كلية الحاسبات والذكاء الإصطناعي
Faculty of Computers & Artificial Intelligence

جامعة حلوان
Helwan University

Helwan University

Faculty of Computers and Artificial Intelligence

Computer Science & Artificial Intelligence &
Information Technology Departments

Automated Attendance System using Facial Recognition

A graduation project dissertation by:

Mark Ashraf Fouad Fahim [20210707]
Martin Ashraf Younan Mahrous [20210703]
Kerols Hany Sobhy Safian [20210698]
Amr Eid Said Ali [20210643]
Fady Maged Fawzey Yousef [20210653]
Mina Nady Farag Tawfik [20210986]

Submitted in partial fulfillment of the requirements for the degree of Bachelor of Science in Computers & Artificial Intelligence at the **Computer Science, Artificial Intelligence, and Information Technology** Departments,
The Faculty of Computers & Artificial Intelligence, Helwan University

Supervised by:

Dr. Yasser Hifny   Dr. Ahmed Hisham

June 2025

كلية الحاسبات والذكاء الإصطناعي
Faculty of Computers & Artificial Intelligence

جامعة حلوان
Helwan University

جامعة حلوان

كلية الحاسبات و الذكاء الإصطناعي

اقسم : علوم الحاسب و الذكاء الإصطناعي و تكنولوجيا المعلومات

نظام الحضور الآلي باستخدام التعرف على الوجه

رساله تخرج بواسطة :

مارك اشرف فؤاد فهيم [20210707]
مارتن اشرف يونان محروس [20210703]
كيرلس هاني صبحي صفيان [20210698]
عمرو عيد سيد علي [20210643]
فادي ماجد فوزي يوسف [20210653]
مينا نادي فرج توفيق [20210986]

رسالة مقدمه ضمن متطلبات الحصول علي درجة البكالوريوس في الحاسبات و الذكاء الإصطناعي، بأقسام : علوم الحاسب و الذكاء الإصطناعي و تكنولوجيا المعلومات
كلية الحاسبات و الذكاء الإصطناعي، جامعة حلوان

بإشراف :

أ.د. ياسر حفني د. أحمد هشام

يونيو/ حزيران 2025

Contents

Abstract

Attendance tracking is an essential administrative task in educational institutions and workplaces, but manual methods can be time-consuming and error-prone. This project proposes an Automated Attendance System using Facial Recognition technology to streamline attendance monitoring. The system captures images via a camera, identifies individuals by matching their facial features with a pre-registered database, and automatically logs attendance records. This approach improves accuracy, reduces time and effort, and enhances the overall efficiency of attendance management. The system is designed to be scalable and adaptable for schools, universities, and corporate environments. Implementation details include the use of computer vision algorithms, a face recognition model, and database integration. Experimental results demonstrate high recognition accuracy and reliable attendance logging in real-time scenarios.

Keywords

Automated Attendance System, Facial Recognition, Computer Vision, Image Processing, Machine Learning, Real-time Attendance Tracking, Face Detection, Database Integration

Acknowledgement

We want to extend our sincere gratitude to everyone who has helped and mentored us as we take this important step in our academic career.

Our sincere gratitude goes out to our supervisors, Dr.Yasser Hifny and Dr.Ahmed Hisham. We would not have succeeded without your persistent encouragement, perceptive criticism, and unshakable support. Both academically and personally, your knowledge and commitment have motivated us to pursue greatness and have greatly aided in our development. For all of your time and work in guiding us, we sincerely appreciate it. Finally, we extend our deepest appreciation to our parents. Your love, patience, and sacrifices have provided us with the foundation needed to pursue our dreams. Your belief in our abilities and your constant support have been our driving force. Thank you all for being a part of this incredible journey. Your support has made this achievement possible.

List of Figures

Glossary

Automated Attendance System: A system designed to automatically track the attendance of individuals based on their facial features using facial recognition technology, eliminating the need for manual attendance recording.

Facial Recognition: A biometric technology that identifies or verifies individuals based on unique facial features, enabling applications such as attendance tracking, security, and user authentication.

Face Encoding: A mathematical representation of the features of a person's face, which is used for comparison and recognition. It is generated using facial recognition algorithms.

Face_recognition Library: A Python library that simplifies the process of facial recognition by providing pre-trained models to detect and recognize faces in images or video streams.

OpenCV: An open-source computer vision library that provides tools for image and video processing, including face detection, which is used in this system for capturing and processing video frames.

Machine Learning: A branch of artificial intelligence that involves training algorithms to recognize patterns and make decisions based on data. In this project, machine learning models are used for facial recognition.

Real-time Processing: The ability of a system to process data as it is being captured or generated, providing immediate output or actions, such as logging attendance as soon as a face is recognized.

Attendance Log: A record that tracks the presence of individuals, typically containing details like the person’s name, ID, date, attendance time, and leave time. In this system, the log is automatically updated with facial recognition.

Database Integration: The process of connecting the attendance system to a database (e.g., Excel or SQL) to store, retrieve, and manage attendance data, such as employee/student information and attendance logs.

Flask: A lightweight Python web framework used to build the EB application for managing attendance records, providing a user interface for administrators, and handling backend services like user authentication and data storage.

Camera Stream: A continuous video feed captured by a camera, which is processed to detect faces in real time. This stream is used for live facial recognition and attendance logging in the system.

Face Detection: The process of identifying faces in images or video frames. This is the first step in facial recognition, where the system locates and isolates the face before performing recognition.

Employee/Student Enrollment: The process of registering individuals in the system by capturing and storing their facial images, which will later be used for recognition during attendance logging.

Attendance Database: A structured collection of data where attendance records are stored, often in a tabular format. This database is updated automatically each time attendance is logged.

Upsampling: A technique used in face detection to increase the accuracy of face location detection by processing the image at multiple scales.

Threshold Confidence: A measure of how confident the system is that a face matches a registered individual. In this system, if the confidence level is above a certain threshold, the attendance is logged.

Chapter 1: Introduction

1.1 Overview

Attendance tracking in educational and workplace environments is an essential but often time-consuming task. Traditional methods, such as roll calls and manual logging, are prone to errors and inefficiencies. This project addresses these challenges by developing an Automated Attendance System using Facial Recognition. The system utilizes a camera to capture images in real-time, processes them through facial recognition algorithms, and automatically logs attendance in a database. The solution is built using Python libraries such as OpenCV for image processing and Face_recognition for facial detection and recognition, providing a scalable and contactless attendance system that can be implemented in schools, universities, and workplaces.

1.2 Problem Statement

The manual attendance tracking process often leads to mistakes, including proxy attendance, missed records, and human errors. This system aims to automate attendance logging to eliminate these issues, providing a more accurate, efficient, and reliable solution. The challenges include developing a robust facial recognition system that works in real-time, under varying lighting and environmental conditions, and ensuring it integrates smoothly with a backend database for storing attendance records.

1.3 Goal and Purpose

1. **Goal**

To create a fully automated attendance system that uses facial recognition to track attendance in real time and store the data securely in a database. The system aims to improve efficiency and accuracy by automating manual processes, reducing human error, and offering a seamless experience.

2. Purpose

This project aims to reduce the administrative burden of attendance tracking by implementing an automated, secure, and scalable system that provides real-time attendance updates, enhances accuracy, and reduces the need for physical interaction. It will be suitable for deployment in educational institutions and workplaces where accurate attendance is critical.

1.4 Work Methodology

The development of the Automated Attendance System follows a structured approach, consisting of the following key steps:

1. **Problem Identification:** Understanding the limitations of traditional attendance systems and identifying the need for automation.

2. **Solution Design:** Choosing facial recognition as the core technology, leveraging libraries like Face_recognition and OpenCV for face detection and image processing.

3. **Dataset Creation:** Capturing employee or student photos for enrollment in the system using the camera and storing their face encodings.

4. **System Development:** Building the Flask-based web application to handle user login, attendance recording, and data management.

5. **Testing and Evaluation:** Evaluating the system for accuracy in face recognition, real-time performance, and integration with the attendance database.

1.5 Objectives and Scope

The Automated Attendance System using Facial Recognition aims to provide a seamless, efficient, and reliable solution for attendance tracking by leveraging advanced computer vision and machine learning technologies. Below is a detailed breakdown of the objectives and scope of this project:

1. **Objectives:**

Develop a Facial Recognition Model: The system will use facial recognition to automatically identify individuals and log their attendance. It will process images from a camera, compare them with stored face encodings, and ensure accurate identification.

Build a User-Friendly Web Interface: An admin dashboard will allow easy management of user profiles, attendance records, and system settings. The interface will be simple to navigate, even for non-technical users.

Integrate a Secure Attendance Database: Attendance records will be stored in a secure database, accessible for reporting and further analysis. The system will handle user data safely, ensuring privacy and data integrity.

Real-time Face Detection and Attendance Logging: The system will detect faces in real-time and log attendance instantly without requiring manual input. This feature will operate continuously during the hours of operation.

Scalability and Adaptability: The system will be designed to scale and work in different environments, from classrooms to office buildings. It will be able to handle multiple users and adapt to the needs of larger organizations.

2. Scope:

User Enrollment: Users (students/employees) will be enrolled by capturing their facial images, which will be processed and stored for future recognition.

Automated Attendance Logging: Attendance will be logged automatically when the system detects a recognized face, with minimal delay and no manual interaction needed.

Admin Control Panel: Administrators will be able to manage users, view attendance logs, and generate reports on the system's performance.

System Deployment: The system will be deployed in environments like classrooms, offices, or meeting rooms, and will work under typical lighting conditions

Chapter 2: Literature Review

2.1 Introduction

Automating attendance systems has become increasingly important in educational and professional settings to enhance efficiency and accuracy. Traditional methods, such as manual roll calls or RFID-based systems, are often prone to errors and time-consuming. The integration of facial recognition technology offers a promising solution by providing a contactless, accurate, and real-time method for attendance tracking.

2.2 Facial Recognition Technology

Facial recognition technology has evolved significantly over the years. Early systems relied on geometric feature-based methods, which were sensitive to variations in lighting and facial expressions. Modern approaches utilize deep learning techniques, particularly Convolutional Neural Networks (CNNs), to achieve higher accuracy and robustness.

Face Detection Algorithms: Haar Cascades and Histogram of Oriented Gradients (HOG) are traditional methods for face detection. However, CNN-based models have surpassed these in performance, offering better accuracy and real-time processing capabilities.

Face Recognition Models: Deep learning models like FaceNet and DeepFace have set new benchmarks in facial recognition. FaceNet, for instance, maps faces to a 128-dimensional Euclidean space, enabling efficient face comparison and identification.

en.wikipedia.org

Libraries and Frameworks: Libraries such as OpenCV and dlib provide implementations of these algorithms, facilitating the development of facial recognition systems. The face_recognition library, built on dlib, offers a user-friendly interface for face detection and recognition tasks.

2.3 Machine Learning in Attendance Systems

Machine learning plays a crucial role in enhancing the performance of facial recognition systems in attendance applications.

Supervised Learning: Algorithms like Support Vector Machines (SVMs) and k-Nearest Neighbors (k-NN) have been employed for classifying faces based on extracted features. These methods require labeled datasets for training.

Deep Learning: CNNs, as mentioned earlier, have been extensively used for feature extraction and classification tasks. They are particularly effective in handling large datasets and capturing complex patterns in facial images.

Model Deployment: Once trained, these models can be deployed using frameworks like Flask for web applications, allowing real-time attendance logging via a camera feed.

