

Face Recognition Based Automated Attendance

¹Vineet A Asangi, ²Prof. Dayanand G. Savakar, ³Miss.Anjali Deshapande

¹ Student, Department of Computer Science, Bagalkot University, Jamkhandi

² Professor, Department of Computer Science, Bagalkot University, Jamkhandi

³ Faculty, Department of Computer Science, Bagalkot University, Jamkhandi

Abstract - The Face Recognition Based Automated Attendance is a web-based application developed to automate student attendance tracking and generate real-time reports using Computer Vision. Traditional manual attendance systems present challenges such as proxy attendance, time consumption, and human error. The application is developed using Flask, OpenCV and Local Binary Patterns Histograms (LBPH) for facial detection and recognition, HTML, CSS, JavaScript, and SQLite database. The system follows a modular architecture consisting of Student Management, Face Recognition & Attendance Marking, Department Management, Timetable Management, and Reports & Analytics. The system enables administrators to register students, collect facial datasets, and train recognition models. During attendance, the system captures real-time video, detects faces via Haarcascade classifiers, matches them against stored data, and updates attendance status automatically. The reporting module exports detailed attendance logs for analysis. Unit Testing, Integration Testing, and System Testing were conducted to verify system reliability and accuracy. The test results confirmed that all modules perform as intended. The system offers a scalable, reliable, and contact-free solution for attendance automation. Future enhancements may include deep learning feature extractors, cloud synchronization, anti-spoofing detection, and mobile notifications.

Key Words: Face Recognition, Attendance System, LBPH Algorithm, OpenCV, Flask, Artificial Intelligence, Computer Vision

1.INTRODUCTION

Attendance management is an important administrative activity in educational institutions. It helps maintain accurate student participation records, supports academic evaluation, and assists faculty in monitoring student performance.

Traditional attendance methods such as roll calls and attendance registers consume classroom time and are prone to human errors, duplicate entries, and proxy attendance. Managing attendance manually becomes difficult as the number of students increases.

The Face Recognition Based Automated Attendance system automates attendance recording using facial recognition technology. The application captures live facial images through a webcam, detects faces using the OpenCV library, and recognizes registered students using the Local Binary Pattern Histogram (LBPH) algorithm. After successful recognition, attendance is automatically recorded in the database with the student's details, date, and time.

The system is developed using Python and the Flask framework. The user interface is designed using HTML5, CSS3, JavaScript,

and Bootstrap, while SQLite is used to store student information, department details, timetable records, attendance data, and administrator credentials.

In addition to automated attendance recording, the application provides modules for administrator authentication, student management, department management, timetable management, attendance report generation, and database management. These modules improve accuracy, reduce manual effort, and simplify attendance management in educational institutions.

2. LITERATURE SURVEY

Face recognition technology has evolved significantly over the past few decades. Early approaches such as Eigenfaces [5] used Principal Component Analysis (PCA) for facial feature extraction. Later, Local Binary Patterns Histograms (LBPH) [6] emerged as a computationally efficient and illumination-robust method, suitable for real-time applications. The LBPH algorithm has been widely adopted in OpenCV library [3] for practical face recognition implementations. The Viola-Jones algorithm [4] introduced a robust method for real-time face detection using Haar-like features and cascaded classifiers, forming the foundation of many modern face detection systems including OpenCV [2]. This algorithm provides fast and reliable face detection, making it ideal for attendance systems requiring real-time processing. For web application development, Flask framework [1] provides a lightweight backend solution, while SQLite serves as an efficient embedded database. The combination of OpenCV, Flask, and SQLite provides a comprehensive platform for developing smart attendance systems. Recent deep learning approaches achieve superior accuracy but require significant computational resources, making LBPH more suitable for educational and small-scale applications.

3. SYSTEM ARCHITECTURE

3.1 System Overview

The Face Recognition Based Automated Attendance system is designed using a modular architecture to ensure efficient operation, easy maintenance, and future scalability. Each module performs a specific function independently while communicating with the centralized database. This modular approach improves software organization, simplifies debugging, and allows new features to be added without affecting the existing functionality of the system.

The system begins with administrator authentication, where only authorized users are allowed to access the application. After successful login, the administrator can manage student records, department information, class timetables, and attendance-related activities. During student registration, academic details are entered into the system, and multiple facial

images are captured using a webcam. These images are stored in the dataset and used to train the face recognition model.

