in HTML. Bootstrap is front end framework which simplifies the process of adding styles, by providing built-in CSS classes used for HTML elements. This saved us time from writing complicated CSS. JavaScript allowed us to add dynamic content to our website such as animation.

## Structure of GUI

After three months, we were successfully able to build an Automated Facial Recognition Attendance System. The following pages demonstrate some of the key functionalities of AFRAS.



Figure 6.1 AFRAS Dashboard

Fig 6.1 illustrates Application GUI developed with Bootstrap, HTML, JavaScript and CSS. This would be used to interact with the system. It has 6 buttons to leverage features of AFRAS.

After successful activation, admin would be presented with functionality whether he wants to register new students by clicking Upload, which will upload their images to training dataset or start capturing process to record current students. Once photos have been added, admin would need to train the model which consists of iterating through all images, extracting essential facial features and landmarks, saving them into classifier. Training has been labelled as Encode.

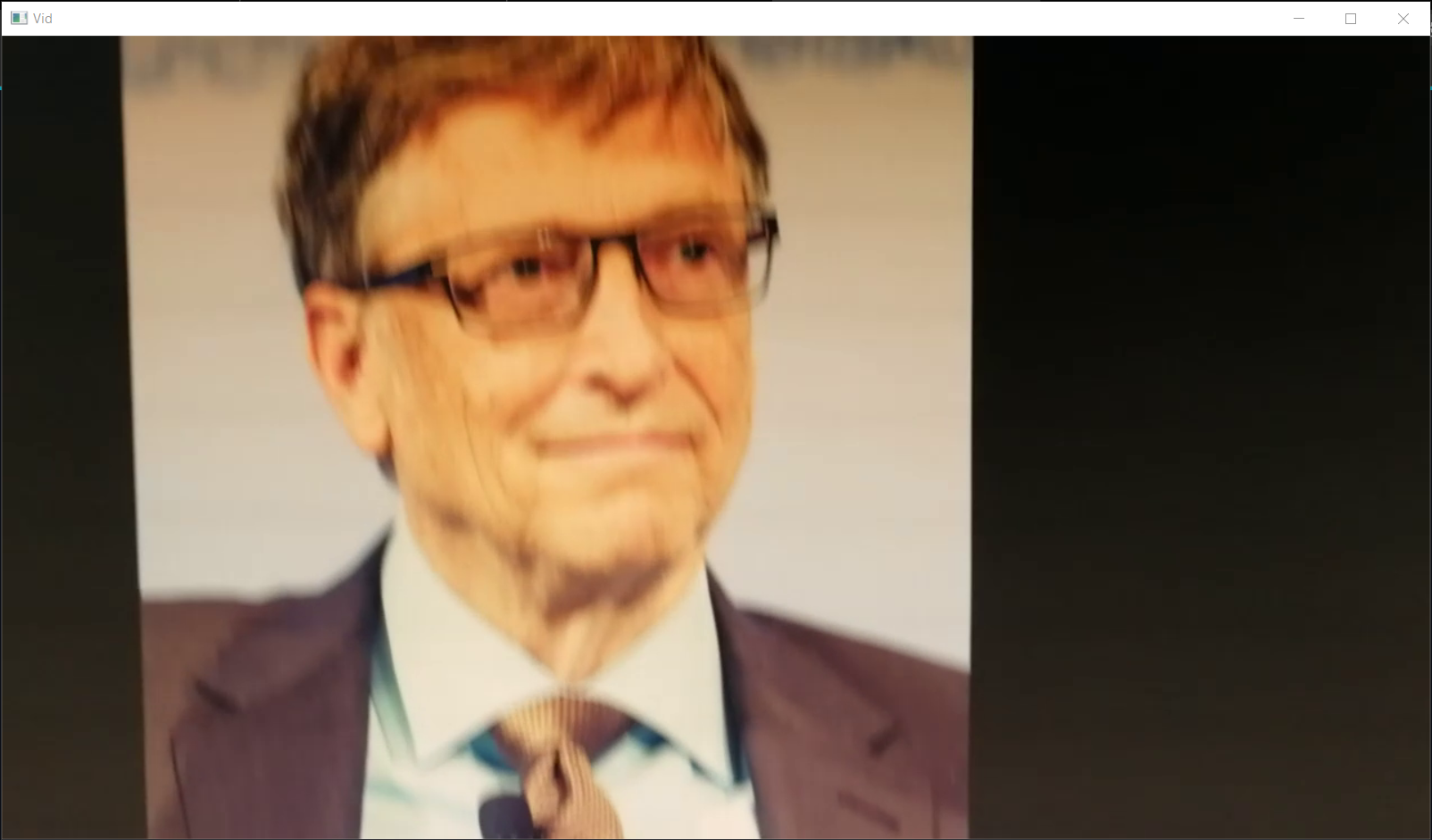


Figure 6.2 System in real time marking attendance

Once training is done, you can begin attendance process and this is how it occurs. Python program would be opened on your system and it would start detecting and recognising faces in real-time shown in fig 6.2.



Figure 6.3 Showing Recorded Students

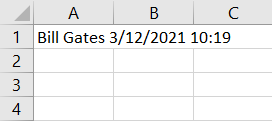


Figure 6.4 Attendance Report

# System Testing

Testing is done to find out how well system works. In hardware and software development, testing is used in key areas whether system achieves its objective. There are different types of testing listed below [24].

## Acceptance Testing

Acceptance testing is done by client to ensure delivered product meets the requirements and works as expected. It falls under black box testing.

## Beta Testing

Beta testing also known as user testing is performed by team outside development, it is not full version of the application. We performed beta-testing by giving beta-version of our application to other Computer Science students to validate its usability, compatibility, functionality and fix any unexpected errors.

## Code Review

We did a systemic review of our code to remove any vulnerabilities that existed in the code such as buffer overflow.