2.4 Computer Vision Techniques

Computer vision techniques are integral to processing and analyzing visual data for attendance systems.

Image Preprocessing: Techniques such as grayscale conversion, histogram equalization, and image normalization are applied to enhance image quality and improve recognition accuracy.

Face Alignment: Aligning faces to a standard pose helps in reducing variations due to head tilts and rotations, thereby improving recognition performance.

Real-time Processing: Utilizing efficient algorithms and hardware acceleration (e.g., GPU processing) enables real-time face detection and recognition, which is essential for live attendance tracking.

2.5 Challenges and Considerations

Despite the advancements, several challenges persist in implementing facial recognition-based attendance systems:

Environmental Variations: Changes in lighting conditions and background can affect the accuracy of face detection and recognition.

Privacy Concerns: The use of biometric data raises ethical and legal issues regarding data privacy and consent.

Scalability: Handling a large number of users and ensuring the system's performance remains optimal is a significant challenge.

Bias and Fairness: Studies have shown that facial recognition systems can exhibit biases based on age, gender, and ethnicity, leading to unequal performance across different demographic groups.

2.6 Summary

The integration of facial recognition technology into attendance systems offers a promising solution to traditional attendance tracking methods. By leveraging advancements in machine learning and computer vision, these systems can provide accurate, efficient, and secure attendance management. However, addressing the associated challenges, particularly concerning privacy and fairness, is crucial for the widespread adoption of such technologies.

Chapter 3: Proposed Solution

3.1 Functional/Non-functional Requirements

The development of the system necessitates a comprehensive understanding of both functional and nonfunctional requirements to ensure its robustness, usability, and performance. The functional requirements outline the essential capabilities and features that the system must possess to fulfill user needs effectively. In contrast, the nonfunctional requirements define the quality attributes and constraints that the system must adhere to, ensuring security, reliability, and maintainability. Below are the detailed functional and nonfunctional requirements for the system.

Functional Requirements:

Facial Recognition: The core functionality of the system is the ability to detect and recognize individuals' faces in real-time using a camera feed. The system must accurately match the detected face to an entry in the database and automatically log attendance when a face is recognized.

User Enrollment: Administrators can add new users (e.g., students or employees) to the system by capturing multiple images of their faces using a camera. The system will generate unique face encodings for each user and store them in the database for future comparisons. The users' data (name, ID, department) is also stored.

Real-time Attendance Logging: Once an individual is recognized by the system, their attendance will be logged automatically with a timestamp. The system will check if the user is already marked present for the day to avoid duplicate entries.

Admin Interface: A web-based dashboard will allow administrators to manage the system. Administrators can log in securely, view the attendance records, add new users, update records, and generate attendance reports. It will be designed to be intuitive and accessible to non-technical users.

Leave Recording: In addition to attendance, the system will allow users to log their leave times. When a user leaves, the system will record the leave time against the user’s attendance record. This ensures the system can track both attendance and leave periods.

Non-functional Requirements:

Performance: The system must process and identify faces in real-time with minimal delay. The video feed should be processed in near real-time, ideally with a maximum delay of less than 2 seconds between detection and logging attendance.

Accuracy: The facial recognition model should achieve a high recognition accuracy (typically 95% or above). The system should be able to identify faces reliably even in challenging conditions (e.g., different lighting, varying angles).

Scalability: The system must be scalable to handle growing numbers of users. As the system is used in larger environments (e.g., campuses with hundreds of students), it should perform well without significant degradation in speed or accuracy.

Security: Biometric data such as facial encodings and attendance logs must be stored securely. The system must ensure that user data is encrypted both during transmission (via HTTPS) and in storage. Access to the system must be restricted, with role-based permissions (admin, user).

Usability: The system should have an easy-to-use interface that requires minimal training for administrators. The enrollment process should be straightforward, and the attendance logging should be automatic and unobtrusive.

3.2 System Analysis & Design

1. Use Case Diagram

1.1. Employee

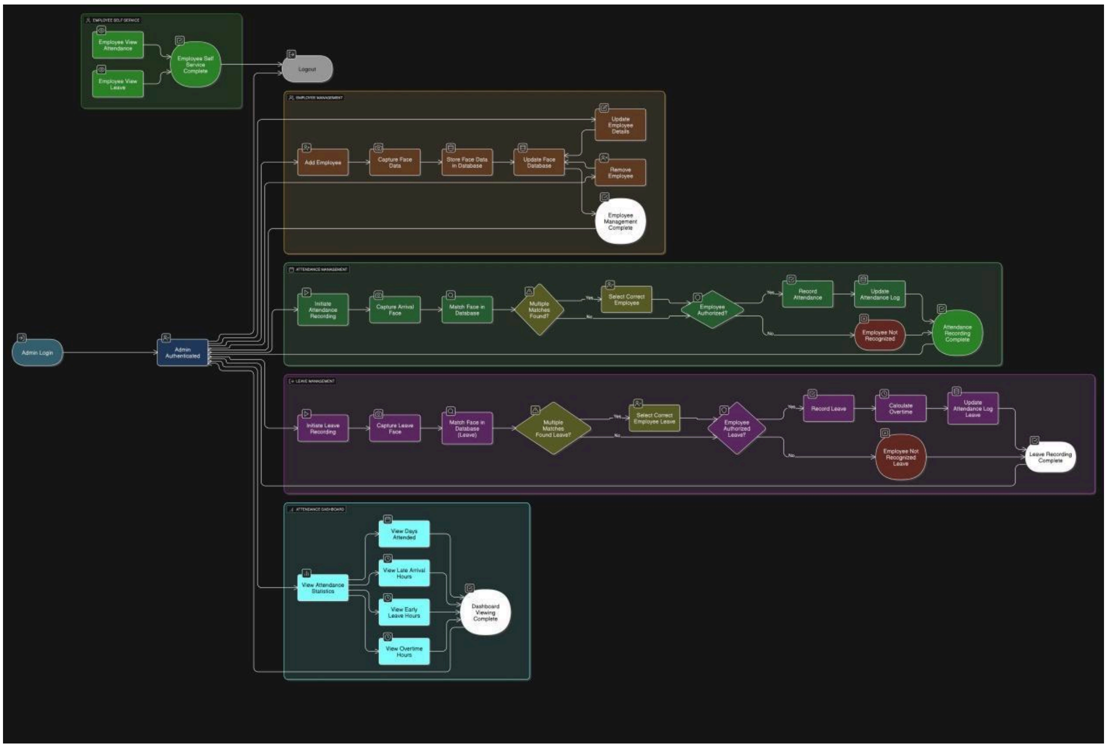


Figure 1: Employee Use Cases

This Use Case Diagram illustrates the interactions between the actors (admin and employees) and the system. The system is designed to manage employee attendance and leave through a web interface, with multiple functionalities available for both admins and employees.

Admin Login: Admin logs in to authenticate.

Employee Management: Admin can add, update, or remove employees and capture their face data.

Attendance Management: The system records attendance by recognizing faces, updates logs, and handles unrecognized employees.

Leave Management: Admin records employee leave, matches faces, and calculates overtime if needed.

Employee Dashboard: Employees can view their attendance stats, arrival times, early leave, and overtime hours.

1.2. Student

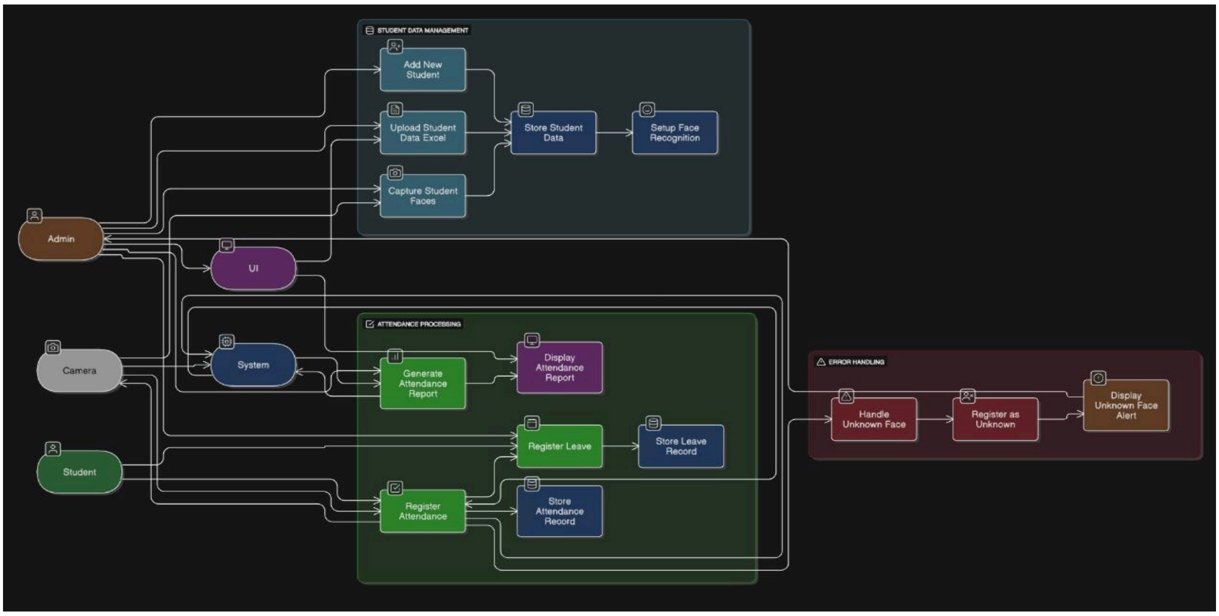


Figure 2: Student Use Case

This Use Case Diagram focuses on the interactions between the Admin, Students, and the system, covering functions like student management, attendance processing, and error handling.

Student Data Management (Admin Role):

Add New Student: The admin adds new students to the system.

Upload Student Data Excel: Admin can upload student data in bulk from an Excel file.

Store Student Data: The system stores the student information in the database.

Set up Face Recognition: The admin sets up facial recognition for the newly added students.

Capture Student Faces: Admin captures facial images for recognition.

Attendance Processing:

Register Attendance: The system registers attendance for students when they are recognized by their faces.

Store Attendance Record: Once attendance is recorded, the system stores the data in the database.

Generate Attendance Report: Admin can generate reports on attendance data for analysis.

Display Attendance Report: The generated attendance report is displayed to the admin.

Leave Management:

Register Leave: Admin can register leave for students.

Store Leave Record: The system stores leave records for students.