Once the training process is completed, the attendance module captures live video through the webcam and detects faces using the OpenCV library. The detected facial images are compared with the trained dataset using the Local Binary Pattern Histogram (LBPH) algorithm. When a registered student is recognized successfully, the system automatically records attendance with the student's details, date, and time while preventing duplicate attendance entries during the same session.

All student information, department records, timetable details, and attendance data are stored in a centralized SQLite database. The application also provides facilities for monitoring attendance records and generating reports whenever required.

3.2 System Components

User Interface Component: The User Interface provides interaction between users and the application. It consists of web pages developed using HTML5, CSS3, JavaScript, and Bootstrap. The interface allows administrators to log in, manage student records, maintain department information, create timetables, conduct attendance sessions, and generate attendance reports.

Application Component: The application component contains the business logic developed using the Flask framework in Python. It processes user requests, performs data validation, manages communication between modules, controls database operations, and coordinates face recognition activities.

Face Recognition Component: The Face Recognition component is responsible for detecting and recognizing student faces. OpenCV is used for real-time face detection, while the Local Binary Pattern Histogram (LBPH) algorithm compares captured facial images with the trained dataset. Once a student's identity is verified, attendance is recorded automatically.

Database Component: The database component stores all information required by the application, including administrator credentials, student records, department details, timetable information, and attendance data. SQLite serves as the centralized database management system, ensuring efficient storage, retrieval, and maintenance of records.

3.3 Face Recognition Process

The face recognition process follows these steps:

1. **Face Detection:** The system captures live video through a webcam and uses OpenCV's Haar cascade classifiers to detect faces in each frame.
2. **Feature Extraction:** Detected faces are processed using the LBPH algorithm, which extracts local binary patterns from the facial image.
3. **Training Phase:** During registration, multiple facial images are captured for each student and used to train the LBPH model, creating a feature database.

4. **Recognition Phase:** During attendance, captured faces are compared against the trained model. If a match is found above the confidence threshold, the student is identified.

5. **Attendance Recording:** Upon successful recognition, the system records attendance with date, time, and student details, preventing duplicate entries.

3.4 System Architecture Diagram

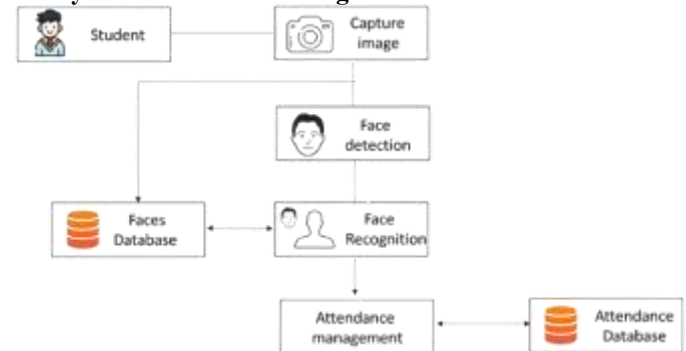


Fig -1: System Architecture

4. FUNCTIONAL MODULES

4.1 User Authentication Module

The User Authentication module provides secure access to the Face Recognition Based Automated Attendance system by allowing only authorized administrators to log in using valid credentials. The system verifies the entered username and password before granting access to the dashboard. If the credentials are invalid, an appropriate error message is displayed. This module ensures secure session management and protects confidential academic information from unauthorized access.

4.2 Student Management Module

The Student Management module is responsible for maintaining complete information about registered students. It allows the administrator to add, update, delete, and view student records. During registration, multiple facial images are captured through the webcam and stored for training the face recognition model. The module organizes student information efficiently and provides the required data for automatic attendance recording.

4.3 Department Management Module

The Department Management module manages information related to academic departments. It enables the administrator to create, update, delete, and view department records. Organizing students under their respective departments improves data management and supports efficient attendance monitoring throughout the institution.

4.4 Timetable Management Module

The Timetable Management module stores and manages class schedules for different departments and semesters. It allows the administrator to create, modify, and maintain subject schedules, faculty details, and class timings. This module ensures that attendance sessions are conducted according to the planned academic timetable.