## Functional Testing

We used this testing technique to test features of the system. We did black box testing, where we tested the system from external user’s perspective. Thomas Hamilton describes black box testing as “It is a Software Testing method that analyzes the functionality of a software without knowing much about the internal structure/design of the item that is being tested and compares the input value with the output value” [25].

## Integration Testing

Integration testing is combining group of elements to produce an output. Often done between hardware and software to find out if they have any relation. In our case we tested if webcam works with our code.

## Performance Testing

Performance testing was done to evaluate speed and effectiveness of the system and to ensure it generates output within time span mentioned in performance requirements. This falls under black box testing.

## Regression Testing

Regression testing was performed to re-execute codes which were impacted by changes. We often did regression tests while working on the system to ensure any new changes have not affected any parts of application. Later at end, we will perform final regression test to validate the build before submitting it.

## Software Testing

Software testing is process of evaluation to detect differences between input and expected output. We did it to assess quality of our system.

## Stress Testing

We did non-functional stress testing to see how system would behave in unfavourable conditions. This type of testing is conducted outside the specification. It falls under black box testing. During this test, system was monitored, after overloading it to ensure it can sustain in stressful conditions.

## System Testing

In system testing we put our system in different environments like OSes (Windows 10 and macOS) and web browsers (Chrome, Firefox and Edge). System testing was done with complete implementation of the system and environment. This falls under black box testing.

## Unit Testing

Testing individual unit or group of related units. We did this to ensure implemented code produces expected output.

## Usability Testing

Usability testing is done in end-users’ perspective, it was difficult measure this but we evaluated it using these parameters: how user-friendly GUI is; level of skill required to run software for novice and expert users; how pleasing design is. This falls under black box testing.

## Validation

Validation ensures product satisfies specified requirements at end of development. In laymen terms, the product is built as client requirements.

## Verification

Verification was done to ensure system satisfies conditions which were mentioned in start of design phase. In laymen terms, the product behaves as we want it to or are we building the right product.