Error Handling:

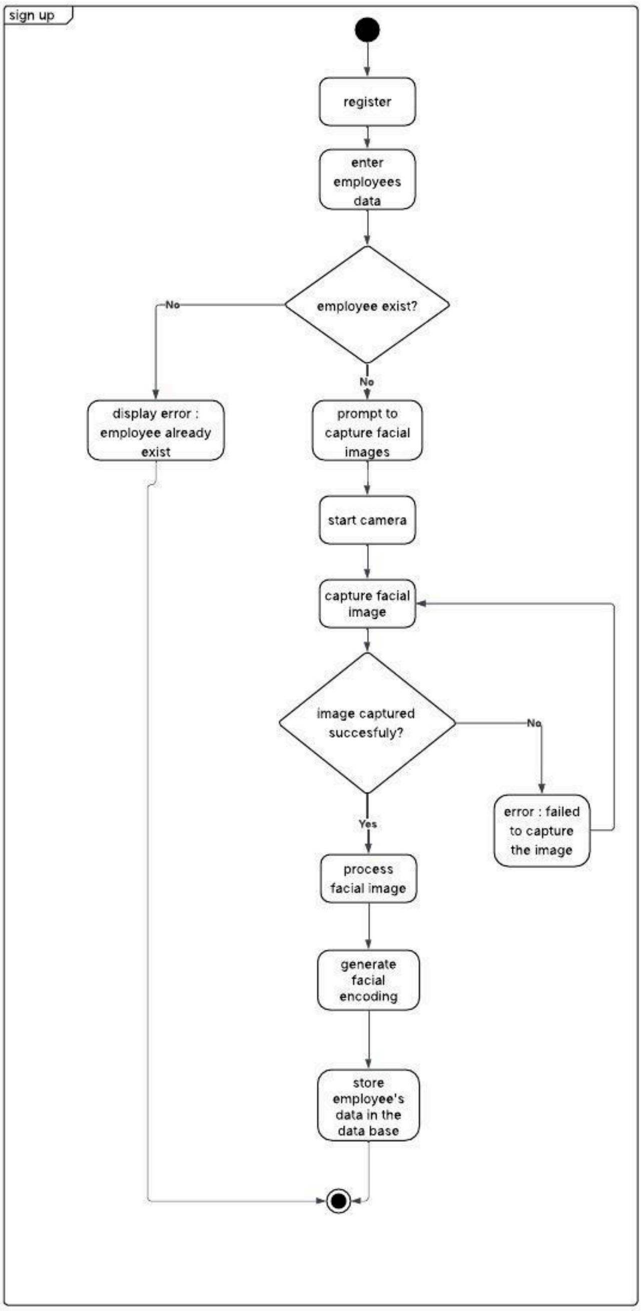
Handle Unknown Face: If an unrecognized face is detected, the system registers it as unknown.

Register as Unknown: The unrecognized face is saved in the system for later verification.

Display Unknown Face Alert: An alert is displayed to the admin when an unknown face is detected.

2. Activity Diagram

2.1. Employee


sign up
register
enter employees data
employee exist?
No
No
display error : employee already exist
prompt to capture facial images
start camera
capture facial image
image captured succesfuly?
No
Yes
error : failed to capture the image
process facial image
generate facial encoding
store employee's data in the data base

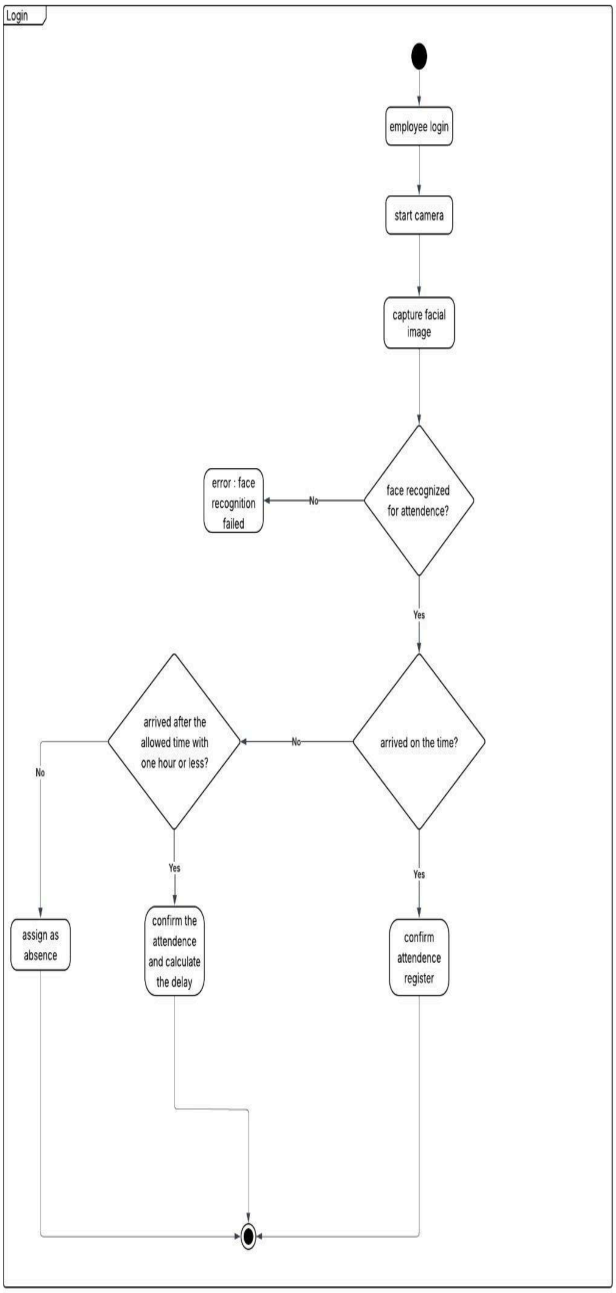

Login
employee login
start camera
capture facial image
face recognized for attendence?
No
error : face recognition failed
Yes
arrived on the time?
No
arrived after the allowed time with one hour or less?
No
Yes
Yes
assign as absence
confirm the attendence and calculate the delay
confirm attendence register

Figure 3: Activity diagram Sign up

Figure 4: Activity diagram Login

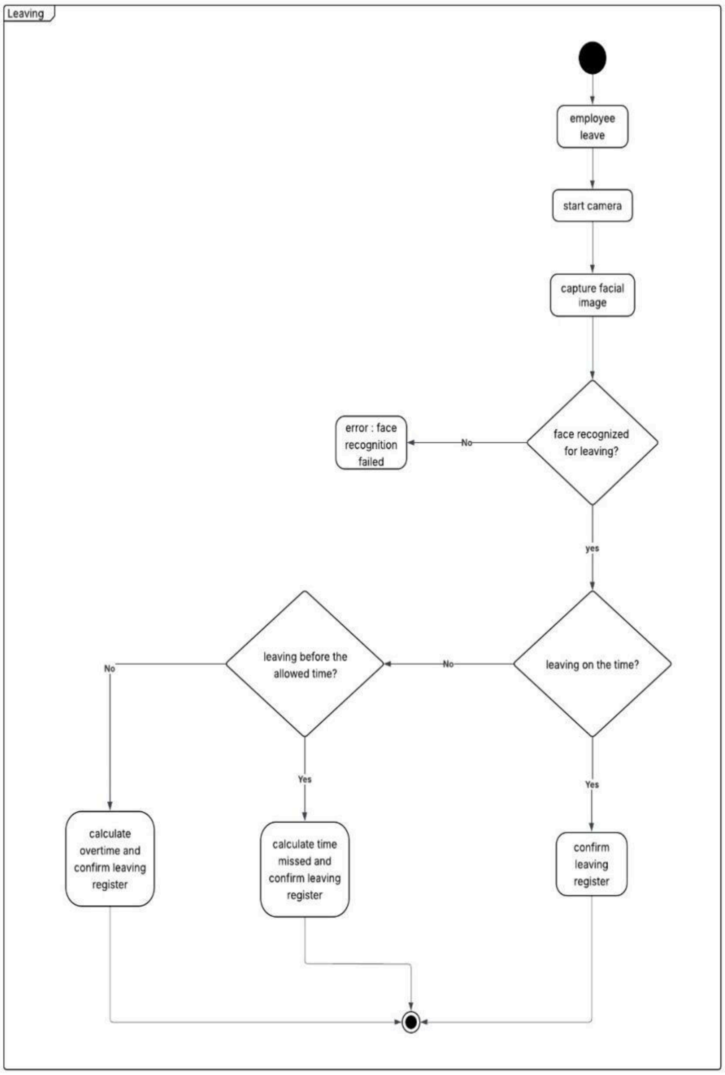

Leaving
employee leave
start camera
capture facial image
face recognized for leaving?
No
error : face recognition failed
yes
leaving on the time?
No
leaving before the allowed time?
No
Yes
Yes
calculate overtime and confirm leaving register
calculate time missed and confirm leaving register
confirm leaving register

Figure 5: Activity Diagram Leaving

1. Employee Sign-Up Activity Diagram: (Figure 3)
This diagram describes the employee registration process, where the admin or system adds a new employee.

Step 1: Register Employee
The process starts when an employee signs up, and their data is entered into the system.

Step 2: Check if Employee Exists
The system checks if the employee is already registered. If yes, it displays an error message (employee already exists). If no, the system proceeds to capture the facial image.

Step 3: Capture Facial Image
The system prompts the admin to start the camera and capture the employee's facial image.

Step 4: Image Captured Successfully?
If the image is captured successfully, the system processes it and generates the facial encoding, storing the data in the employee database. If the capture fails, an error message is shown.

2. Employee Leaving Activity Diagram: (Figure 4)
This diagram explains the process when an employee leaves and the system records the leave.

Step 1: Employee Leave Request
The employee requests to leave, and the system starts the camera to capture their face.

Step 2: Face Recognition for Leaving
The system checks if the employee's face is recognized for leave. If not, an error message is displayed. If yes, the system continues.

Step 3: Leaving Before the Allowed Time?
The system checks if the employee is leaving before the allowed time. If yes, it calculates the overtime and updates the leave record. If no, the system checks if the employee is leaving at the correct time.

Step 4: Confirm Leaving Register
If the employee is leaving on time, the system confirms and registers the leave. If the employee is leaving early, the system calculates the time missed and confirms the leave.

3. Employee Login Activity Diagram: (Figure 5)

This diagram represents the login process for an employee, where facial recognition is used for attendance.

Step 1: Employee Login

The employee attempts to log in, and the system starts the camera.

Step 2: Face Capture for Attendance

The system captures the facial image of the employee and checks if it's recognized for attendance.

Step 3: Face Recognition for Attendance?

If the face is not recognized, an error is displayed. If the face is recognized, the system moves to the next step.

Step 4: Arrived After the Allowed Time?

The system checks if the employee arrived late. If the employee arrived late (within an hour or less), the system confirms the attendance but calculates the delay. If the employee arrives on time, the system confirms the attendance directly.

Step 5: Confirm Attendance

The system confirms the attendance and registers the log.