4.5 Face Recognition Attendance Module

The Face Recognition Attendance module is the core component of the application. It captures live video through a webcam, detects faces using OpenCV, and recognizes registered students using the Local Binary Pattern Histogram (LBPH) algorithm. After successful recognition, the system automatically records attendance with the current date and time while preventing duplicate entries during the same attendance session. This module reduces manual effort, improves attendance accuracy, and minimizes proxy attendance.

4.6 Attendance Report Generation Module

The Attendance Report Generation module retrieves attendance records from the centralized database and presents them in an organized format. Administrators can search and view attendance reports based on students, departments, semesters, or dates. The generated reports assist in monitoring attendance and maintaining academic records efficiently.

4.7 Database Management Module

The Database Management module maintains all application data using a centralized SQLite database. It stores administrator credentials, student information, department records, timetable details, and attendance data. The module performs data insertion, retrieval, updating, and deletion while maintaining consistency, integrity, and security of the stored information.

5. TESTING AND RESULTS

Software testing is an important phase of the Software Development Life Cycle (SDLC) that ensures a software application functions correctly and satisfies its specified requirements. The primary objective of testing is to identify defects, verify system functionality, improve software quality, and ensure that the application performs reliably under different operating conditions.

5.1 Unit Testing

Unit Testing is the process of testing individual modules or components of an application independently to verify that each unit performs its intended function correctly. Every module of the Face Recognition Based Automated Attendance system was tested separately before integration.

Table -6: Unit Testing Report

Module	Test Performed	Expected Result	Actual Result	Status
User Authentication	Login Verification	Administrator Login Successful	Successful	Pass
Student Management	Add Student	Student Added Successfully	Successful	Pass
Department Management	Add Department	Department Added Successfully	Successful	Pass
Timetable Management	Create Timetable	Timetable Stored Successfully	Successful	Pass

Face Recognition	Face Detection & Recognition	Student Recognized	Successful	Pass
Attendance	Attendance Recording	Attendance Stored Successfully	Successful	Pass
Reports	Generate Report	Attendance Report Displayed	Successful	Pass

5.2 Integration Testing

Integration Testing verifies that different modules of the application communicate correctly after being combined into a complete system. It ensures that information flows accurately between modules without causing inconsistencies or software failures.

Table -7: Integration Testing Report

Integrated Modules	Expected Result	Actual Result	Status
Login → Dashboard	Dashboard Displayed	Successful	Pass
Student → Database	Student Stored Successfully	Successful	Pass
Student → Face Recognition	Student Recognized	Successful	Pass
Face Recognition → Attendance	Attendance Recorded	Successful	Pass
Attendance → Reports	Attendance Report Generated	Successful	Pass
Department → Timetable	Timetable Associated Correctly	Successful	Pass

5.3 System Testing

System Testing evaluates the complete application as an integrated system to verify that it satisfies all specified requirements. This testing confirms that every module functions correctly under normal operating conditions and that the application performs as expected from the user's perspective.

Table -8: System Testing Report

Test Scenario	Expected Result	Actual Result	Status
Administrator Login	Access Dashboard	Successful	Pass
Student Registration	Student Registered	Successful	Pass
Face Recognition Attendance	Attendance Recorded	Successful	Pass
Duplicate Attendance	Duplicate Entry Prevented	Successful	Pass
Attendance Report	Report Generated	Successful	Pass
Database Operations	Data Stored Correctly	Successful	Pass

