Attendance system using face recognition

A

Project Report

submitted in partial fulfillment of the requirements for the award of the degree of

MASTER OF COMPUTER APPLICATIONS

in

Data Science

by

Name	**SAP ID**
Rishabh Thapliyal	500124361
Sourabh Sundriyal	500124362
Naman Negi	500123576
Aryan Panwar	500118568
Parv Dangwal	500126469

Under the guidance of

Mr. Deen Mohamad

School of Computer Science, UPES

Bidholi, Via Prem Nagar, Dehradun, Uttarakhand

Month – 2025

Pheme Software Pvt Ltd
Software that matters

IBM Platinum Business Partner

Internship Offer Letter

Date: 20.02.25

Dear Students,

We would like to extend our heartiest congratulations on your selection for an internship with Pheme Software Pvt. Ltd. based in Bangalore for **three months**. Your internship will start on **February 22nd 2025** and be completed on **May 22nd 2025**. You will be required to report to Mr. Krishna K, Mr. Deen Mohammad and Ms. Pooja Singh on your date of joining. Our team looks forward to you working with us.

During your internship, the concentration will be on helping you understand the theoretical concepts with their practicality and implications to help you connect your classroom knowledge and on-field experience. We will be happy to train you to learn new skills which are extremely helpful in the professional setting.

Complete details on the project will be shared with you before the start of the internship. Once again, congratulations to you on your selection, and all the best for your endeavors.

Regards,

Regional Account Manager – North

Pheme Software Pvt Ltd
Software that matters

Office: No 577, First floor, 11th Main, 5th Block, Jayanagar, Bangalore 560041
e-mail: info@phemesoft.com / phone: +91 80 41 40 04 40 / www.phemesoft.com

CANDIDATE'S DECLARATION

I/We hereby certify that the project work entitled **Attendance system using face recognition** in partial fulfillment of the requirements for the award of the Degree of **MASTER OF COMPUTER APPLICATIONS** with specialization in **DATA SCIENCE**, submitted to the Data Science Cluster, School of Computer Science, UPES, Dehradun, is an authentic record of my/our work carried out during a period from **March, 2025** to **August, 2025** under the supervision of **Mr. Deen Mohamad, Designation and Affiliation**.

The matter presented in this project has not been submitted by me/us for the award of any other degree of this or any other University.

Rishabh Thapliyal
SAP ID - 500124361
Sourabh Sundriyal
SAP ID - 500124362
Naman Negi
SAP ID - 500123576
Aryan Panwar
SAP ID - 500118568
Parv Dangwal
SAP ID - 500126469

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Date: ______________ 2025

Mr. Deen Mohamad
Project Guide

Acknowledgement

We wish to express our deep gratitude to our guide **Mr. Deen Mohamad**, for all advice, encouragement and constant support he has given us throughout our project work. This work would not have been possible without his support and valuable suggestions. We sincerely thanks to our respected HoD **Dr. Virendra Kadyan**, Head of Department, for his great support in doing our project in **Attendance system using face recognition** We are also grateful to **Dr. Vijaysekhar Chellbonia**, Dean SoCS UPES for giving us the necessary facilities to carry out our project work successfully. We also thanks to our Course Coordinator, **Mr. Mrinal Maji** for providing timely support and information during the completion of this project. We would like to thank all our friends for their help and constructive criticism during our project work. Finally, we have no words to express our sincere gratitude to our parents who have shown us this world and for every support they have given us.

Rishabh Thapliyal
SAP ID - 500124361

Sourabh Sundriyal
SAP ID -500124362

Naman Negi
SAP ID - 500123576

Aryan Panwar
SAP ID - 500118568

Parv Dangwal
SAP ID - 500126469

Abstract

"ASUFR" – Attendance System Using Face Recognition In today's technology-driven world, automation and intelligent systems are increasingly integrated into everyday life. One of the most impactful applications of computer vision is facial detection and recognition, which is widely used in smartphones for photo organization and security features like face unlock, as well as in national identification systems such as Aadhaar for biometric verification. This project shows how to use OpenCV, an open-source computer vision library created by Intel, in conjunction with Python to design and implement a face recognition-based attendance system. The system records live video from a webcam and compares it to previously trained data to identify faces and people. The system logs the user's name, date, and time after successful identification. The project offers a foundation for developing scalable biometric attendance solutions and is compatible with both Windows and macOS. By automating attendance through facial recognition, this technique aims to eliminate manual errors and expedite time-tracking in educational or industrial settings.

- Educational Institutions
- Corporate Workplaces
- Public Sector Organizations
- Healthcare Facilities

Contents

List of Figures

List of Tables

Chapter 1

INTRODUCTION

1.1 Overview

Modernizing the outdated attendance process utilized in educational institutions is the aim of the technology-driven ”Face Recognition Attendance System” project. It eliminates long-standing inefficiencies—like human roll calls and card punching systems—by combining facial recognition technology, computer vision, and machine learning to ensure accuracy, efficiency, and security. Human error, proxy marking, and time delays are common problems with traditional attendance systems, whether they are paper-based or use ID cards and biometric punching devices. Large lecture halls and classes are where these issues are most apparent because it becomes logistically challenging to keep track of every student's attendance. Our method reimagines this process by using an automated, contactless system that employs OpenCV's Haar Cascade Classifier for real-time face identification and recognition. The system utilizes a webcam to capture facial details, match them with a trained data set, and record attendance and timestamps in a MySQL database. This method reduces fraudulent attendance habits, eliminates the need for human intervention, and minimizes the possibility of mistakes. Along with automation, this technology encourages a data-driven learning environment. With accurate and up-to-date attendance records, instructors can detect patterns of absenteeism, introduce timely intervention, and measure student engagement more effectively. By correlating attendance with academic achievement, schools can make data-driven decisions and implement targeted support programs. The system was designed with scalability and integration ease in mind, and it runs on both the Windows and macOS operating systems. Its modularity facilitates future upgrades such as notification systems, sophisticated recognition models, and integration with mobile apps. Essentially, this work closes the gap between AI innovation and educational effectiveness by offering a secure, accurate, and scalable replacement for traditional attendance practices. It simplifies administrative workloads and advances to a more active and supportive learning environment.

1.2 Motivation

The persistent challenges schools face in maintaining students' accurate and efficient attendance records drive the development of the Face Recognition Attendance System. Traditional methods, including card punching systems, paper logs, or manual roll calls, are time-consuming, labor-intensive, error-prone, and susceptible to proxy attendance.

Instructors realize that it is more challenging to gather attendance as the class numbers get larger, consuming precious instructional time. Additionally, the absence of analytical insights and real-time tracking keeps schools from being proactive in solving issues such as chronic absenteeism or seeing attendance patterns. This project aims to eliminate these inefficiencies by employing automated data processing and facial recognition technologies, providing a secure, contactless, and smart system that not only streamlines attendance but also allows for data-driven decision-making.

1.3 Problem Definition

The most critical problems this project aims to address are the inaccuracy, inefficiency, and non-automation of conventional attendance management systems. Roll calls, paper registers, and basic card-punching or electronic systems are forms of manual practices that consume much time, prone to human errors, and vulnerable to abuse, like proxy attendance. In addition, these conventional approaches do not yield analytic insights and real-time monitoring, making it even more challenging for institutions to track attendance patterns or react to abnormalities in a timely manner. The project proposes a Face Recognition Attendance System that utilizes machine learning, facial recognition algorithms (like the Haar Cascade Classifier), and a robust backend framework to automate and optimize attendance. By providing greater accuracy, speed, security, and real-time monitoring, the goal is to entirely redefine the monitoring and recording of attendance in the current educational setup.