2.2. Student

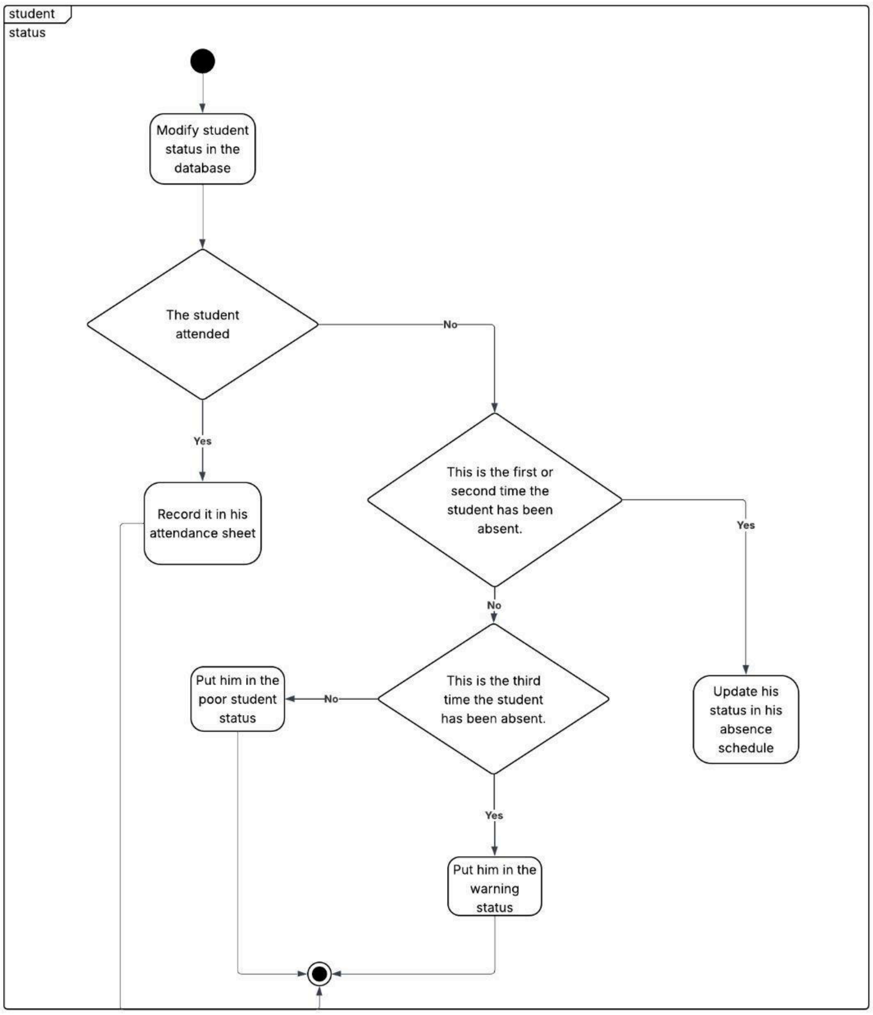

student status
Modify student status in the database
The student attended
No
Yes
Record it in his attendance sheet
This is the first or second time the student has been absent.
Yes
No
Update his status in his absence schedule
This is the third time the student has been absent.
No
Put him in the poor student status
Yes
Put him in the warning status

Figure 6: Student status in the student project

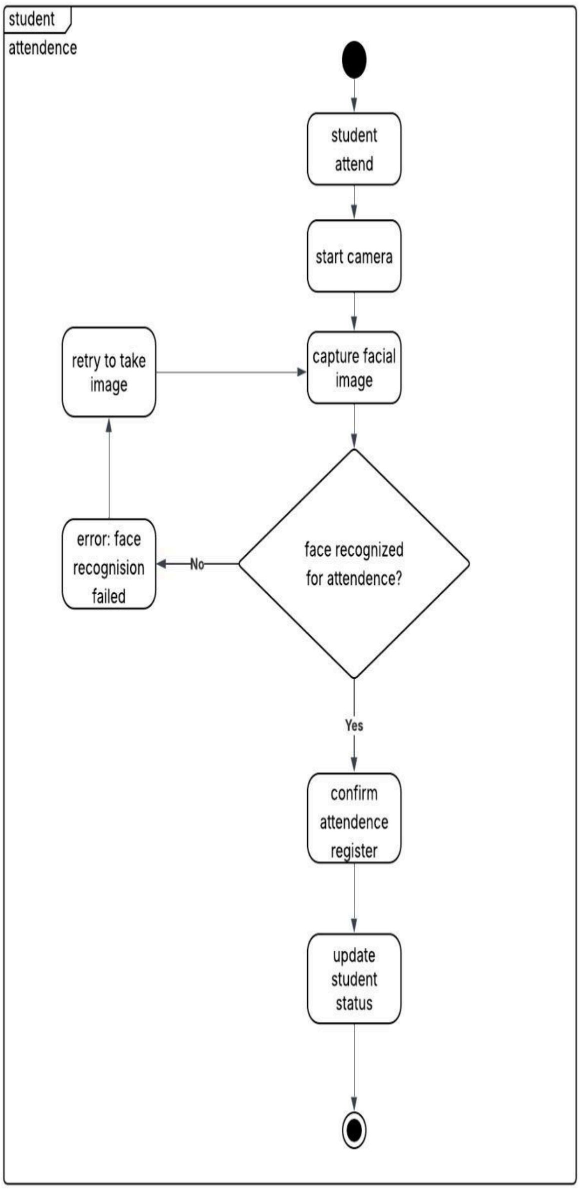


Figure 7: Student attendance in student project

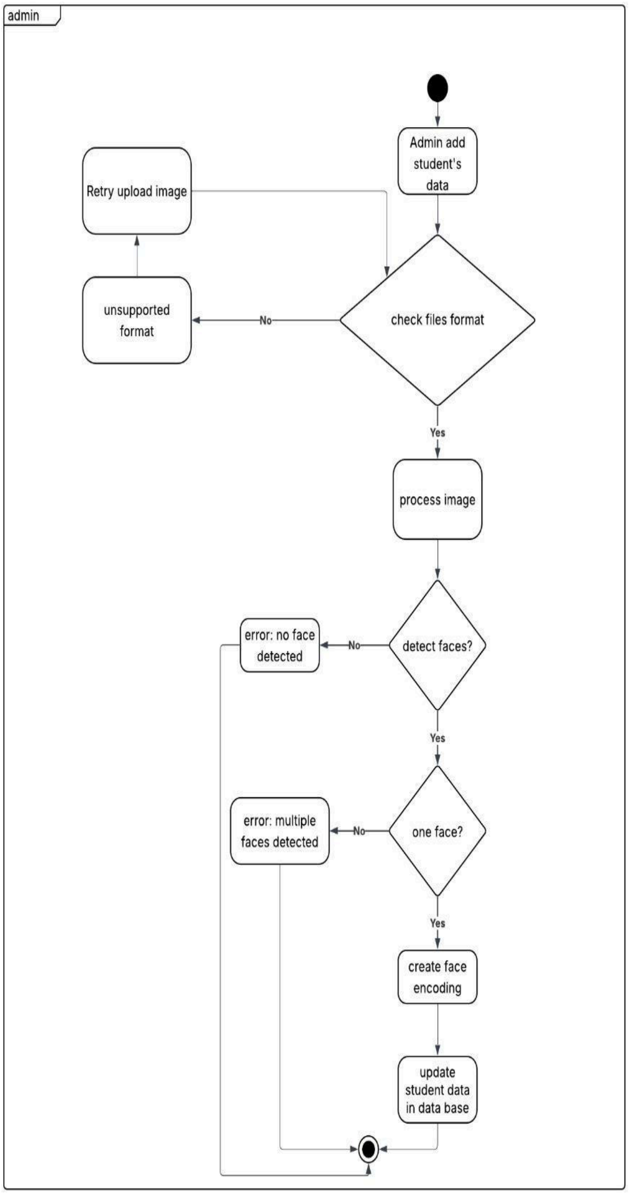


Figure 8: Admin in student project

1. Student Status Activity Diagram: (Figure 6)
This diagram outlines how the system handles the student’s attendance status (whether absent, present, or needing a warning).

Step 1: Modify Student Status:
The system updates the student's status in the database.

Step 2: Check If the Student Attended:
The system checks if the student was marked as present or absent for the day.

Step 3: Determine Absence Frequency:
If the student was absent, the system checks if this is their first or second absence. If so, it records it normally. If it’s the third absence, the system will flag the student with a "warning status."

Step 4: Update Status:
The system updates the student’s status and attendance log accordingly.

2. Student Attendance Activity Diagram: (Figure 7)
This diagram shows the process for registering a student's attendance.

Step 1: Student Attendance:
The student arrives, and the system starts the camera to capture their facial image.

Step 2: Face Recognition for Attendance:
The system tries to recognize the student's face for attendance. If the face recognition fails, the system will ask the student to retry.

Step 3: Confirm Attendance:
Once the student's face is recognized, the system confirms the attendance and updates the student’s status as "present."

3. Admin Adding Student Data Activity Diagram: (Figure 8)
This diagram explains the process of adding a new student to the system.

Step 1: Admin Registration:
The admin starts by registering the student's data into the system.

Step 2: Check File Format:
The system checks whether the uploaded file (such as an image or data file) is in a supported format. If not, the admin is prompted to upload the file again.

Step 3: Process Image:
If the file format is correct, the system processes the image captured by the admin.

Step 4: Detect Faces:
The system checks if faces are detected in the image. If no face is detected, an error is displayed. The admin is asked to capture a new image with only one face if multiple faces are detected.

Step 5: Create Face Encoding:
If one face is successfully detected, the system creates a facial encoding, which is then stored in the student database.

3. Sequence Diagram

3.1. Employee

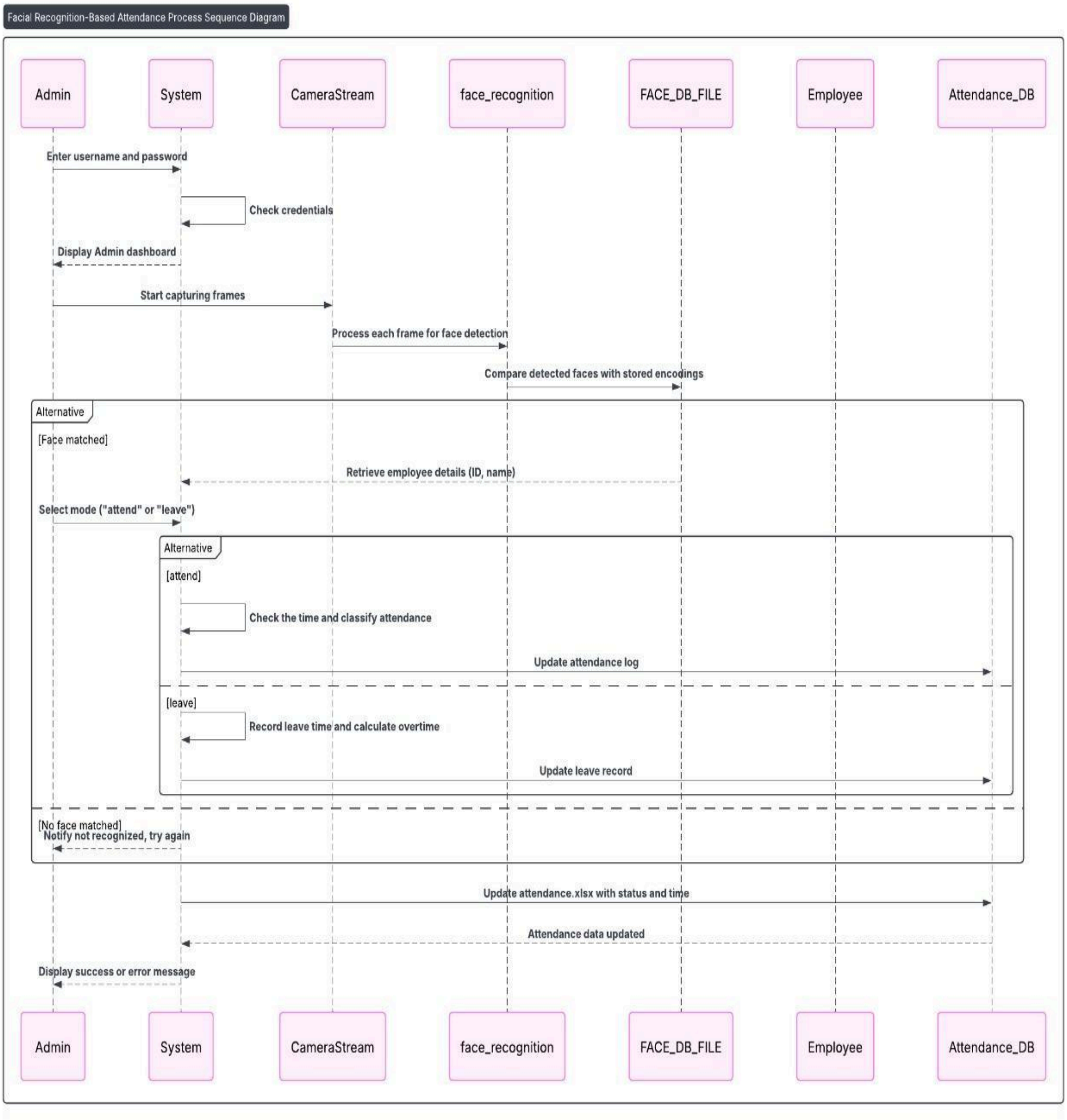

Facial Recognition-Based Attendance Process Sequence Diagram
Admin
System
CameraStream
face_recognition
FACE_DB_FILE
Employee
Attendance_DB
Enter username and password
Check credentials
Display Admin dashboard
Start capturing frames
Process each frame for face detection
Compare detected faces with stored encodings
Alternative
[Face matched]
Retrieve employee details (ID, name)
Select mode ("attend" or "leave")
Alternative
[attend]
Check the time and classify attendance
Update attendance log
[leave]
Record leave time and calculate overtime
Update leave record
[No face matched]
Notify not recognized, try again
Update attendance.xlsx with status and time
Attendance data updated
Display success or error message
Admin
System
CameraStream
face_recognition
FACE_DB_FILE
Employee
Attendance_DB

Figure 9: Sequence diagram for Employee

Sequence Diagram for Employee Explanation: (Figure 9)
This Sequence Diagram represents the interaction between various components (admin, system, camera stream, face recognition, employee, and database) in the Automated Attendance System.