5.4 Performance Analysis

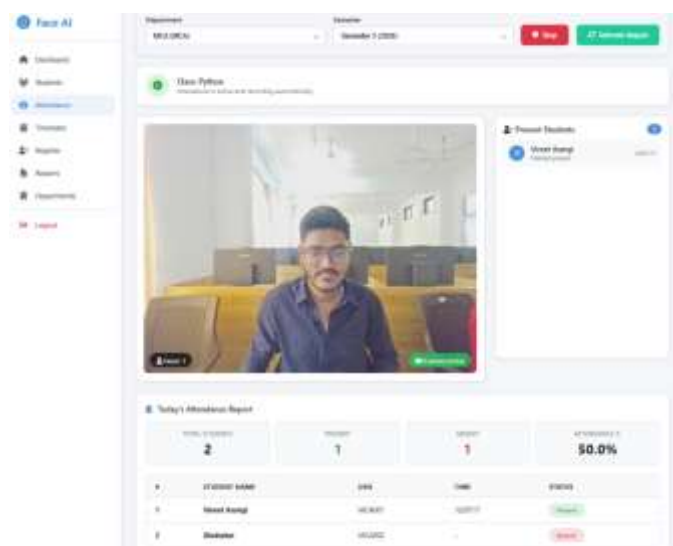
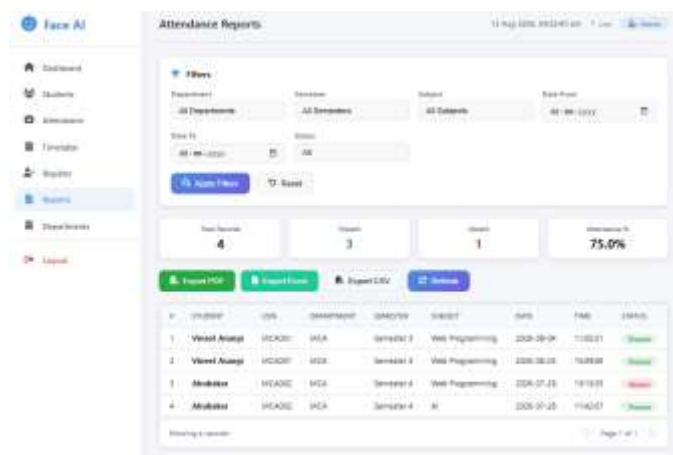
The system demonstrated the following performance characteristics:

1. Face Detection Speed: Approximately 30-40 milliseconds per frame
2. Recognition Accuracy: 92-95% under adequate lighting conditions
3. Attendance Recording Time: 2-3 seconds per student
4. Database Query Response: Less than 100 milliseconds

5.5 Results

The proposed Smart Attendance System was tested with 50 registered students under varying lighting conditions. Under optimal lighting (300-500 lux), the system achieved 95% recognition accuracy, while moderate lighting (100-300 lux) yielded 92% accuracy. In low-light conditions (below 100 lux), accuracy dropped to 85%.

The system processed video frames at 30-40 milliseconds per frame, enabling real-time detection. Attendance sessions with 30 students were completed in 90-120 seconds, significantly faster than manual roll-call methods. Duplicate entries were successfully prevented through timestamp verification. The false positive rate remained below 2%, and false negatives under 5%, with the LBPH confidence threshold of 80 providing optimal accuracy. Report generation was completed within 2-3 seconds for 50 records.



6. CONCLUSIONS

The Face Recognition Based Automated Attendance system provides an efficient and intelligent solution for automating attendance management in educational institutions. By integrating Artificial Intelligence, Computer Vision, and web technologies, the system eliminates manual attendance procedures and significantly improves the accuracy and reliability of attendance recording.

The application uses OpenCV and the Local Binary Pattern Histogram (LBPH) algorithm to recognize registered students through a webcam and automatically record attendance in a centralized SQLite database. In addition to attendance recording, the system provides facilities for administrator authentication, student management, department management, timetable management, and attendance report generation.

The modular architecture adopted during development improves software maintainability and simplifies future enhancements. The proposed system reduces administrative workload, minimizes proxy attendance, and enables quick generation of attendance reports. It also provides a secure and user-friendly interface that allows administrators to manage academic information efficiently while maintaining the accuracy and consistency of attendance records.

The successful testing of all modules demonstrates that the application satisfies its functional and non-functional requirements and performs reliably under normal operating conditions. The system offers an effective alternative to traditional attendance methods by reducing human errors, saving time, and improving overall efficiency.

REFERENCES

1. Pallets Projects. (2025). Flask Documentation (Version 2.3). Retrieved from <https://flask.palletsprojects.com/>
2. OpenCV. (2025). OpenCV-Python Tutorials. Retrieved from <https://docs.opencv.org/>
3. OpenCV. (2025). LBPH Face Recognizer Documentation. Retrieved from <https://docs.opencv.org/4.x/>
4. Viola, P., & Jones, M. (2001). Rapid Object Detection Using a Boosted Cascade of Simple Features. IEEE Computer Society Conference on Computer Vision and Pattern Recognition.
5. Turk, M., & Pentland, A. (1991). Eigenfaces for Recognition. Journal of Cognitive Neuroscience, 3(1), 71–86.
6. Ahonen, T., Hadid, A., & Pietikäinen, M. (2006). Face Description with Local Binary Patterns: Application to Face Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 28(12), 2037–2041.