## Test Environment

|  |  |  |
| --- | --- | --- |
| **No.** | **Resource Type** | **Details** |
| **1** | OS Name | Windows 10 Home, 64 bits |
| **2** | CPU | Intel Core i7-4720HQ @2.60 GHz |
| **3** | RAM | 16 GB |
| **4** | Storage | 515 GB |
| **5** | Webcam Resolution | HP Truevision Full HD 640x480 |
| **6** | Source Code Editor | Visual Studio Code |
| **7** | Browser | Google Chrome |

Table 7.1 Test Environment

## Test Cases

|  |  |
| --- | --- |
| **ID** | **TC01** |
| **Test Name** | Web cam integration and configuration |
| **Pre-Requisite** | Web cam is connected and integrated with the system |
| **Test Action** | 1. Run the system 2. Encode 3. Click on capture to begin camera |
| **Expected Result** | System automatically turns on web cam in web browser |
| **Status** | Pass |

Table 7.2 Test Case 01

|  |  |
| --- | --- |
| **ID** | **TC02** |
| **Test Name** | Face detection and storage in training set |
| **Pre-Requisite** | Detected faces captured, cropped and stored in training set |
| **Test Action** | 1. Click on Encode 2. Capture |
| **Expected Result** | By clicking the capture button, python window opens which starts detecting faces, detected faces are cropped and set in training set folder, after detection ends, website leads to new page with all students who attended and downloadable excel file is created with names of students who attended. |
| **Status** | Pass |

Table 7.3 Test Case 02

|  |  |
| --- | --- |
| **ID** | **TC03** |
| **Test Name** | Face recognition and generation of excel attendance sheet |
| **Pre-Requisite** | Images of students on system are compared with detected faces and according to that attendance sheet is created |
| **Test Action** | 1. Encode the system 2. Click on Capture 3. Click on Download |
| **Expected Result** | System training completes and camera starts detecting faces and compares with stored dataset. Recognized faces are marked present and their names are recorded in excel sheet. |
| **Status** | Pass |

Table 7.4 Test Case 03

|  |  |
| --- | --- |
| **ID** | **TC04** |
| **Test Name** | Upload student image |
| **Pre-Requisite** | File should be in image format e.g., PNG, JPEG |
| **Test Action** | 1. Run the system 2. Click Upload 3. Type Student Name 4. Upload Image |
| **Expected Result** | Image is saved in students folder with name |
| **Status** | Pass |

Table 7.5 Test Case 04

|  |  |
| --- | --- |
| **ID** | **TC05** |
| **Test Name** | Face detection and recognition in unsuitable environments |
| **Pre-Requisite** | Face is detected in real-time, recognized and attendance is marked |
| **Test Action** | 1. Encode the system 2. Click on Capture |
| **Expected Result** | System encode completes and camera starts detecting faces and compares with stored dataset. Recognized faces are marked present and their names are recorded in excel sheet. |
| **Status** | Fail |
| **Reason** | Faces fail to detect due to bad lighting conditions |

Table 7.6 Test Case 05

|  |  |
| --- | --- |
| **ID** | **TC06** |
| **Test Name** | Dashboard links |
| **Pre-Requisite** | User login |
| **Test Action** | 1. Run the system 2. View Dashboard |
| **Expected Result** | All links work and dashboard is fully functional |
| **Status** | Pass |

Table 7.7 Test Case 06

## Testing Result (Output)

These are the results of tests we conducted in real-time. Below are all the screenshots of different functions.

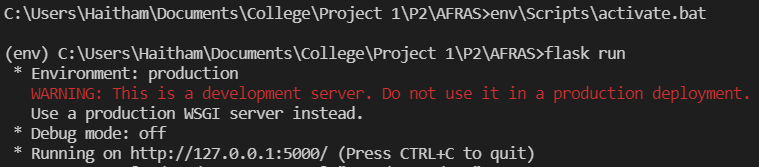


Figure 7.1 Successfully Ran the System

At first to begin the system we need to run it using “flask run” command.