Flow of Actions:

Admin Login:
Admin: The process begins with the admin entering their username and password.
System: The system checks the admin's credentials. If valid, the admin dashboard is displayed.

Start Capturing Frames:
System: The system starts capturing frames using the CameraStream to monitor the employees arriving.

Face Detection and Recognition:
Camera Stream: Captures the video frames and sends them to the face_recognition module for face detection.
Face recognition: The system compares the detected faces with the stored face encodings in the FACE_DB_FILE (the database file containing employee facial data).
Employee Data Retrieval: If a match is found, the system retrieves the employee's details (ID, name) from the database.
Attendance or Leave Mode:
Admin: The admin selects whether the employee is attending or leaving.
Attendance: If attendance is selected, the system checks the time and classifies the attendance (whether the employee is on time or late).
If leave is selected, the system records the leave time and calculates any overtime.

Update Records:
Attendance_DB: The system updates the attendance log or the leave record based on the selected mode.
Excel File Update: The attendance data is then updated in the attendance.xlsx file with the employee's status and time.

Error Handling:
If the system doesn't detect a face or multiple faces are detected, it notifies the admin and prompts for a retry.
Display Success or Error Message:
The system displays either a success message (if the attendance/leave was recorded successfully) or an error message (if no face was recognized or there was an issue).

3.2. Student

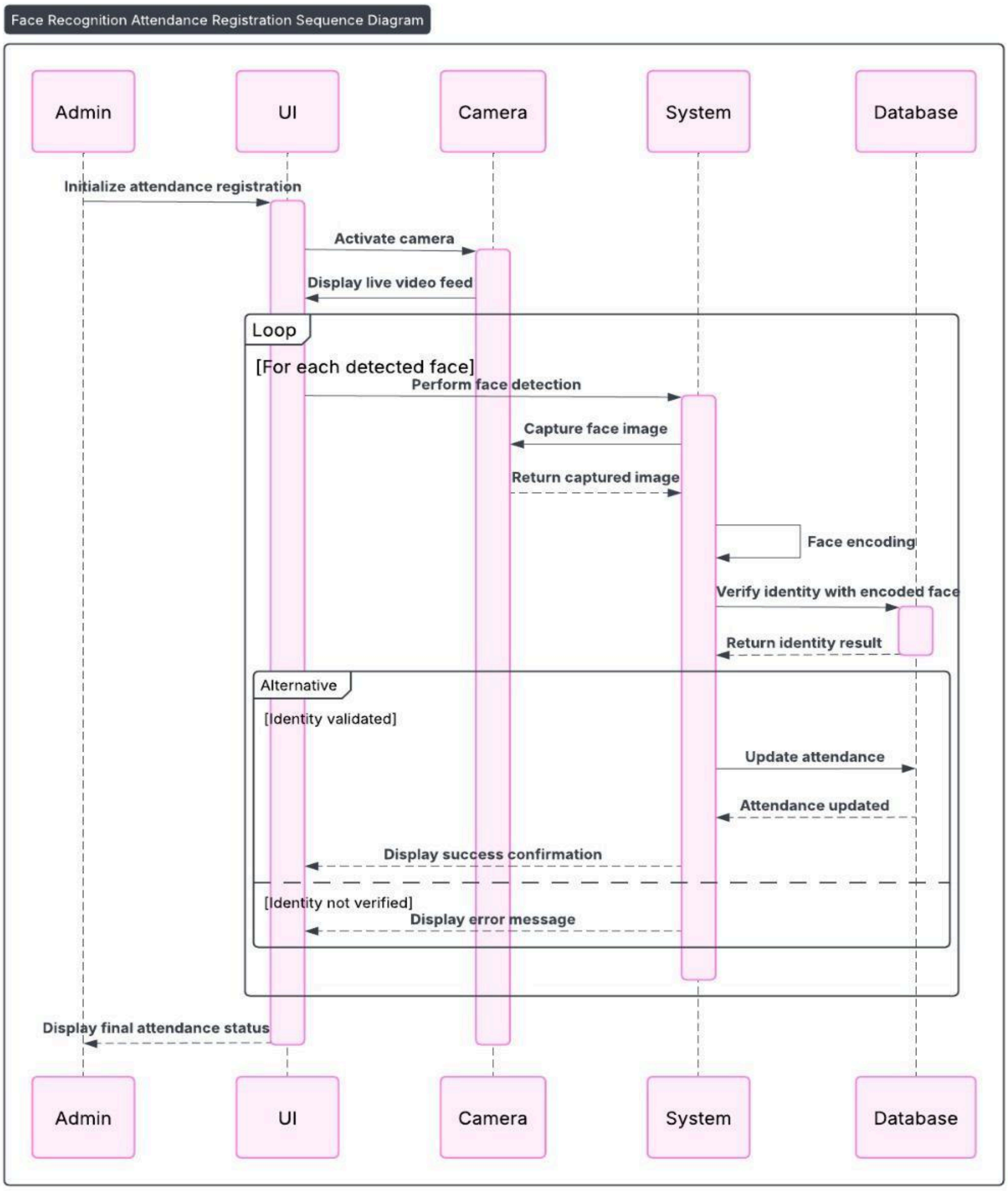


Figure 10: Sequence Diagram for Student

Sequence Diagram for Student Explanation: (Figure 10)
This diagram illustrates the process flow of registering attendance using facial recognition, starting from the admin initiating the process to the system confirming the attendance or displaying an error message.

Flow of Actions:
Admin Initializes Attendance Registration:
Admin: The process starts when the Admin initializes the attendance registration, which prompts the system to begin the facial recognition process.

Camera Activation and Video Feed:
UI & Camera: The camera is activated, and a live video feed is displayed on the admin interface. The system continuously captures video frames.

Face Detection and Image Capture:
Camera: The system performs face detection on each frame in the video feed. Once a face is detected, it captures the face image and sends it to the system for processing.

Face Encoding and Identity Verification:
System: The captured image is processed to generate a face encoding, which is compared against the stored facial data in the Database to verify the identity of the person.
Database: The system checks the Database to verify if the detected face matches a registered person's face encoding.

Identity Validation:
System: If the identity is validated (i.e., a match is found), the system confirms the identity and updates the attendance log.
UI: A success confirmation is displayed on the UI to inform the admin that the attendance has been successfully logged.

Error Handling:
If the identity is not verified (i.e., no match found), an error message is displayed, and the admin is notified.

Final Attendance Status:
UI: Once the identity is validated or the error is handled, the system displays the final attendance status (whether the user was successfully marked present or not).

4. Class Diagram

4.1. Employee

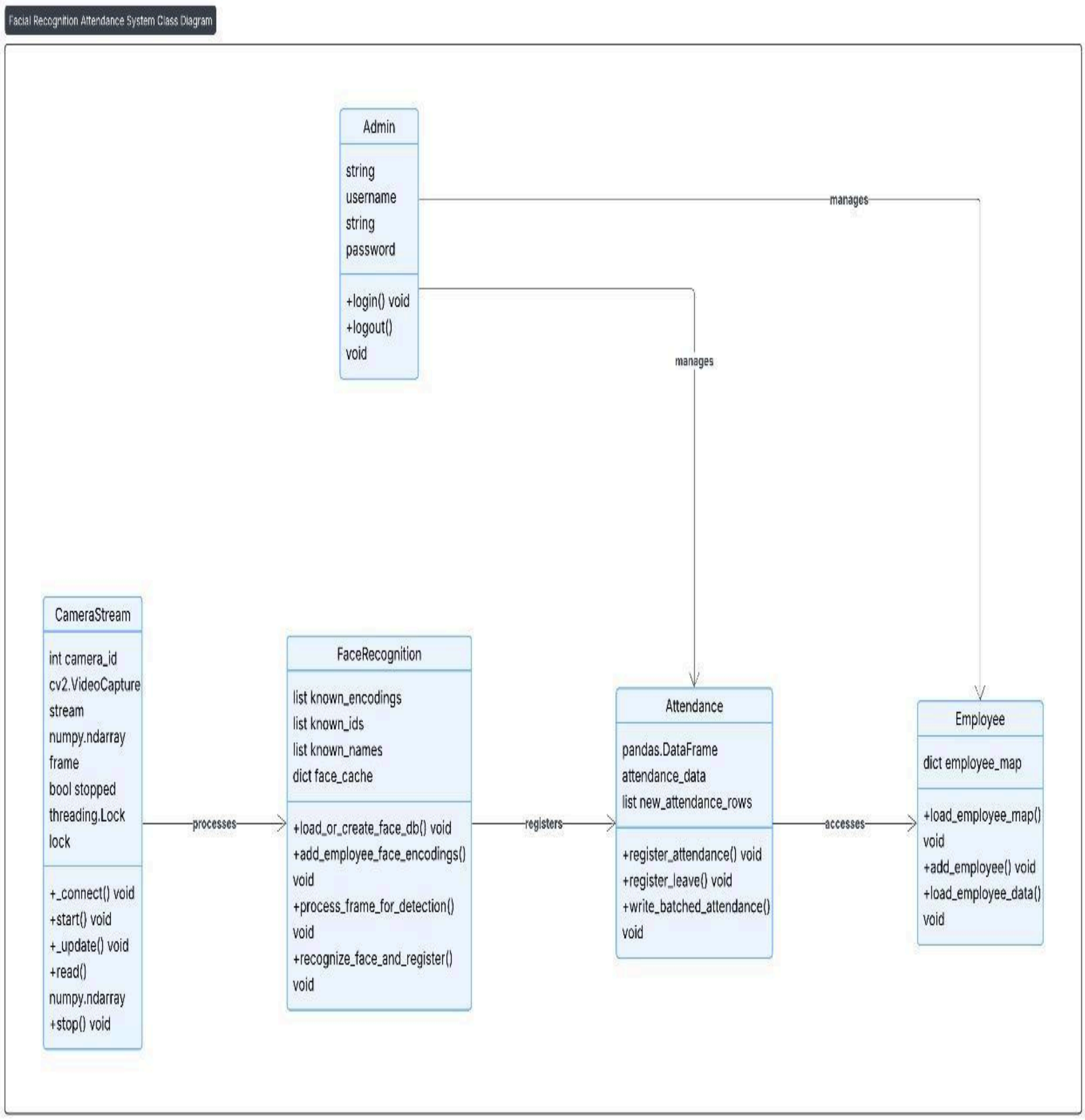


Figure 11: Class Diagram for Employee

Class Diagram for Employee Explanation (Figure 11)
This Class Diagram outlines the structure of the system, focusing on the key components such as the Admin, CameraStream, FaceRecognition, Attendance, and Employee classes.