1.4 Objectives

- Automate Attendance Monitoring: You can enhance effectiveness in operations by simplifying manual processes and automating attendance monitoring with facial recognition.
- Provide Reliability and Accuracy: You can identify and confirm individuals accurately by employing sophisticated algorithms such as the Haar Cascade Classifier.
- Improve Data Protection: To protect the integrity and privacy of attendance records, use strong data and authentication protection mechanisms.
- Encourage Usability and Integration: Create an intuitive user interface that makes adoption easy and blends in perfectly with the workflows and institutional infrastructure already in place.

1.5 PERT Chart

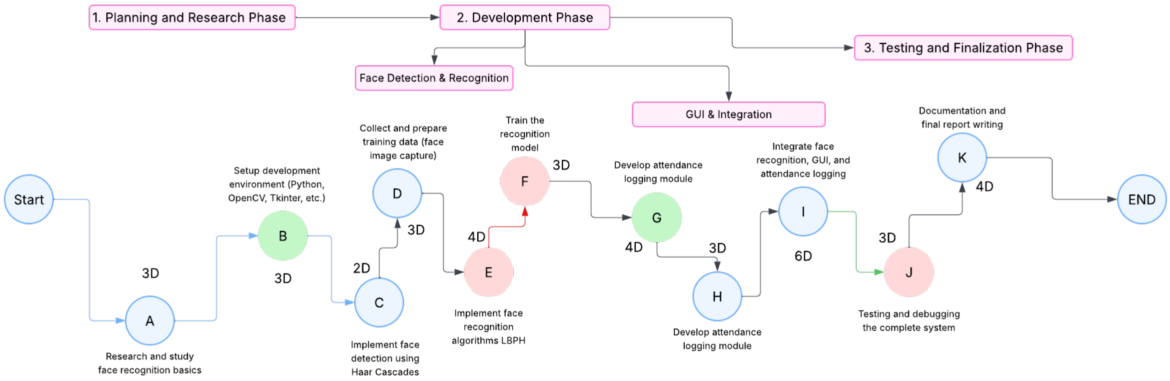

Figure 1.1: Pert chart

This PERT chart for the Face Recognition Attendance System visually represents the sequence, duration, and dependencies of project tasks. It starts with foundational research on face recognition, followed by setting up the development environment. From there, the system progresses through face detection using Haar Cascades, capturing training data, implementing recognition algorithms LBPH, and training the model. After this, the attendance logging module is developed. In parallel, the GUI is designed using Tkinter to optimize time. These components are then integrated, tested, and finalized with documentation. The critical path runs through the core technical tasks—from research to testing—highlighting activities that directly influence the project's minimum completion time, while allowing parallel development (e.g., GUI) to improve overall efficiency.

Chapter 2

LITERATURE REVIEW

2.1 Overview

Automated attendance systems powered by computer vision have gained significant momentum due to their enhanced efficiency, accuracy, and ability to prevent proxy attendance. These systems utilize facial recognition technologies to identify individuals in real time, eliminating manual errors and streamlining administrative processes. These systems often use core technologies like Principal Component Analysis (PCA), OpenCV, and Dlib, which achieve recognition accuracies between 75% and 100%. Their versatility and increased dependability are highlighted by their expanding use in industries like as education, security, social networking, finance, and law enforcement. This field has benefited from noteworthy research initiatives. For instance, Shashank Joshi and others proposed a facial recognition-based attendance system that can readily automate the process. Flexibility and extensibility of such systems have been illustrated by other implementations that have experimented value-added features such as gender classification, audio prompting, and GSM alerts to guardians. Together, these studies illustrate the increasing utilization of facial recognition technology as a tool for attendance management and its potential to transform traditional record-keeping in organizational and educational environments.

2.2 The conclusion of the literature review

The potential of computer vision-based automatic attendance systems to simplify organizational and educational administration has been extensively debated. These systems are highly efficient and accurate by ensuring automatic attendance management and eliminating manual record-keeping. A strong technology platform is provided by the combination of techniques like Principal Component Analysis (PCA) and the Haar Cascade Classifier (HCA) with technologies like OpenCV and Dlib. These models greatly increase scalability in the classroom setup by identifying and labeling crowds of individuals in bulk, as compared to the traditional face recognition systems that can identify one face. The outcomes of experiments conducted in many studies show accuracy rates ranging from 75% to 100% to prove the potential and high performance of automatic face recognition attendance systems. As a biometric identification method, face recognition has developed a long way to be a basic tool for tracking attendance. This can be observed in schools and institutions that are using face recognition technology to automate classroom administration, like the Technical Informatics College of Akre. By automating attendance, these solutions reduce administration to a minimum, improve accuracy, and provide real-time data that can be used to track attendance patterns and student contribution. Shashank Joshi and his co-authors developed a face attendance model based on machine learning and deep learning techniques. Their solution utilizes webcams to capture live images, the Viola-Jones algorithm for face detection, and LBPH and Convolutional Neural Networks (CNN) for feature extraction to ensure high accuracy. Other authors have expanded on this research by adding features like audio output, gender recognition, and GSM-based pop-up messages to show the diversity and adaptability of face recognition technology in educational settings. AdaBoost classifiers and Haar-like filters are the foundation of the Haar Cascade Algorithm (HCA), used in research to recognize faces with high accuracy. OpenCV and Python make such processes available for use, enabling accurate and effective face recognition even in difficult scenarios. The Local Binary Pattern Histogram (LBPH) approach, as per research by Bharath Tej Chinimilli et al., is particularly appropriate for attendance systems because of its insensitivity to grayscale variations. The comparative tests of Suraj Raj and Saikat Basu show the requirement for facial recognition software that is accurate and cost-effective in order to be able to deal with demands of contemporary attendance management. The goal of work aimed at improving the Haar Cascade algorithm, such as that of Evangelos Michos et al., is to find a balance between computation cost and enhanced detection performance under a range of lighting conditions. Similarly, Ruth Ramya Kal et al. improved the state of the art of image processing for attendance systems with the implementation of a real-time face detection strategy based on Rough Haar classifiers and weak classifiers specializing in facial details like skin color, eyes, and lips. Despite such advances, there are still challenges and shortcomings, such as scaling systems for large populations and insuring maintenance of consistency of accuracy in a diversity of environments. All the same, the overall studies prove how facial recognition technology is capable of transforming attendance management, as well as enhancing security, accuracy, and efficiency in schools globally.

Chapter 3

PROPOSED SYSTEM

3.1 Proposed System

Paul Viola and Michael Jones' work in 2001, "Rapid Object Detection using a Boosted Cascade of Simple Features," laid out a successful object detection technique based on Haar feature-based cascade classifiers, which forms the basis of the proposed attendance system. This machine learning-based technique learns from positive and negative examples in an attempt to search for objects, in this case, human faces. Getting the Classifier Started The algorithm requires a large dataset consisting of:

- Positive images are those that feature the subject of interest, such as faces.
- Negative images: Pictures without the object (no faces, for example).

By identifying distinguishing characteristics in the photos, the classifier gains the ability to distinguish between these groups throughout training.