Figure 7.2 Fully Functional Dashboard

After application has activated, user is presented with GUI to make the system easy-to-use. In the GUI, we have included following features:

* Start Page to restart the system
* Capture button to begin attendance process
* Encode to train the images
* Upload to add new student images in training dataset
* Download to get attendance report
* Contact in case of any enquires etc.



Figure 7.3 Encoding Complete



Figure 7.4 Camera Turns on and Detects Faces Successfully

In proper lighting conditions, students face would be efficiently captured.



Figure 7.5 Recognition Completed and Marked Successfully

Images are cropped and sent to facial recognition algorithm. If person is present than his attendance is marked, like here we showed only Elon Musk and Bill Gates images to the webcam hence these two were recognised and marked present.

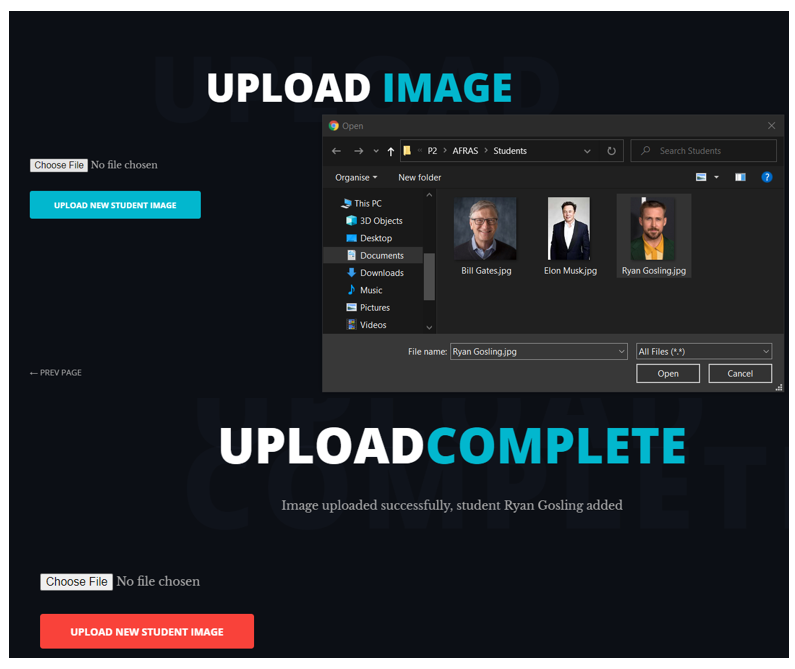


Figure 7.6 Upload Successful

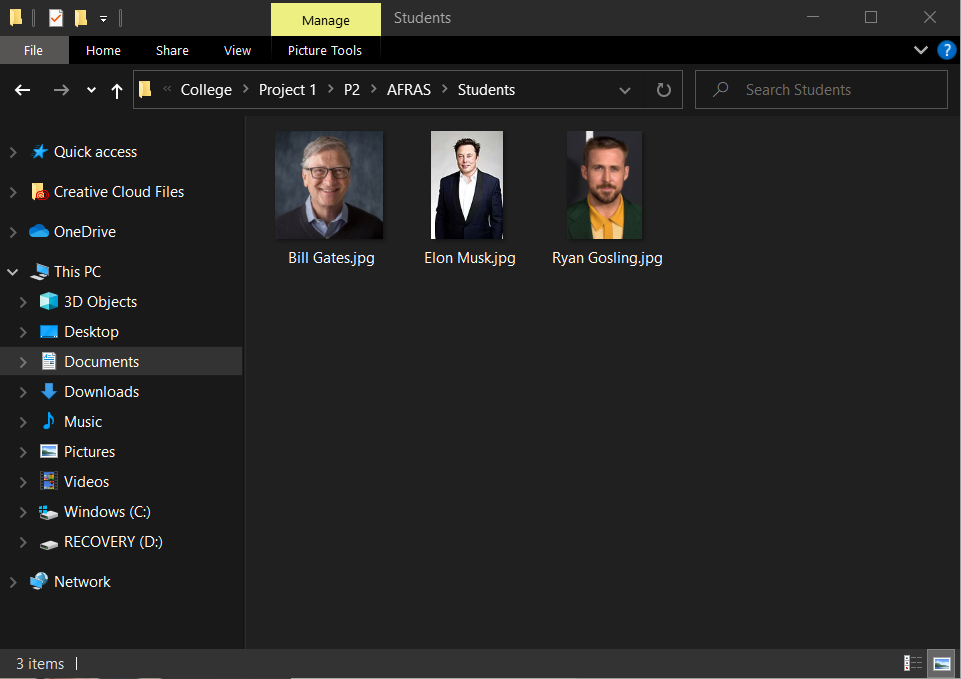


Figure 7.7 Images in Dataset

All enrolled student images are stored in /Student folder. For more accuracy we can increase the number of training images but that would compromise speed of our system.

## Deployment of System

When functional and non-functional requirements are met, project would be completed and submitted.

## Maintenance

If there are any issues or improvements during feedback, they would be added to the system to further enhance it.

# Benefits of Afras

AFRAS has various benefits compared to traditional attendance systems listed below.

## Easy to Install

AFRAS is versatile and able to program in any computer or environment.

## Eliminates Manual Attendance

Firstly, it eliminates the pen and paper method of attendance taking. Secondly, electronic medium of data storage provides better maintenance of records.

## Eliminates Proxy Attendance

Attendance is taken in real-time with student present in class therefore it completes eradicates “buddy punching” students.

## High Success Rate

Facial recognition technology has high success rate and fast. It can match multiple users at once.

## Saves Money

Traditional methods use papers and files to record and store whereas some automated such as RFID use cards. Facial recognition system eliminates all these objects required to record attendance. Since the system is automated, it eliminates human error. AFRAS promises good value with enormous return and saving.