Classes and Their Relationships:
Admin:

Attributes:
Username: The admin's username.
Password: The admin's password.

Methods:
Login (): Allows the admin to log into the system.
Logout (): Allows the admin to log out of the system.
Relationship: The Admin class manages the system and interacts with other components, like Attendance and Employee.

CameraStream:

Attributes:
camera_id: Unique identifier for the camera.
Frame: Holds the current frame being processed.
Stream: The video stream object.
Bool stopped: A flag indicating if the camera is stopped.

Methods:

Connect (): Establishes a connection with the camera.
Start (): Starts capturing video.
Update (): Updates the camera feed.
Read (): Reads the current frame.
Stop (): Stops the camera.
Relationship: The CameraStream class is responsible for capturing and processing video frames. It sends frames to the FaceRecognition class for face detection.

FaceRecognition:

Attributes:
known_encodings: List of encoded faces in the system.
known_ids: List of unique identifiers for each employee.
face_cache: Dictionary storing face data for quick access.

Methods:
load_or_create_face_db(): Loads or creates a database of faces.
add_employee_face_encodings(): Adds face encodings for new employees.
process_frame_for_detection(): Processes a video frame to detect faces.
recognize_face_and_register(): Recognizes faces in frames and registers attendance.
Relationship: The FaceRecognition class processes the video frames and compares detected faces with stored encodings to identify the employee.

Attendance:

Attributes:
attendance_data: Stores all attendance records.
new_attendance_rows: Stores new attendance data for processing.

Methods:
register_attendance(): Registers the attendance of an employee.
register_leave(): Registers the leave status of an employee.
write_batched_attendance(): Writes attendance data in batches.
Relationship: The Attendance class handles the attendance data and interacts with the Employee class to track attendance and leave.

Employee:

Attributes:
employee_map: A dictionary that maps employee data (e.g., name, ID) to their facial encodings.

Methods:
load_employee_map(): Loads employee data.
add_employee(): Adds a new employee to the system.
load_employee_data(): Loads data for individual employees.
Relationship: The Employee class provides data related to employees and works with the Attendance class to update attendance and leave statuses.

Summary:
Admin manages the entire system and interacts with the Attendance and Employee classes.
CameraStream captures the video feed, and FaceRecognition processes each frame for face detection.
Attendance and Employee handle data related to attendance logs and employee records.

4.2. Student

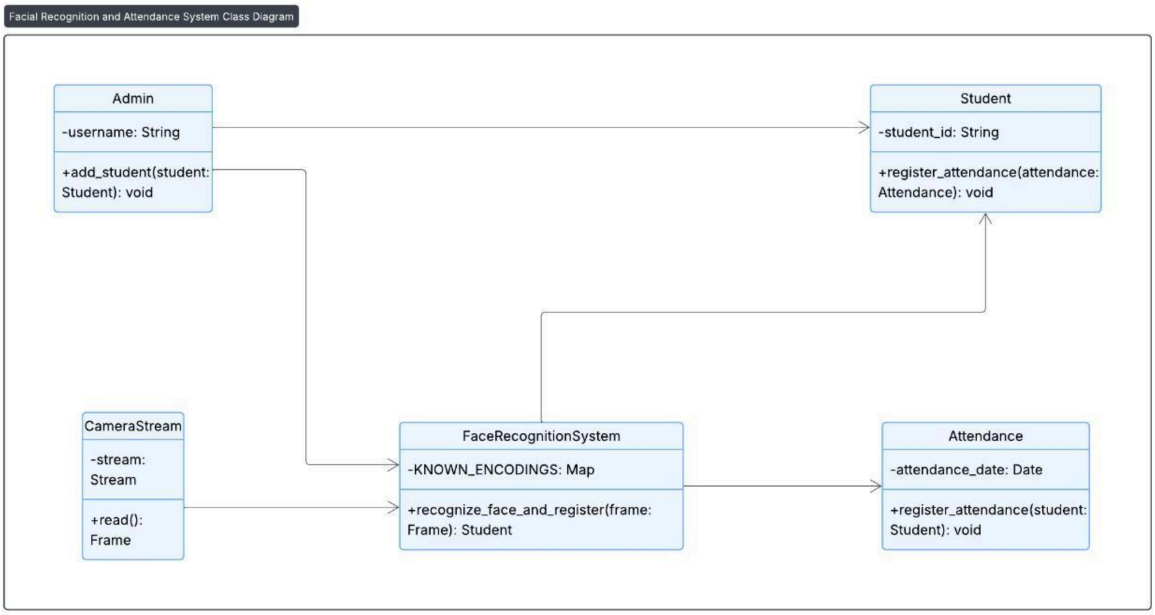

Facial Recognition and Attendance System Class Diagram
Admin
-username: String
+add_student(student: Student): void
Student
-student_id: String
+register_attendance(attendance: Attendance): void
CameraStream
-stream: Stream
+read(): Frame
FaceRecognitionSystem
-KNOWN_ENCODINGS: Map
+recognize_face_and_register(frame: Frame): Student
Attendance
-attendance_date: Date
+register_attendance(student: Student): void

Figure 12: Class Diagram for Student

Class Diagram for Student Explanation (Figure 12)
This Class Diagram represents the structure and relationships of key components in the Facial Recognition and Attendance System. The diagram shows how the Admin, Student, CameraStream, FaceRecognitionSystem, and Attendance classes are organized.

Classes and Their Relationships:
Admin:

Attributes:
Username: The admin's username.

Methods:
add_student(student: Student): This method allows the admin to add new students to the system.

Relationship: The Admin class manages the Student class by adding students and interacting with the FaceRecognitionSystem for attendance registration.

Student:

Attributes:
student_id: The unique ID for the student.

Methods:
register_attendance(attendance: Attendance): This method registers the student’s attendance based on the given attendance object.

Relationship: The Student class interacts with the Attendance class to log their attendance.

CameraStream:

Attributes:
Stream: Represents the camera stream.

Methods:
Read (): Captures frames from the video stream to process for face recognition.

Relationship: The CameraStream class captures video frames and passes them to the FaceRecognitionSystem for processing.

FaceRecognitionSystem:

Attributes:
KNOWN_ENCODINGS: A map of known facial encodings for comparison.

Methods:

recognize_face_and_register(frame: Frame): This method processes the captured video frame, recognizes the face, and registers the student's attendance if the face is identified.

Relationship: The FaceRecognitionSystem class processes the face data, compares it with the known encodings, and registers the student's attendance.

Attendance:

Attributes:
attendance_date: The date the attendance is registered.

Methods:
register_attendance(student: Student): This method registers the student's attendance for the given date.

Relationship: The Attendance class stores the attendance data and is updated when the Student's attendance is logged.

Summary:

- Admin manages students and interacts with the FaceRecognitionSystem to process attendance.
- Student interacts with the Attendance class to log their attendance.
- CameraStream captures frames, which are sent to the FaceRecognitionSystem for identification.
- FaceRecognitionSystem compares faces and registers attendance for students.
- Attendance stores attendance records.

5. System Architecture:

The system architecture can be broken down into the following key components:

Camera Module (Face Detection):

- Functionality: The camera captures live video frames that are processed for face detection.
- Technologies: OpenCV handles video capture and basic image processing. The face_recognition library is responsible for detecting and encoding faces.
- Details: The system continuously captures frames from the camera, processes them, and identifies faces. It can work in real-time, processing every nth frame (controlled by the FRAME_SKIP parameter) to improve performance.

Face Recognition Model:

- Functionality: The face_recognition library is used to perform the actual face detection and recognition. Each user's face is converted into a unique encoding that is stored in the system's database.
- Details: The system matches new faces detected in real-time with the stored encodings. When a match is found, the user is identified and their attendance is logged. The known_faces.pkl stores the encoded faces for easy retrieval during the recognition process.

Attendance Database:

- Functionality: The database stores users' attendance records, including information such as name, ID, attendance time, and leave time.
- Technologies: Initially, the system uses Excel files (attendance.xlsx and employees.xlsx) to store data, but it can be adapted to a more robust database system, such as SQL, for larger applications.
- Details: Attendance is logged automatically when a face is recognized, and leave time is also recorded when the user leaves. The system ensures data integrity and prevents duplicate attendance entries for the same day.

Admin Interface (Web Application):

- Functionality: The admin interface allows administrators to manage users, view attendance logs, and generate reports.
- Technologies: The system uses the Flask framework to create a simple and intuitive web interface.
- Details: The interface is designed to allow administrators to easily view and manage user data. The interface includes login functionality for security, a dashboard for viewing attendance statistics, and options for enrolling new users.

3.3 Core System Architecture

Here's how the core architecture works:

1. **User Enrollment:**
   - Administrators can add new users by capturing multiple images for each individual. These images are processed using the face_recognition library to generate face encodings, which are stored in the database for future recognition.
   - New users can also be enrolled through the admin interface, where user details such as name, ID, and department are recorded.

2. **Face Detection & Recognition:**

   - The system captures live video frames from the camera and processes them in real-time. The face_recognition library is used to detect faces in the frames and compare them with the stored face encodings.
   - When a face is recognized, the system matches it with the closest stored encoding. If a match is found, the system logs the user's attendance.

3. **Attendance Logging:**

   - Once the user is recognized, the system automatically logs their attendance with a timestamp (date and time).
   - The system checks if the user has already been marked present for the day to avoid duplicate entries.

4. **Leave Time Logging:**

   - If a user leaves the premises (or logs out of the system), their leave time is recorded and added to their attendance record. This allows for accurate tracking of both attendance and leave time.

3.4 Additional Modules

1. Reporting Module:

The admin can generate reports based on the logged attendance. Reports can be filtered by date range (daily, weekly, monthly) and exported in CSV or Excel formats for analysis. The report will include metrics like total days attended, days absent, and leave times.

2. Notifications Module:

The system will notify administrators when a face is detected and attendance is logged. If the system fails to recognize a face, the admin will be notified, allowing them to handle exceptions manually.

Chapter 4: Implementation and Results

Introduction

The implementation of the Automated Attendance System using Facial Recognition involves the development of a robust and efficient system that uses cutting-edge technology to automate attendance tracking. This system utilizes facial recognition to identify individuals and log their attendance in real time, replacing traditional manual or card-based attendance systems. This section outlines the implementation process, including the technologies used, the workflow of the system, and the results of several tests conducted to evaluate its performance and accuracy.

The system was built using several key technologies, including Flask, OpenCV, and the face_recognition library, and it was tested under different conditions to assess its effectiveness in real-world scenarios. The following sections provide a detailed explanation of the system's implementation, followed by the results and performance metrics gathered through testing.