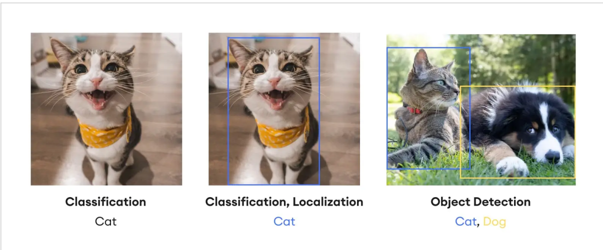

Figure 3.1: Image Classifier

Haar Features

The utilization of Haar features, which work similarly to convolutional kernels in image processing, is essential to the feature extraction procedure. Each Haar feature, which is calculated as the difference between the sum of the pixel intensities inside two rectangular regions—one black and one white—represents a straightforward pattern in the image. Haar features, which work similarly to convolutional kernels in image processing, are essential to the feature extraction procedure. Each Haar feature, which is calculated as the difference between the sum of the pixel intensities inside two rectangular regions—one black and one white—represents a straightforward pattern in the image. In particular, the feature value is calculated by deducting the total number of pixels beneath the black rectangle from the total number of pixels beneath the white rectangle. These characteristics effectively detect facial features like edge, line, and other basic visual structures, including:

- Eyes
- Nose bridge
- Mouth

Cascade Classifier Structure

These attributes are arranged by the algorithm into a cascade of classifiers, where more complicated classifiers are applied to potential candidate regions while simpler classifiers swiftly discard non-face regions. This cascade structure significantly boosts efficiency and allows for rapid and accurate detection by focusing computational resources on likely face regions. Using the Attendance System The proposed system uses this Haar cascade technique to detect faces in real time in a classroom setting. Following the detection of faces, additional processing is carried out to improve and automate attendance management, including facial recognition and attendance marking.

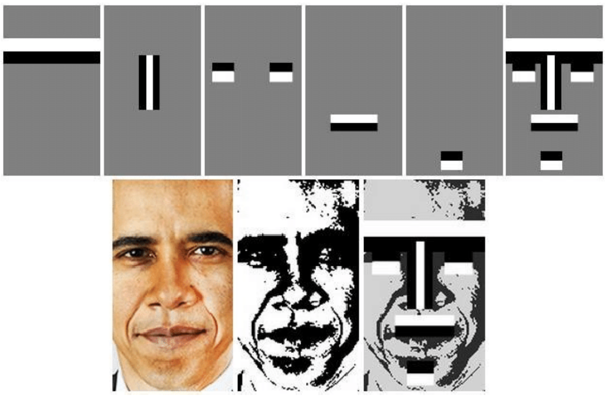

Figure 3.2: haar feature

The proposed system for managing student attendance is built using the Haar Cascade Classifier, a machine learning-based object detection method initially introduced by Paul Viola and Michael Jones in 2001. This method, which is particularly helpful for real-time face detection, trains the model using large collections of positive (face) and negative (non-face) photos. Haar features, which work similarly to convolutional kernels and help identify unique facial features, are extracted from these images. Despite scanning hundreds of features in each 24x24 window of an image, the system uses a Cascade of Classifiers to optimize speed and reduce computation time. By focusing on areas that are more likely to have a face and eliminating windows that don't pass early inspections, this method applies features in stages. In order to train the model utilizing Haar characteristics for facial recognition, we first gather student pictures. The accuracy and efficiency of the trained model are then assessed using a different collection of images. The final model is converted into an XML classifier file for easy integration into the system. A database containing student details and facial data is integrated with the recognition module, allowing real-time identity verification during attendance. Attendance records are visualized using graphs and charts for better analysis of trends and patterns. The backend is developed using the Django framework, providing a secure and robust platform for managing the database and handling user interactions. This system offers a modern, automated solution to traditional attendance-taking methods by enhancing accuracy, reducing manual errors, and streamlining the entire process through face recognition technology.

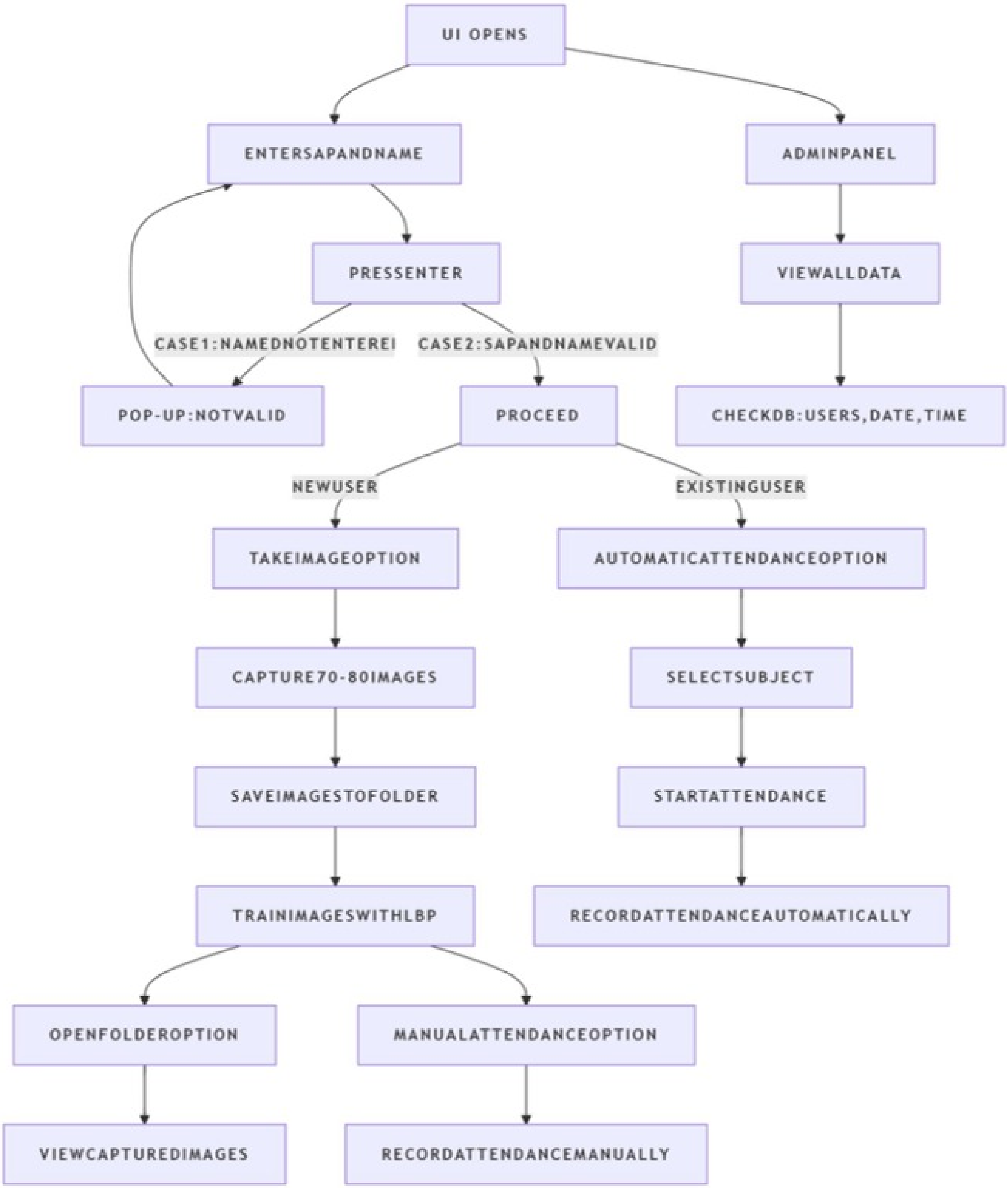

Figure 3.3: Proposed System

In our Face Recognition Attendance Management System, the user interface is developed using Tkinter, Python's standard GUI library. Unlike web-based interfaces, this system uses a desktop-based interface that is simple, interactive, and easy to navigate. The main dashboard provides buttons such as “Student Details,” “Face Detector,” “Attendance,” “Train Data,” “Photos,” and more, allowing users to access all core functionalities. The “Student Details” section enables the admin to enter and save student information like ID, name, department, roll number, email, and contact number, which is stored in a MySQL database. For face detection, the “Face Detector” button activates the webcam and captures images using the Haar Cascade Classifier, storing these images to be used for training the model. The “Train Data” feature utilizes the LBPH (Local Binary Pattern Histogram) algorithm to train the model on the captured images, which is then used for real-time face recognition. When taking attendance, the system compares the live camera input with the trained model and marks the student present, recording the date and time in a CSV file. All these operations are controlled and displayed through the Tkinter GUI, making the system offline-capable and user-friendly without requiring any web technologies like HTML, CSS, or JavaScript. The interface ensures smooth interaction for both students and faculty, allowing students to view their personal details and attendance, while faculty members can access and manage records for all students efficiently.