## Saves Time

Traditional systems have lecturers calling each student’s name with respect to their ID which is time consuming and automated attendance often are slow such as RFID and biometric scanners where multiple students want to scan their cards or finger, they have to do it one at a time. AFRAS has recognize and mark attendance of multiple students at a time.

## Secure

We haven’t used any external database or server. Our database is self-generated on computer; hence it has hierarchical architecture of folders to save data from website. We have made it secure my adding admin rights and securing folders with passwords. Therefore, any unauthorised users won’t be able to access any data or make changes.

## Contact Less System

In post-pandemic world, it’s better to reduce all types of physical contact in public places like educational institutes. Today, there is a demand for contactless technologies. Therefore, AFRAS can be a great choice for people who want to reduce frequency of contact with individuals at workplace and minimize virus transmission.

# Limitations

There are few limitations with this system. Firstly, the input image has to be frontal and upright. Secondly, under extreme illumination conditions, the accuracy of system will drop. Thirdly, if images stored are blurred than there are chances of false recognition. Finally, if a student applies makeup in an image than face recognition will be affected as important features would be covered.

We can eliminate illumination issue by using a good camera and start attendance process in better lighting conditions. In AFRAS, the laptop built-in camera would be the default device. However, lighting source of camera varies from each device hence causes the system to be unstable. In future, we would use an external web cam to obtain better results.

The classifier takes time in training each image hence for large number of students, it would be time consuming. Although training the classifier only needs to be done when new student is added or when you start the system initially, it would be convenient if classifier takes lesser time in training while still maintaining accuracy of the system.

# Conclusion

This system has been proposed to replace existing systems such as RFID, fingerprint scanning, manual attendance taking etc. Biometric systems can only check one student at a time, and now, during coronavirus, it is not wise to use one due to the potential spread of germs. Manual attendance systems are time consuming and confusing in large classrooms, therefore, it is important to implement a face recognition attendance system.