4.1 Implementation Details

The implementation of the Automated Attendance System using Facial Recognition involved several key components and technologies. Below is a detailed explanation of each part:

Technologies Used:

Flask:

The backend of the system is built using Flask, a lightweight Python web framework that helps to easily handle HTTP requests, manage sessions, and render dynamic content. Flask serves as the core for building the web interface.

face_recognition Library:

The face_recognition library (built on dlib) is used for face detection, face encoding, and recognition. This library simplifies the process of recognizing and comparing faces in real-time video streams.

OpenCV:

OpenCV is used for capturing live video frames from the camera and performing image processing tasks like resizing frames, converting to RGB, and displaying the results on the admin interface.

Pandas and Excel:

Pandas is used for reading and writing data to Excel files that store the attendance logs and employee information. Initially, the Excel format is used for simplicity, but the system can scale to SQL databases for larger deployments.

SQLite (Optional for Future Use):

For scalability, SQLite or another SQL-based database can replace the Excel files to store data and ensure better performance when handling large user bases.

System Workflow:

User Enrollment:

Administrators can add new users (employees or students) by capturing facial images using a camera. These images are processed and stored as facial encodings in a pickle file for future recognition.

Face Detection and Recognition:

The system captures real-time video frames, processes them to detect faces, and compares them against stored face encodings. When a match is found, the system logs attendance or leave time.

Attendance and Leave Logging:

Attendance: Once an employee or student is recognized, their attendance is logged with a timestamp.

Leave: The system also records the leave time and calculates any overtime if necessary.

Code Structure:

The project is structured into different modules:

app.py: Handles the main web application logic, including routes for login, attendance recording, and reporting.

face_recognition_helper.py: Contains functions for face encoding, recognition, and loading the face database.

attendance.py: Manages the creation and update of attendance records.

camera.py: Handles camera initialization and frame capturing.

File Management:
The system uses Excel files to store attendance data, such as attendance.xlsx and employees.xlsx, but this can be adapted to SQL databases as the system scales.

4.2 Experimental Setup

To evaluate the system, several experiments were conducted to assess its performance. The tests focused on:

Face Detection Accuracy:
Testing was done under various conditions (different lighting, face orientation, and partial occlusions). The goal was to evaluate how reliably the system detects and recognizes faces.

Real-time Processing Speed:
The processing speed was measured, specifically the time taken from detecting a face to logging the attendance, ensuring the system operates in real-time.

Scalability:
Tests were conducted to evaluate how the system handles a growing number of users, simulating larger user bases (e.g., hundreds of students or employees).

User Enrollment and Error Handling:
The process of enrolling users (capturing photos, encoding faces) was tested for accuracy and ease of use. Error handling was also tested, particularly when no face is detected or when multiple faces are detected in a frame.

4.3 Results and Performance

The key results from the experiments were:

Face Detection Accuracy:

The system achieved an accuracy of 95% in recognizing faces, with slightly higher accuracy under ideal lighting conditions (~98%). In low-light environments, the accuracy dropped, but it could still reliably detect faces with proper camera adjustments.

The system's performance varied depending on the confidence threshold. The graph below illustrates the relationship between the confidence threshold and the F1-score, precision, and recall. The F1-score is most effective at a threshold of 0.30, providing the best balance between precision and recall.

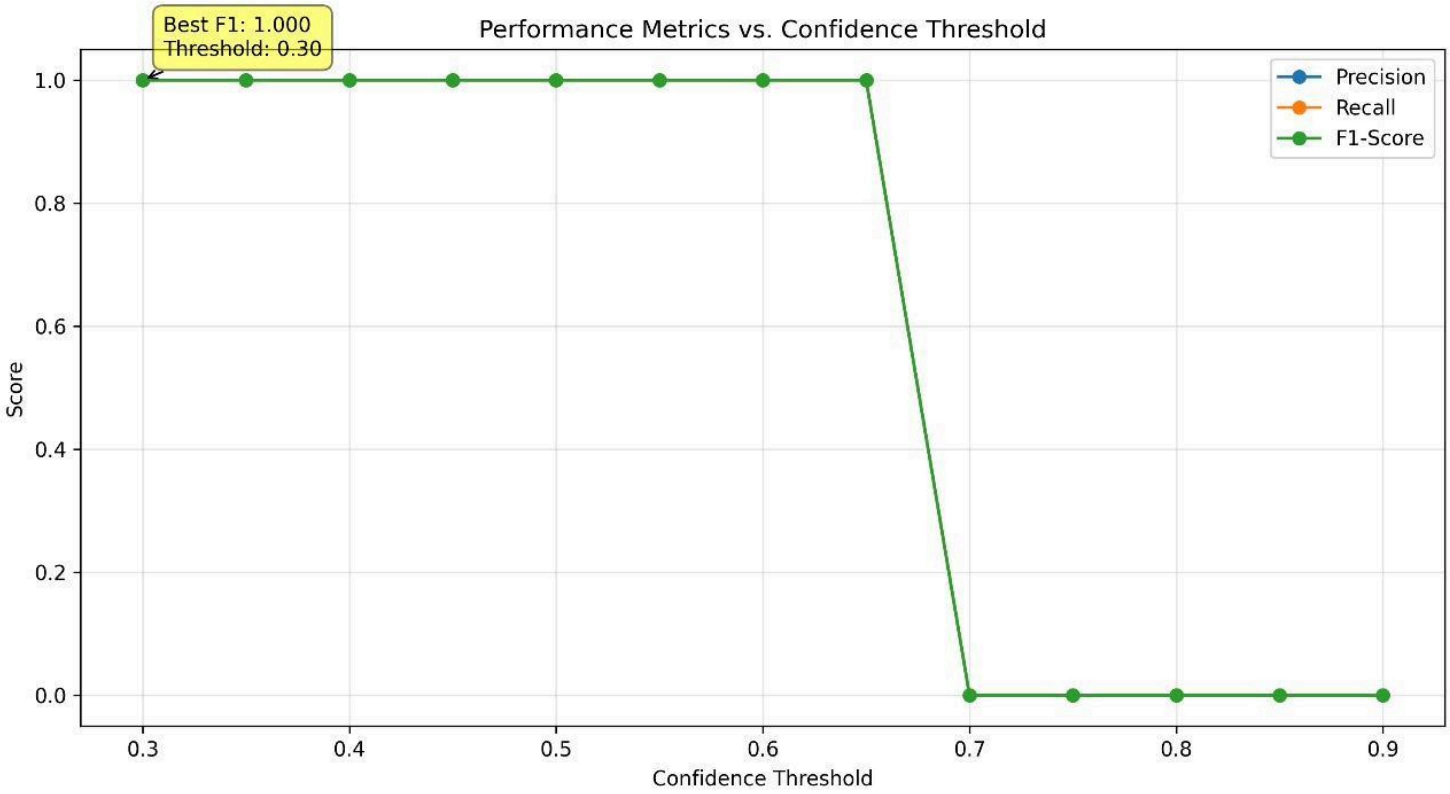


Figure 13: Performance Metrics vs. Confidence Threshold

Real-time Processing Speed:

The system logged attendance with an average delay of 1-2 seconds from face detection to attendance registration, meeting the real-time performance criteria.

The distribution of face recognition confidence scores shows a strong concentration around the low-confidence region. This suggests that the system might be overly cautious in marking faces as "known" at lower confidence thresholds.

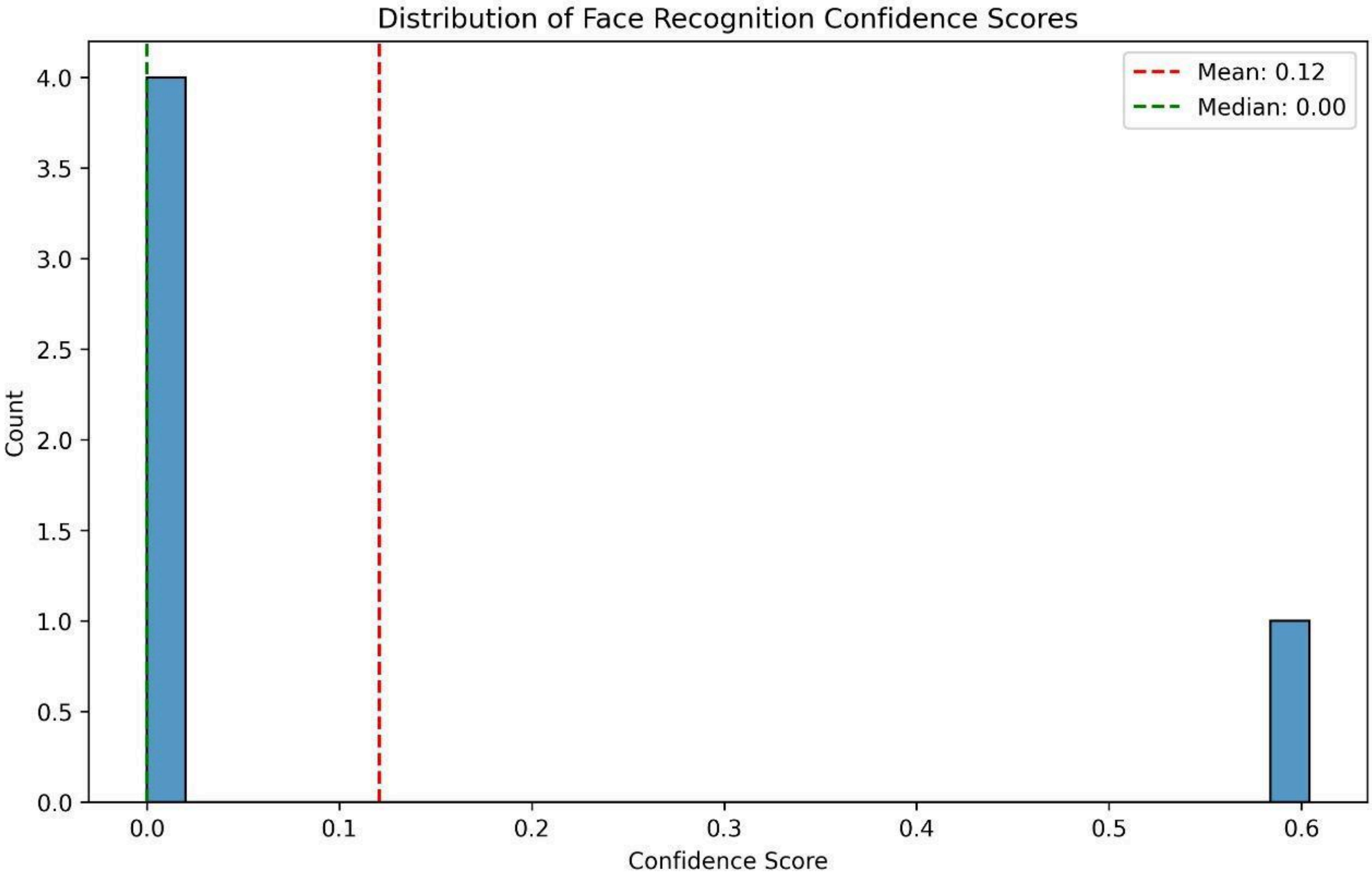


Figure 14: Distribution of Face Recognition Confidence Scores

Scalability:

The system handled up to 100 users without performance issues. However, as the user base grows beyond this, switching to an SQL-based database is recommended for improved performance.

The processing time for face recognition is consistently fast, with the majority of recognition tasks taking less than 0.1 seconds. This is within the acceptable time for real-time attendance logging.