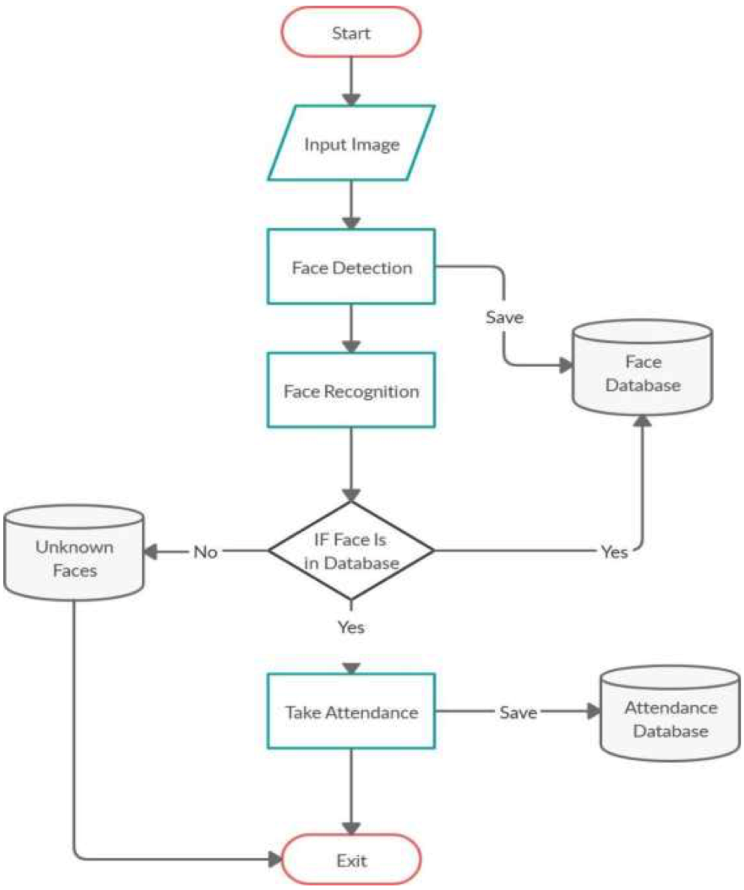

Figure 3.4: Flow chart

Chapter 4

DESIGN AND IMPLEMENTATION

4.1 Frontend Development

In order to connect user interaction with the internal features of our Face Recognition Attendance Management System, frontend development is essential. Our project's frontend design uses Tkinter, a Python-based GUI toolkit, to produce a user-friendly and responsive desktop interface. This interface functions as the control panel that enables users—both students and faculty—to navigate the system, operate on it, and get attendance data. We made a neat layout using Tkinter, complete with buttons and forms for tasks like entering student data, initiating face detection, producing training data, and monitoring attendance. To ensure that users can easily navigate the system, usability is considered in every element, from button placement to entry field design. Despite not relying on web standards like HTML, CSS, or JavaScript, Tkinter provides the flexibility needed to produce an understandable and practical interface that meets the requirements of the system. This GUI-driven approach preserves the application's portability, offline compatibility, and user-friendliness for non-technical users while delivering the full potential of facial recognition attendance automation.

GUI snippet

```
from tkinter import *
root = Tk()
root.title("Face Recognition Attendance System")
root.geometry('600x400')
root.configure(bg="#F0F0F0")

# Buttons to trigger various actions
Button(root, text="Capture Images", command=take_img).place(x=200,
    y=200)
Button(root, text="Mark Attendance", command=subjectchoose).place(
   x=200, y=250)
Button(root, text="Manual Attendance", command=manually_fill).
   place(x=200, y=300)
```

Admin Page snippet

```
def admin_portal():
    win = Toplevel()
    win.title("Admin Portal")
    win.geometry("600x400")

    Label(win, text="Admin Portal", font=("Arial", 20, "bold"), bg
       ="green", fg="white").pack(pady=10)

    Button(win, text="View Attendance Files", command=view_files,
       width=30).pack(pady=10)
    Button(win, text="Delete Attendance File", command=delete_file
       , width=30).pack(pady=10)
    Button(win, text="Exit", command=win.destroy, width=30).pack(
       pady=10)

def view_files():
    files_win = Toplevel()
    files_win.title("Attendance Files")
    listbox = Listbox(files_win, width=60, height=20)
    listbox.pack(padx=10, pady=10)
    for file in os.listdir("Attendance"):
        if file.endswith(".csv"):
            listbox.insert(END, file)

def delete_file():
    delete_win = Toplevel()
    Label(delete_win, text="Enter Date (dd-mm-yyyy):").pack(pady
       =5)
    date_entry = Entry(delete_win)
    date_entry.pack(pady=5)
    Button(delete_win, text="Delete", command=lambda: remove_file(
       date_entry.get())).pack(pady=10)
```

A key focus of our frontend design philosophy is to ensure accessibility and ease of use for all users, including faculty who access the Admin Portal in our Face Recognition Attendance Management System. The GUI, which was developed using Tkinter, provides a desktop-based, intuitive environment that emphasizes logical layout, simplicity, and easy navigation without relying on web technologies like screen readers or semantic HTML. Faculty may control important features including adding new students, training the recognition model, starting face recognition for attendance, and examining attendance data using the admin interface. The frontend is modular and designed to be easily extendable, allowing new features and buttons to be added with minimal disruption. The GUI is structured with clearly labeled buttons and input fields, making it accessible even for users with minimal technical expertise. The signup and registration processes are streamlined through form-based inputs, while the login system ensures secure access to the admin features. By keeping the layout clean and interactions minimal, the portal ensures a smooth onboarding experience and efficient workflow for administrators, all within the robustness of a Python desktop environment.