Also, during the period of COVID-19, medical institutes and government authorities are urging people to follow social distancing. Therefore, with implementation of such systems social distancing can be followed easily. This system can also be modified and used in public places as surveillance. In the next phase we would be finding out systems and design requirements for this system.

# Future Enhancement

AFRAS is able achieve the task of marking attendance through facial recognition automatically and attendance report is exported in an excel sheet in real-time. Although, to implement a system in educational institute, a dedicated system has to be built which is insensitive to lighting conditions. We can also create an online database of attendance of all students and automatically update their attendance. These enhancements can improve AFRAS drastically.

Additionally, mail server can be set up to automatically send an email to absentees’ respective parents instead of messaging each parent individually.

# References

|  |  |
| --- | --- |
| [1] | Chao and Wei-Lun, “Face Recognition,” *GCIE, National Taiwan University.* |
| [2] | S.Aanjanadevi, Dr.V.Palanisamy and R.AnandhaJothi, “A Study on Secure Online Voting System using Biometrics,” *.International Journal for Modern Trends in Science and Technology,* p. V3(8), 2017. |
| [3] | N. M. Ara, N. S. Simul and M. Islam, “Convolutional neural network approach for vision based student recognition system,” in *International Conference of Computer and Information Technology (ICCIT)*, 2017. |
| [4] | X. Liu and W. Wang, “User Centered Design of Attendance Record System Based on Mobile Terminals,” *International Symposium on Computer Science and Intelligent Controls,* pp. 141-146, 2017. |
| [5] | S. Chintalapati and M. Raghunadh, “Automated Attendance Management System Based On Face Recognition Algorithms,” *IEEE International Conference on Computational Intelligence and Computing Research,* pp. pp. 1-5, 2013. |
| [6] | Y. KAWAGUCHI, T. SHOJI, W. LIN, K. KAKUSHO and M. MINOH, “Face Recognition-based Lecture Attendance System,” *Academic Center for Computing and Media Studies, Kyoto University,* 2005. |
| [7] | R. Satpute, S. Sontakke, B. Gondaliya, T. Sonawane and K. Suryawanshi, “ATTENDANCE MANAGEMENT SYSTEM USING FACE RECOGNITION,” *International Research Journal of Engineering and Technology (IRJET),* vol. Volume 7, no. Issue 05 , 2020. |
| [8] | A. Katara, M. V. Kolhe, M. A. P. Zilpe, M. N. D. Bhele and M. J. Bele, “Attendance System Using Face Recognition and Class Monitoring System,” *International Journal on Recent and Innovation Trends in Computing and Communication,* vol. 5, 2017. |
| [9] | P. Viola and M. J. Jones, “Robust Real-Time Face Detection,” *International Journal of Computer Vision,* pp. 137-154, 2001. |
| [10] | N. K. Balcoh, M. H. Yousaf, W. Ahmad and M. I. Baig, “Algorithm for Efficient Attendance Management: Face Recognition based approach,” *International Journal of Computer Science Issues,* 2012. |
| [11] | T. H. K. S. Theo Ephraim, “Real-Time Viola-Jones Face Detection in a Web Browser,” *Canadian Conference on Computer and Robot Vision,* 2009. |
| [12] | R. M and Dr.Manish, “A comparative study on Convolutional Neural Network and Viola-Jones Algrithm for faster Face Recognition,” *International Research Journal of Engineering and Technology (IRJET),* vol. 7, no. 7, 2020. |
| [13] | H. Mujtaba, “Face Detection using Viola Jones Algorithm,” 2 September 2020. [Online]. Available: https://www.mygreatlearning.com/blog/viola-jones-algorithm/. [Accessed 24 05 2021]. |