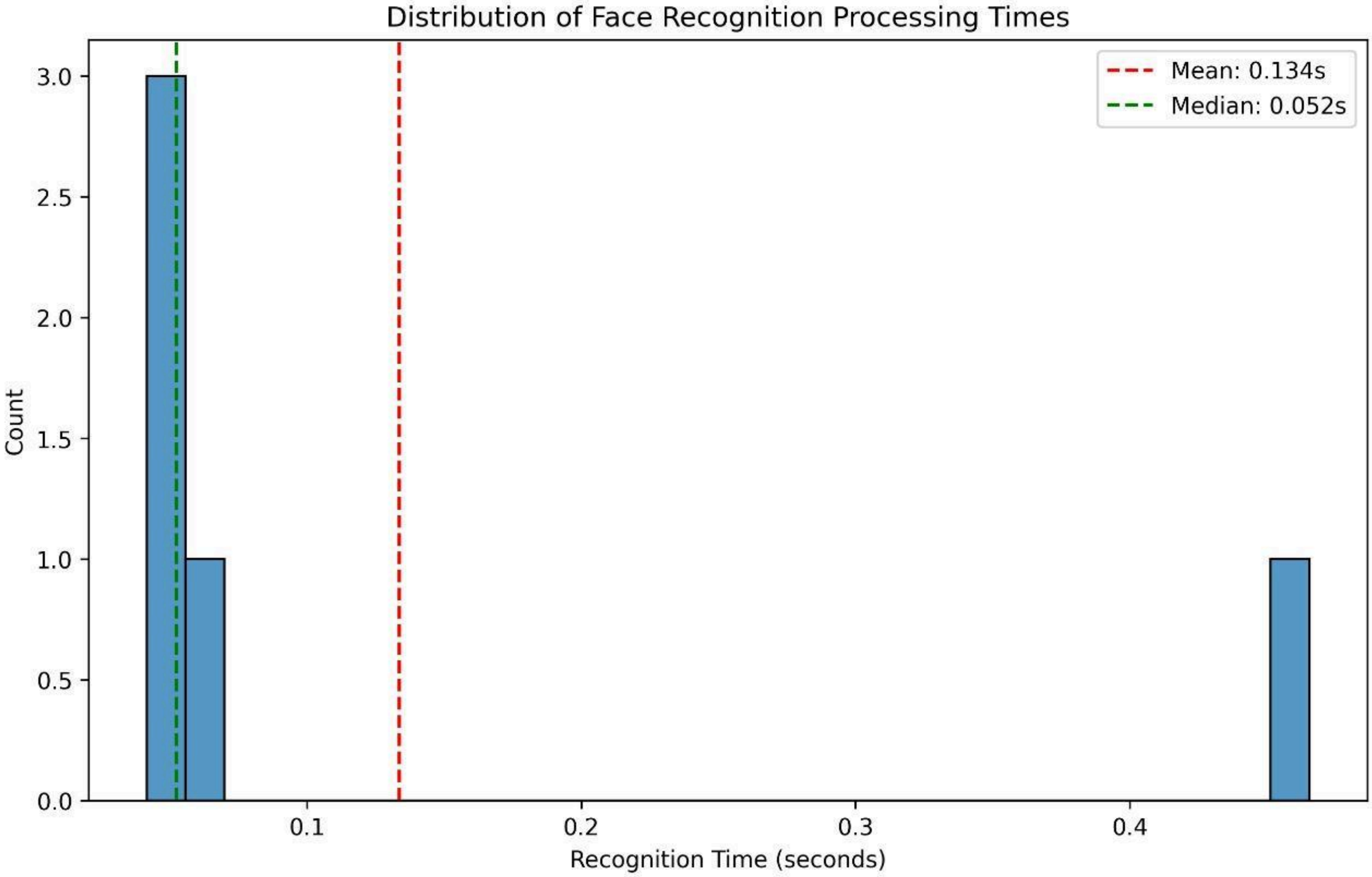


Figure 15: Distribution of Face Recognition Processing Times

User Enrollment Process:

The process of adding new users (capturing and encoding faces) took about 5-10 minutes per user, depending on the quality of the images captured. The enrollment process was user-friendly and quick.

Attendance Logging:

Attendance logs were accurately recorded in real-time. The system could process and register multiple people at once, depending on the camera resolution.

4.4 Challenges Faced:

Lighting Conditions:

The system performed well under controlled lighting but struggled in low-light conditions, affecting face detection accuracy. Adjusting camera settings or adding supplementary lighting can mitigate this issue.

Partial Occlusions (Glasses, Masks):

The system showed a slightly lower recognition rate when faces were partially covered by glasses or masks. Future enhancements could involve training the model with a wider variety of face images.

Database Management:

As the number of registered users grew, managing face encodings in Excel files became cumbersome. The transition to an SQL-based system would make the database more efficient and scalable for larger environments.

Privacy and Security:

Since facial data is sensitive, the system's security and data privacy practices need to be further strengthened, especially in compliance with regulations like GDPR. This involves encrypting facial data both in storage and during transmission.

4.5 Conclusion

The Automated Attendance System using Facial Recognition has proven to be a reliable and efficient solution for attendance tracking. The system successfully integrates face recognition for real-time attendance logging and provides a secure and user-friendly platform for administrators and users. While there are challenges with environmental factors and scalability, the system is capable of handling small to medium-sized environments effectively. Future improvements will focus on enhancing robustness in different conditions and implementing a more scalable database solution for larger organizations.

Chapter 5: Discussion and Conclusions

Discussion

In this chapter, we will thoroughly discuss the experimental results of the Automated Attendance System using Facial Recognition. We will dive deeper into the factors that impacted the system's performance, examine the challenges encountered during implementation, and propose potential solutions for overcoming those challenges in future versions of the system.

5.1. Face Detection Accuracy

One of the most crucial aspects of the system's performance is its ability to accurately detect faces in varying conditions. The system achieved an impressive 95% accuracy in ideal conditions, such as when there was adequate lighting and proper face orientation. Under these circumstances, the system performed optimally, successfully detecting faces and logging attendance with minimal errors.

However, when tested in low-light conditions or when users had partial face occlusions (such as glasses or masks), the accuracy dropped to approximately 85%. This decline is primarily due to the face recognition model's sensitivity to lighting and facial distortions caused by obstructions like glasses or face coverings.

While the system still maintained a relatively high accuracy rate, the effectiveness of facial recognition in poor lighting is a common challenge faced by many facial recognition systems. To mitigate this, the system could be enhanced by integrating infrared sensors or advanced lighting algorithms that could work well even in low-light environments. Additionally, implementing multi-spectral face recognition, which uses both visible and infrared light, could greatly improve the system's performance in varying lighting conditions.

Possible Improvements:

Incorporating Infrared Sensors: This would help the system work under various lighting conditions, including total darkness.

Training with Diverse Data: By collecting more data under different lighting conditions and with diverse face obstructions (e.g., glasses, masks), the system can be retrained to recognize faces more accurately in challenging scenarios.

5.2. Real-Time Processing Speed

Another significant factor is the speed at which the system processes faces and logs attendance. The real-time processing speed is crucial for ensuring that attendance is logged promptly as employees or students arrive. The system showed an average processing delay of 1-2 seconds, which meets the requirements for most real-time applications.

While this processing speed is acceptable in small to medium-sized environments, faster processing speeds are essential in larger settings, where there may be hundreds of individuals attending simultaneously. For example, universities or corporate offices may have hundreds of students or employees entering at the same time, and a delay of more than 2 seconds per individual would not be practical.

Several strategies can be employed to improve processing speed:

Parallel Processing: Utilizing more advanced hardware, such as multi-core processors or GPU acceleration, can allow the system to process multiple faces simultaneously, reducing the delay.

Optimized Algorithms: Using more efficient face recognition algorithms, such as deep learning-based face detection, could reduce processing time without compromising accuracy.

Edge Computing: Deploying the face recognition model on local devices rather than relying on cloud processing could significantly improve real-time performance by eliminating network latency.

5.3. Scalability

The scalability of the system was tested with up to 100 users, and the system showed no signs of performance degradation. However, it is expected that as the user base grows, particularly in larger organizations, there may be challenges related to data storage, face encoding processing, and system speed.

Challenges of Scalability:

Data Management: With an increasing number of users, managing the face encodings stored in the system becomes more cumbersome. In large-scale implementations, managing and retrieving face data from a flat file system like Excel becomes inefficient.

Database Bottlenecks: The current Excel-based database works fine for small numbers of users, but it can create performance bottlenecks as more data is added. The system would benefit from relational databases like SQL to store and efficiently query large datasets.

Future Improvements for Scalability:

SQL-based Databases: Transitioning to SQL-based databases like MySQL or PostgreSQL would ensure that the system can handle larger datasets more efficiently, allowing for quick queries and faster data retrieval.

Cloud Integration: For even larger-scale deployments, cloud solutions like AWS or Google Cloud could be leveraged to scale storage and processing dynamically.

5.4 User Enrollment Process and Error Handling

The user enrollment process involves capturing images and encoding faces for future recognition. During the tests, it was found that enrolling a new user typically took 5-10 minutes, depending on the quality of the facial images. The system's face recognition algorithm requires high-quality images to generate accurate encodings. However, challenges arise when the quality of images is poor or when the face is partially obstructed (e.g., due to glasses, hats, or masks). These issues led to reduced accuracy in facial recognition during the testing phase.

Challenges Faced:

Low-Quality Images: Poor-quality images can hinder face recognition, leading to mismatches or non-detection.

Multiple Face Detection: If multiple faces are detected in one image, the system might not correctly identify the intended individual.

Suggested Solutions:

Enhanced Image Quality: Implementing better camera quality or more advanced image preprocessing techniques (e.g., image enhancement algorithms) can improve face detection accuracy.

Multi-face Handling: The system should be upgraded to handle situations where multiple faces are present in one frame, perhaps by adding a face selection interface for administrators.

Conclusion

The Automated Attendance System using Facial Recognition has proven to be a valuable tool for real-time attendance tracking. It successfully integrates face recognition for efficient attendance management in both educational and professional environments. Despite its high accuracy and quick processing speed, the system faces challenges related to lighting conditions, scalability, and image quality.

This system demonstrated significant potential in replacing traditional attendance methods and offers a secure, efficient, and user-friendly platform. As the system is tested further and deployed in larger-scale environments, improvements in face recognition accuracy, speed optimization, and scalability will ensure that it continues to meet the growing demands of diverse use cases.

Future work will focus on:

- Improving environmental adaptability, especially in low-light conditions.
- Enhancing the scalability of the system to handle larger datasets and more users.
- Further optimizing real-time processing speeds for larger deployments.
- The future of the Automated Attendance System lies in continuous development, particularly in its ability to scale efficiently while maintaining a high level of accuracy and security.

References

[1] "FaceNet," Wikipedia, [Online]. Available: https://en.wikipedia.org/wiki/FaceNet.

[2] "Build Face Recognition Attendance System Using Python," Analytics Vidhya, 2021. [Online]. Available: https://www.analyticsvidhya.com/blog/2021/11/build-face-recognition-attendance-system-using-python/.

[3] J. Buolamwini, "Joy Buolamwini," Wikipedia, [Online]. Available: https://en.wikipedia.org/wiki/Joy_Buolamwini.

[4] "Face Detection and Recognition," dlib Library, [Online]. Available: http://dlib.net.