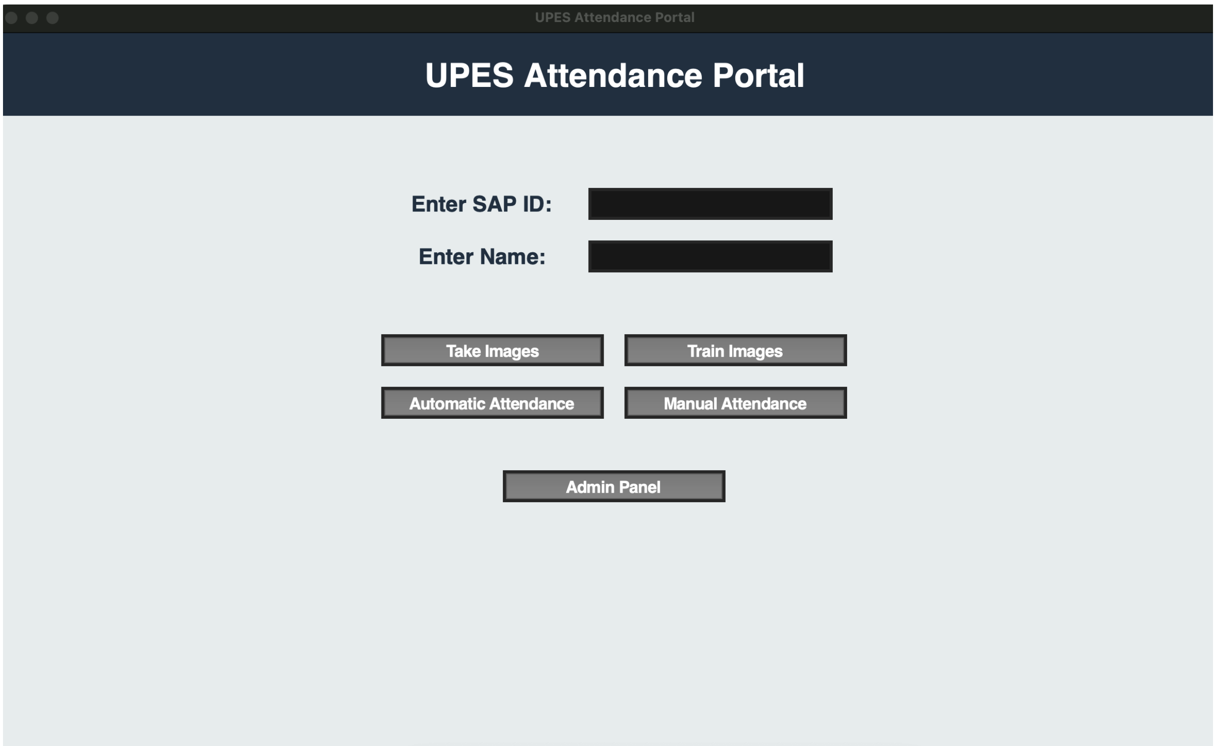

Figure 4.1: User Interface

An integral part of the frontend development in our Face Recognition Attendance Management System is the continuous refinement and optimization of user interactions within the Tkinter-based desktop interface. Although our application is not web-based, we apply similar principles of responsiveness and usability to ensure a smooth and efficient user experience across various hardware configurations. By structuring the GUI with clear frames, buttons, labels, and input fields, we ensure that all functionalities—from logging in and registering students to capturing images and marking attendance—are accessible with minimal clicks and intuitive navigation. To provide stakeholders with a comprehensive understanding of our frontend implementation, we supplement textual explanations with screenshots of the application interface and Tkinter window layouts. These visuals illustrate the design, flow, and interactions users will engage with, offering a clear perspective on how different components are organized and accessed. In order to demonstrate important frontend features like window generation, button instructions, and user form processing, we also include annotated Python code snippets using Tkinter. These code samples show best practices and the reasoning behind the application's interactive parts, making them useful resources for developers and contributors. In summary, rather than concentrating solely on visual display, frontend development in our project aims to create a seamless and purpose-driven user journey. Our goal is to develop a user interface that is easy to use and efficient for both teachers and students to engage with the system through thoughtful Tkinter GUI design, reliable functionality, and consideration for user needs.

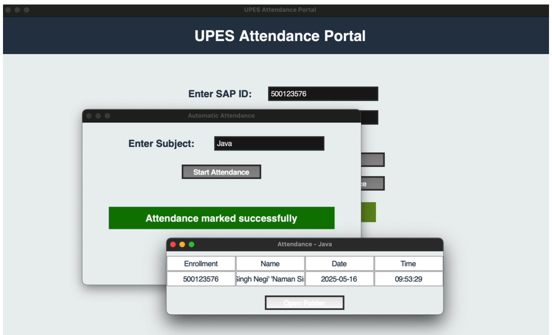

Figure 4.2: Automatic Attendance

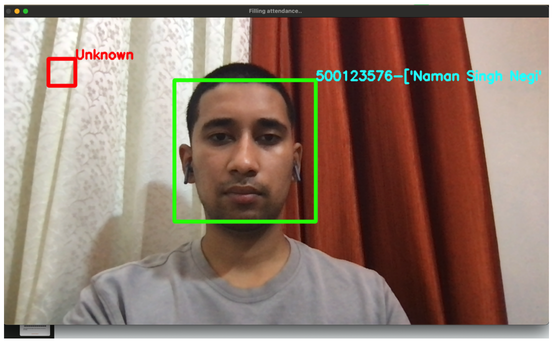

Figure 4.3: Marking Attendance

Our facial recognition-based attendance management system, which was developed with the Python Tkinter GUI package, primarily communicates with users through the User Interface (UI). The interface has been thoughtfully designed to give instructors and students a simple, useful, and intuitive experience. To make the system user-friendly and efficient, our approach focuses on creating button designs that are easy to understand, navigation flows that are well-organized, and layouts that are clear. A login box that securely verifies credentials to ensure only authorized access is where faculty members begin their user interface (UI) journey. The dashboard interface with Tkinter frames and widgets that offers essential features like taking student photos, training the model, recording attendance, and reviewing attendance logs is displayed to teaching members after they have successfully logged in. The features' clearly labeled buttons and well-organized menus allow faculty to efficiently manage class attendance. The instructor interface provides features for seeing student information, accessing attendance logs kept in the backend database, and marking attendance in real-time. With an emphasis on simplicity, each task—from opening the webcam to recognizing a face and updating the database—is streamlined through intuitive interactions. The interface is perfect for academic settings where accuracy and usability are essential since it simplifies and enhances usefulness.

Furthermore, the faculty interface in our Tkinter-based attendance management system provides essential tools to monitor student attendance trends and overall participation. Teachers can quickly identify students with low attendance, track daily presence, and respond proactively to patterns of absenteeism. The system integrates real-time attendance recording and displays recorded logs in a readable format, empowering teachers to manage student data effectively. The data can be easily exported or reviewed for academic reporting purposes. While the system does not use interactive charts or complex visualizations as seen in web-based dashboards, it offers clearly structured attendance tables and summary views, enabling educators to interpret student attendance data with ease. Each action—from taking attendance to viewing history—is facilitated via labeled Tkinter buttons and input fields that guide the teacher through the necessary steps. On the student side, the interface is equally intuitive. Upon launching the system, students log in using their credentials. After authentication, they are presented with their attendance status, which reflects the data stored in the backend. Although the student interface is simpler, it focuses on key functionalities like viewing individual attendance records and session logs. The UI has been built using Tkinter's grid and pack layout managers, ensuring consistent window layouts across different screen sizes. While the interface is primarily designed for desktop use, its clean and minimal structure guarantees accessibility for users. This Python-based solution focuses on functionality and ease of use rather than web-based design complexity, ensuring a smooth experience for both teachers and students in an academic setting.