| [14] | Behera and G. Shankar, “Face Detection with Haar Cascade,” 24 December 2020. [Online]. Available: https://towardsdatascience.com/face-detection-with-haar-cascade-727f68dafd08. [Accessed 29 May 2021]. |
| [15] | S. Lee, “Understanding Face Detection with the Viola-Jones Object Detection Framework,” 22 May 2020. [Online]. Available: https://towardsdatascience.com/understanding-face-detection-with-the-viola-jones-object-detection-framework-c55cc2a9da14. [Accessed 21 June 2021]. |
| [16] | S. Lee, “Understanding Face Detection with the Viola-Jones Object Detection Framework,” 20 May 2020. [Online]. Available: https://towardsdatascience.com/understanding-face-detection-with-the-viola-jones-object-detection-framework-c55cc2a9da14. [Accessed 25 May 2021]. |
| [17] | R. Gupta, “Breaking Down Facial Recognition: The Viola-Jones Algorithm,” Towards Data Science, 7 August 2019. [Online]. Available: https://towardsdatascience.com/the-intuition-behind-facial-detection-the-viola-jones-algorithm-29d9106b6999. [Accessed 2020 07 18]. |
| [18] | M. Pietikäinen, “Local Binary Patterns,” 2010. [Online]. Available: http://www.scholarpedia.org/article/Local\_Binary\_Patterns. [Accessed 18 June 2021]. |
| [19] | K. S. d. Prado, “Face Recognition: Understanding LBPH Algorithm,” 11 November 2017. [Online]. Available: https://towardsdatascience.com/face-recognition-how-lbph-works-90ec258c3d6b. [Accessed 29 May 2021]. |
| [20] | T. Ahonen, A. Hadid and M. Pietikainen, “Face Description with Local Binary Patterns: Application to Face Recognition,” *IEEE,* vol. 28, no. 12, 2006. |
| [21] | Yashwant, “LBPH algorithm for Face Recognition,” [Online]. Available: https://iq.opengenus.org/lbph-algorithm-for-face-recognition/. [Accessed 22 June 2021]. |
| [22] | M. Sundaram and A. Mani, Face Recognition: Demystification of Multifarious Aspect in Evaluation Metrics, IntechOpen, 2015. |
| [23] | Reddy, N.Sudhakar, M.V.Sumanth and S. Babu, “A Counterpart Approach to Attendance and Feedback System using Machine Learning Techniques,” *Journal of Emerging Technologies and Innovative Research (JETIR),* vol. Volume 5, no. Issue 12, 2018. |
| [24] | Tutorials Point, “Software Testing Dictionary,” [Online]. Available: https://www.tutorialspoint.com/software\_testing\_dictionary/index.htm. [Accessed 3 12 2021]. |
| [25] | T. Hamilton, “What is BLACK Box Testing? Techniques, Example & Types,” 8 October 2021. [Online]. Available: https://www.guru99.com/black-box-testing.html. [Accessed 3 12 2021]. |
| [26] | Thales Group, “Face recognition – fascinating and intriguing,” 22 May 2021. [Online]. Available: https://www.thalesgroup.com/en/markets/digital-identity-and-security/government/biometrics/facial-recognition. |
| [27] | M. Chauhan and M. Sakle, “Study & Analysis of Different Face Detection Techniques,” *(IJCSIT) International Journal of Computer Science and Information Technologies,* pp. pp. 1615-1618, 2014. |
| [28] | Mishra and Kajal, “Challenges Faced by Facial Recognition System,” 18 August 2020. [Online]. Available: https://www.pathpartnertech.com/challenges-faced-by-facial-recognition-system/. |
| [29] | Reddy, N.Sudhakar, M.V.Sumanth and S. Babu, “A Counterpart Approach to Attendance and Feedback System using Machine Learning Techniques,” *International Journal of Emerging Technologies and Innovative Research (www.jetir.org),* vol. 5, no. 12, pp. 190-193, December 2018. |
| [30] | N. Krüger, “How to Write a Software Requirements Specification (SRS Document),” 23 October 2018. [Online]. Available: https://www.perforce.com/blog/alm/how-write-software-requirements-specification-srs-document. [Accessed 10 June 2021]. |