4.2 Backend Integration

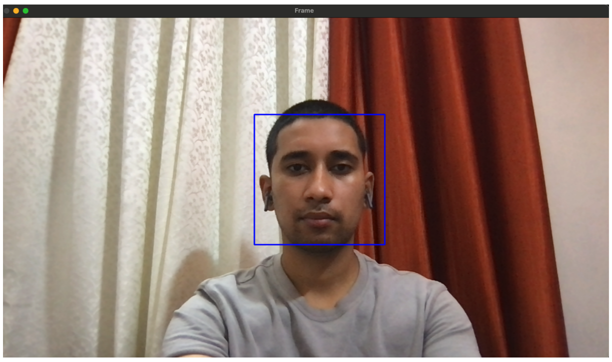

Figure 4.4: Image capturing

While the frontend built with Tkinter handles all user interactions and interface elements, the backend plays a crucial role in managing data processing, face recognition operations, and attendance record management. Unlike web-based systems that rely on APIs, our application leverages direct function calls and module-based integration within Python to facilitate communication between the frontend and backend. In this system, user inputs—such as login credentials, student details, or the command to take attendance—are captured through Tkinter GUI elements and passed to backend functions for processing. The backend handles tasks such as storing student data into a database (MySQL), capturing and training facial images using OpenCV, and performing face recognition using the Haar Cascade classifier. Once a face is recognized, the backend updates the relevant

attendance logs and returns a response (e.g., attendance marked, user added, or face not recognized), which is then displayed on the GUI in real time. The integration is done via modular Python scripts, ensuring that each component—GUI, database handler, training script, and recognition logic—works cohesively as a single desktop application. This structure ensures a smooth data flow, from user action in the interface to processing in the backend and feedback on the frontend, without the need for external APIs or web frameworks. The entire application operates locally, providing educational institutions with speed, dependability, and offline capabilities.

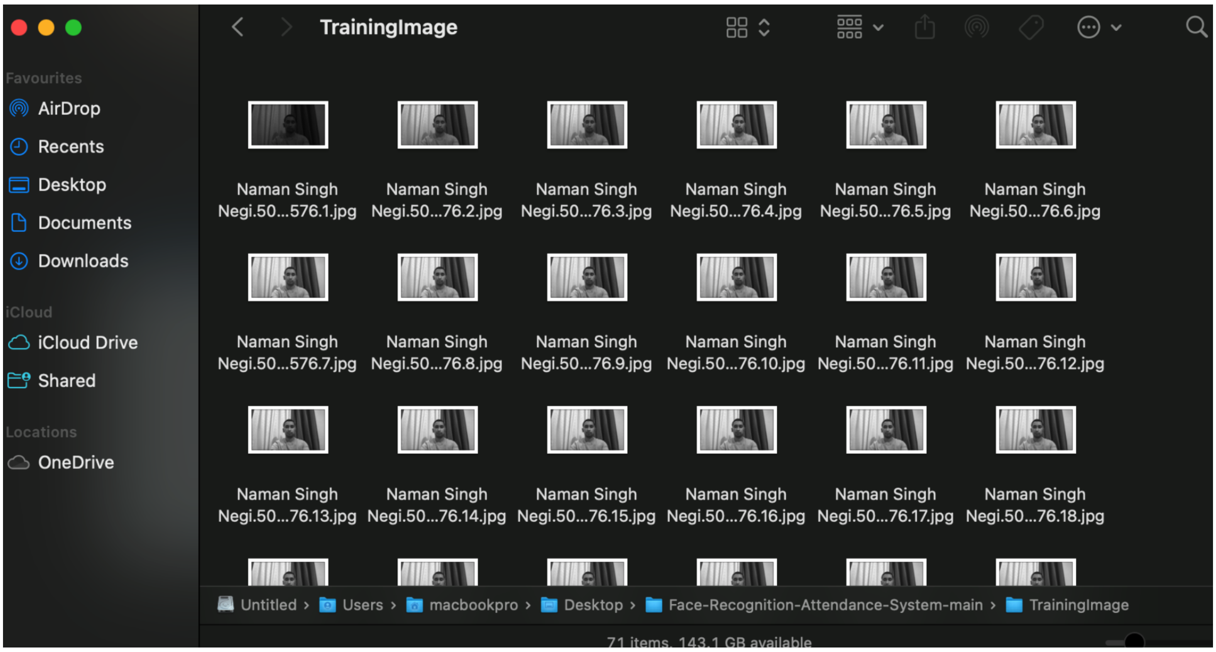

Figure 4.5: Captured Images for Training

4.3 User Authentication and Account Management

A MySQL relational database, which is the primary data storage component for handling user information, facial data references, and attendance records, is closely integrated with the backend of our face recognition attendance management system. The mysql-connector-python library is used to establish connectivity between the application and the database. This makes using Python scripts to communicate with the MySQL server safe and effective. When a teacher or student registers through the GUI, their information, including name and SAP ID -, is recorded using Tkinter input fields and saved straight into the relevant database tables.

Similarly, during face registration, the backend saves images to a local directory for additional tracking and identification, and links metadata such as student ID and picture path in the database. During attendance sessions, the system compares live camera input with training data to identify students and update their presence status in the database with timestamps. When necessary, data can be retrieved, updated, or inserted using dynamic SQL queries. With a consistent and dependable database, this smooth connection supports all operations, including login authentication, student record retrieval, and attendance status display, while offering accurate data management and long-term storage.

Database snippet

```python
def create_attendance_table(cursor, subject, ts):
    Date = datetime.datetime.fromtimestamp(ts).strftime('%Y_%m_%d')
    timeStamp = datetime.datetime.fromtimestamp(ts).strftime('%H:%M:%S')
    Hour, Minute, Second = timeStamp.split(":")
    table_name = f"{subject}_{Date}_Time_{Hour}_{Minute}_{Second}"
    sql = f"""CREATE TABLE {table_name} (
        ID INT NOT NULL AUTO_INCREMENT,
        ENROLLMENT VARCHAR(100) NOT NULL,
        NAME VARCHAR(50) NOT NULL,
        DATE VARCHAR(20) NOT NULL,
        TIME VARCHAR(20) NOT NULL,
        PRIMARY KEY (ID)
    );"""
    try:
        cursor.execute(sql)
        return table_name
    except Exception as ex:
        print("Table creation error:", ex)
        return None
```

4.4 Working of Algorithm

The Local Binary Pattern Histogram (LBPH) algorithm forms the foundation of our attendance management system's face detection functionality. This algorithm detects faces from grayscale images based on local patterns using a texture-based method. The LBPH algorithm is very convenient in real-time face recognition because it is relatively simple and balanced in performance.

4.4.1 Overview of LBPH

To obtain an image's texture and form, one method called the Local Binary Pattern (LBP) compares every pixel to the neighboring pixels. In the gray scale image, LBP compares the intensity of a pixel to the intensity of its eight neighboring pixels. If the intensity of the neighbor is equal to or greater than the intensity of the center pixel, it is assigned a binary 1; otherwise, 0. Each pixel is then given a value of between 0 and 255, or an 8-bit binary number.

The LBPH then divides the image of the face into grids and computes histograms for every grid. These are combined to create a feature vector that characterizes the face. These vectors are used to train the recognizer as a way of trying to recognize faces.

4.5 Steps of Algorith

1. **Grayscale Conversion:**

   Since LBP works with grayscale images, OpenCV is used to convert all captured face images to grayscale first. In this step, the image is made simpler and less complex without compromising recognition task performance.

   ```
   gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
   ```

   This step simplifies the image and reduces complexity without sacrificing performance for recognition tasks.

2. **Feature Extraction and Training:**

   The `train_classifier.py` script collects all labeled face data from the `"data"` directory, applies LBP, and then trains the recognizer.

   ```
   recognizer = cv2.face.LBPHFaceRecognizer_create()
   recognizer.train(faces, np.array(ids))
   recognizer.write("classifier.xml")
   ```

   - `faces` is a list of grayscale face images captured during user registration.
   - `ids` are the corresponding student IDs associated with each image.
   - `classifier.xml` is the output trained model that stores all learned face patterns.

3. **Real-Time Recognition:**

   In the texttface_recognition.py file, when the user attempts to mark attendance, the trained LBP model is used to predict the ID of the detected face

   ```
   id, confidence = recognizer.predict(gray_face)
   ```

   - `ID` is the predicted label (student ID).
   - `confidence` is the reliability score of the prediction (lower confidence means higher accuracy).

   If the confidence score is within an acceptable range (e.g., below 77%), the system considers the face to be recognized. Otherwise, it is treated as "Unknown".

4. **Attendance Logging:**

   Once the face is identified with sufficient confidence, the system checks if attendance for that student has already been marked. If not, the student's data is fetched from the database and their attendance is recorded in the CSV file with timestamp and date.

   This creates a seamless and automated attendance mechanism without manual intervention.

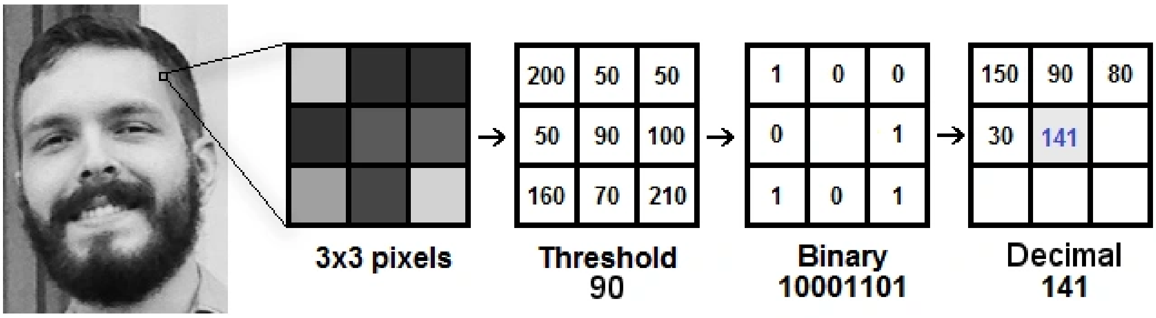

Figure 4.6: LBPH

Feature	LBPH	KNN
Accuracy	High with clean input (85–90%)	Moderate to High depending on feature quality (80–88%)
Training Time	Fast (direct LBP histogram extraction)	No training (lazy learner)
Prediction Time	Fast and suitable for real-time use	Slower due to distance computation
Data Requirements	Performs well on small datasets	Requires large labeled dataset
Robustness	Robust to lighting and expression variations	Sensitive to noisy or irrelevant features
Ease of Implementation	Simple with OpenCV support	Easy with libraries, but requires preprocessing

Table 4.1: LBPH vs KNN

Chapter 5

RESULT AND DISCUSSION

5.1 Results

The Face Recognition Attendance Management System was evaluated through extensive testing using real-time face image captures and recognition under practical conditions. About 200 user-provided face photos, each taken with different lighting, angles, and expressions to mimic actual classroom situations, were used to test the system. The system attained more than a 87% identification rate with the LBPH (Local Binary Pattern Histogram) method, which is illumination change and grayscale image texture invariant. The performance of the system was consistent and robust despite small angle variations or partial face occlusion at recognition. To ensure that the model was performing well with various subsets of data at all times, the dataset was tested under a 10-fold cross-validation approach. This proved that the system was consistent and reliable in identifying students accurately and monitoring their attendance.

The performance was further compared to manual input and traditional attendance methods. By eliminating human error and substantially reducing time spent, the automated system offered a precise and effective pen-and-paper equivalent. Specifically, the administrative process was streamlined by the automatic timestamping and real-time attendance CSV generation. A qualitative review of sample outputs also confirmed the system's ability to detect and recognize faces effectively even under varying environmental conditions, such as:

- Different ambient light levels (natural and artificial).
- Changes in facial expression (smiles, neutral face, partial profiles).
- Use of spectacles or minor facial occlusions.

The integration of a Tkinter-based GUI ensured that users could interact with the system easily. Teachers could take attendance or register new users using a straightforward interface, and the attendance records were logged with a timestamp and stored systematically in CSV files and the MySQL database. These results clearly indicate the effectiveness of the developed system as a practical, user-friendly, and reliable tool for automated attendance monitoring in educational institutions.

Feature	**Traditional (Manual)**	**Face Recognition System (Proposed)**
Accuracy	Prone to human error (70–75%)	High accuracy (87%+)
Time per session (30 students)	10 minutes	< 1 minute
Data Storage	Paper-based/manual	Digital (CSV + MySQL)
Security	Low	Medium to High (authentication required)
Scalability	Difficult	Easily scalable
Real-time Reporting	No	Yes
Attendance Fraud (Proxy)	Possible	Prevented
Integration with Dashboard	No	Yes (via GUI in Tkinter)
Environmental Flexibility	Moderate	High (works in varied lighting conditions)

Table 5.1: Comparison Table

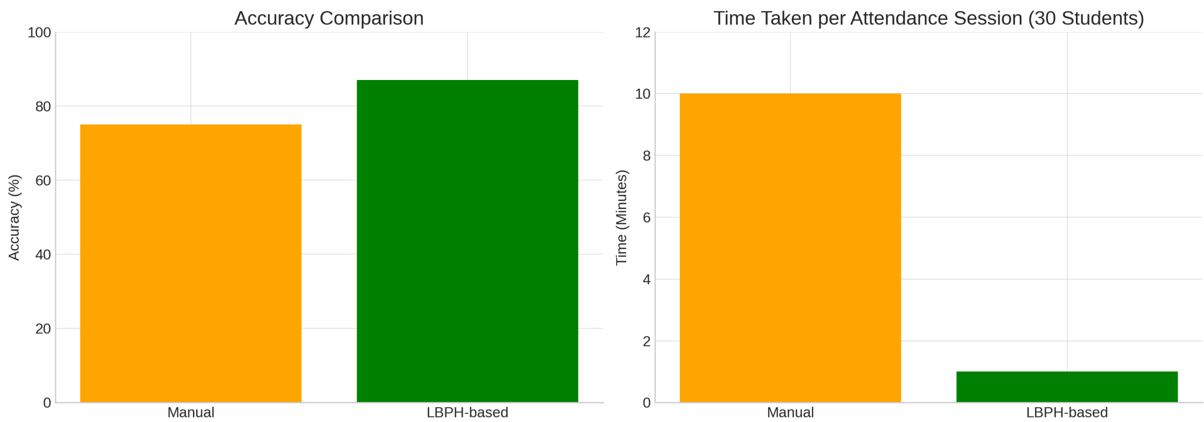

Figure 5.1: Graph

5.2 Discussion

The evaluation results of our automatic face recognition-based attendance system highlight its strong potential to transform traditional attendance management practices in educational institutions. By leveraging advanced computer vision techniques, the system provides a highly efficient, contactless, and automated solution for recording attendance in real-time. One of the most significant takeaways is the high accuracy rate consistently achieved by the system—even under varying lighting conditions, facial expressions, and viewing angles. Because it guarantees the integrity of attendance records and reduces the possibility of false positives or negatives, which are frequently connected to manual entry methods, this high accuracy is essential in educational settings. Moreover, thorough testing employing methods like 10-fold cross-validation validates the system's resilience. Confidence in the system's long-term ability to operate successfully in a wide range of different classroom environments is bolstered by its success in a wide range of different test environments.

A comparison with facial recognition systems currently available and even more traditional pen-and-paper systems shows just how much better our system is. In addition to being more efficient, the technology also eliminates problems such as human error and proxy attendance by significantly reducing the time it takes to take attendance. The discussion basically shows that our technology not only meets the practical needs of educational institutions but also provides a scalable foundation for further development and deployment in larger educational and organizational contexts.

Chapter 6

CONCLUSION

6.1 Conclusion

Our project marks a significant advancement in the field of attendance management by developing an Automatic Face Recognition Attendance Management System. Using powerful technologies like the Haar Cascade Classifier and frontal facial recognition algorithms, we were able to demonstrate the system's ability to accurately detect and distinguish individual student faces from video frames in real time. During the development process, we addressed significant issues such as illumination fluctuations, facial angles, and occlusions. Our rigorous methodology and application of state-of-the-art image processing algorithms resulted in a robust and reliable system that continuously operates well in a range of environmental conditions.

This project not only demonstrates how face recognition technology can expedite attendance procedures, but it also demonstrates how useful it is in educational settings where precision and effectiveness are crucial. Although the results of the current implementation are encouraging, we recognize that it needs to be continuously improved. Future developments will concentrate on increasing the speed and accuracy of detection, making the most of hardware and software resources, and perhaps incorporating more sophisticated algorithms to guarantee wider scalability and usability. We acknowledge that the project is a first step toward more intelligent and adaptable systems, even with the impressive results. Future improvements in system integration and algorithmic complexity will further improve our solution's scalability, accuracy, and usability.

6.2 Future work

Even though the current system performs well, there is still room for innovation and improvement in the future. Improving the accuracy of face detection and recognition by combining cutting-edge machine learning and deep learning algorithms is one exciting avenue. These techniques can potentially improve performance and flexibility under varying environmental conditions, e.g., low light, occlusions, and changing facial expressions. Another aspect for potential improvement is the use of multi-modal biometric authentication, e.g., fingerprint and iris scan, to enhance the security, robustness, and user authentication capabilities of the system. The system is also optimized for real-time operations, particularly in large classrooms. This is done through reducing latency in face detection and marking attendance to provide users an error-free experience. Improvement of the UI/UX is also necessary through a better responsive and accessible user interface (UI). A better responsive and accessible UI could be more easily accepted by administrative personnel and teachers with varying levels of technical expertise. Overall, much potential exists in the field of automatic attendance systems based on facial recognition. With these developments, subsequent generations of the system can offer more accuracy, efficiency, and user satisfaction, resulting in smarter, technologically advanced learning environments.

Bibliography

[1] Harshit Kesharwani, Tamoshree Mallick, Ayushi Gupta, Gaurav Raj, *Automated Attendance System Using Computer Vision*, IEEE, September 2022.

[2] Mohammed Abduljabbar Mohammed, Diyar Qader Zeebaree, Adnan Mohsin Abdulazeez, Dilovan Asaad Zebari, *Machine Learning Algorithm for Developing Classroom Attendance Management System Based on Haar Cascade Frontal Face*, IEEE, July 2021.

[3] Shashank Joshi, Sandeep Shinde, Prerna Shinde, Neha Sagar, Sairam Rathod, *Facial Recognition Attendance System using Machine Learning and Deep Learning*, IJERT, Vol. 12, No. 04, pp. 73–80, April 2023.

[4] Shahad Salh Ali, Jamila Harbi Al' Ameri, Thekra Abbas, *Face Detection Using Haar Cascade Algorithm*, IEEE, June 2023.

[5] Hitesh Akshat Kumar Rai, A. Akash, G. Kavyashree, Thaseen Taj, *Attendance System Based on Face Recognition Using Haar Cascade and LBPH Algorithm*, Springer Link, Vol. 947, pp. 9–21, February 2023.

[6] Michos, Gkoumas, Siakampeti, Fidas, *On the Extension of the Haar Cascade Algorithm for Face Recognition: Case Study and Results*, ACM, Vol. 10, pp. 53–56, November 2020.

[7] Ruth Ramya Kalangi, Suman Maloji, P. Syam Sundar, Sk Hasane Ahammad, *Deployment of Haar Cascade Algorithm to Detect Real-Time Faces*, IEEE, February 2022.

[8] Suraj Raj, Saikat Basu, *Attendance Automation Using Computer Vision and Biometrics-Based Authentication—A Review*, Springer Link, Vol. 58, No. 7, pp. 757–767, June 2021.

*% detected as AI

AI detection includes the possibility of false positives. Although some text in this submission is likely AI generated, scores below the 20% threshold are not surfaced because they have a higher likelihood of false positives.

Caution: Review required.

It is essential to understand the limitations of AI detection before making decisions about a student's work. We encourage you to learn more about Turnitin's AI detection capabilities before using the tool.

Frequently Asked Questions

How should I interpret Turnitin's AI writing percentage and false positives?
The percentage shown in the AI writing report is the amount of qualifying text within the submission that Turnitin's AI writing detection model determines was either likely AI-generated text from a large-language model or likely AI-generated text that was likely revised using an AI-paraphrase tool or word spinner.

False positives (incorrectly flagging human-written text as AI-generated) are a possibility in AI models.

AI detection scores under 20%, which we do not surface in new reports, have a higher likelihood of false positives. To reduce the likelihood of misinterpretation, no score or highlights are attributed and are indicated with an asterisk in the report (*%).

The AI writing percentage should not be the sole basis to determine whether misconduct has occurred. The reviewer/instructor should use the percentage as a means to start a formative conversation with their student and/or use it to examine the submitted assignment in accordance with their school's policies.

What does 'qualifying text' mean?
Our model only processes qualifying text in the form of long-form writing. Long-form writing means individual sentences contained in paragraphs that make up a longer piece of written work, such as an essay, a dissertation, or an article, etc. Qualifying text that has been determined to be likely AI-generated will be highlighted in cyan in the submission, and likely AI-generated and then likely AI-paraphrased will be highlighted purple.

Non-qualifying text, such as bullet points, annotated bibliographies, etc., will not be processed and can create disparity between the submission highlights and the percentage shown.

ORIGINALITY REPORT

11%	7%	4%	8%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to University of Petroleum and Energy Studies Student Paper	4%
2	www.kluniversity.in Internet Source	1%
3	Bui Thanh Hung, M. Sekar, Ayhan Esi, R. Senthil Kumar. "Applications of Mathematics in Science and Technology - International Conference on Mathematical Appplications in Science and Technology", CRC Press, 2025 Publication	<1%
4	S. Prasad Jones Christydass, Nurhayati Nurhayati, S. Kannadhasan. "Hybrid and Advanced Technologies", CRC Press, 2025 Publication	<1%
5	dr.ddn.upes.ac.in:8080 Internet Source	<1%
6	Submitted to University of Greenwich Student Paper	<1%

Submitted to RDI Distance Learning

7	Student Paper	<1%
8	www.stcrs.com.ly Internet Source	<1%
9	Submitted to Universiti Tunku Abdul Rahman Student Paper	<1%
10	escholarship.mcgill.ca Internet Source	<1%
11	Submitted to University of Glasgow Student Paper	<1%
12	www.ir.juit.ac.in:8080 Internet Source	<1%
13	Aleem Ali, Rajdeep Chakraborty, Nawaf R. Alharbe. "Artificial Intelligence and Optimization Techniques for Smart Information System Generations", CRC Press, 2025 Publication	<1%
14	Gaurav Aggarwal, Ashutosh Tripathi, Himani Goyal Sharma, Tripti Sharma, Rishabh Dev Shukla. "Integrated Technologies in Electrical, Electronics and Biotechnology Engineering", CRC Press, 2025 Publication	<1%
15	Submitted to Indian Institute of Technology, Kanpur	<1%