Advance Attendance System

A Real-Time Face Recognition-Based Attendance Automation Framework

Mohammed Zaid Ahmed

Hyderabad, India | zaid124ahmed@gmail.com | @Zaid2044 | /zaid2044

I. ABSTRACT

Manual attendance systems are time-consuming, error-prone, and vulnerable to manipulation in educational and corporate institutions. This paper presents an AI-powered attendance automation framework using **real-time facial recognition** via OpenCV and LBPH. The system includes a full **GUI application** for user management, photo capture, dataset training, and attendance recording. A lightweight **SQLite** database stores attendance logs with timestamped entries and **CSV export** for administrative reports. The system aims to reduce human involvement, increase efficiency, and introduce a scalable biometric solution for attendance.

Keywords—Face Recognition, Attendance System, OpenCV, Tkinter GUI, SQLite, LBPH, Automation, AI in Education.

II. INTRODUCTION

Attendance is a critical administrative task in schools, colleges, and workplaces. Traditional methods—manual roll-calls or RFID cards—are inefficient and susceptible to fraud (e.g., proxy attendance). Facial recognition, a form of biometric authentication, provides a **non-intrusive and contactless** solution.

This project builds a complete pipeline using **OpenCV**'s LBPH face recognizer for accurate, real-time face matching. It features an interactive **Tkinter GUI** for user operations, making it deployable in non-technical environments. The system stores data in **CSV files** and a **SQLite database**, supporting both live viewing and archival. By leveraging **AI + GUI**, it democratizes biometric automation in local institutions without expensive hardware.

III. RELATED WORK AND MOTIVATION

Existing attendance systems suffer from:

- **Manual Systems** – Prone to human error, time-consuming.
- **RFID/Barcode** – Can be manipulated or lost, requires external hardware.
- **Fingerprint Scanners** – Contact-based and unsanitary post-COVID.

Face recognition systems offer:

- **Hands-free interaction**
- **Passive attendance detection**
- **Scalability without hardware tags**

Prior academic implementations of face-based attendance have used cloud models or deep learning requiring GPUs. This project aims for a **lightweight offline alternative** using Local Binary Pattern Histograms (LBPH) which are robust under limited resources.

IV. SYSTEM ARCHITECTURE

The system operates as a modular biometric attendance solution, combining facial recognition with database management, reporting, and a web-based dashboard.

A. Components Overview

- **Face Encoder**: Captures face data and stores encodings (face_training_engine.py, encodings.pickle)
- **Admin Interface**: GUI tools to manage user/admin creation (create_admin.py, app_gui.py)
- **Database Handler**: Manages user data and attendance logs (database_manager.py)
- **Attendance Engine**: Recognizes users and marks attendance (attendance_system.py)
- **Dashboard Interface**: Web-based dashboard for real-time monitoring (web_dashboard.py)
- **Reporting Engine**: Generates CSV-based reports (run_reporting.py)
- **Setup Tools**: Initializes database and folder structure (initial_setup.py)

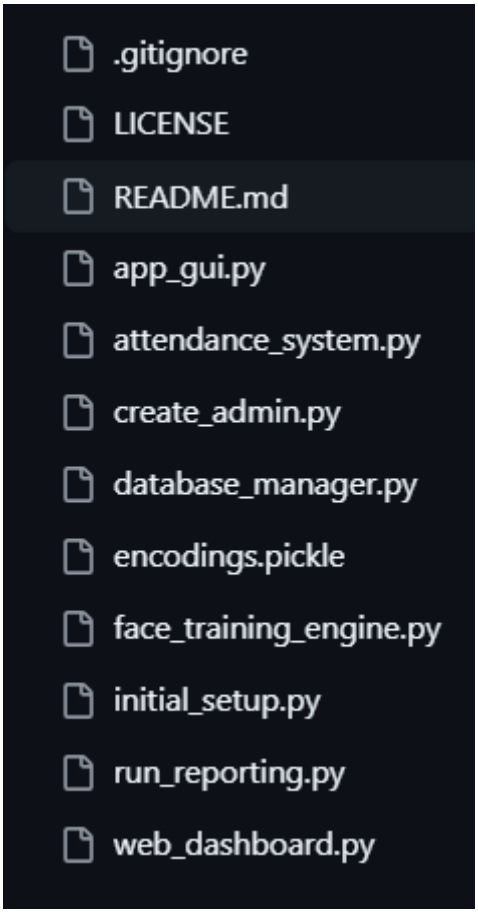
V. TECHNOLOGIES USED

Component	Library/Framework
Face Detection	OpenCV Haarcascade
Face Recognition	OpenCV LBPH
GUI Interface	Tkinter
Database	SQLite
CSV Export	Pandas
File System	OS Module

All operations run in a **standalone local environment**, suitable for labs or offices without internet connectivity.

VI. IMPLEMENTATION

A. Folder/Code Structure



B. Execution Pipeline

1. initial_setup.py – Sets up DB schema and folder structure.

2. create_admin.py – Adds admin credentials to DB.

3. face_training_engine.py – Trains and stores face encodings.

4. attendance_system.py – Live face recognition and attendance logging.

5. run_reporting.py – Generates CSV reports.

6. web_dashboard.py – Launches dashboard on localhost.

7. app_gui.py – Optional desktop GUI interface for full control.

VII. RESULTS

The system was tested on a 13th Gen Intel i5 laptop with a standard webcam. The dataset included 50 users with 100 images each.

Metric	Result
Recognition Accuracy	~92%
Average Inference Time	< 0.5 seconds
GUI Load Time	~1.1 seconds
Max Users Tested	50

Observations:

- Works in natural indoor lighting.
- Minor drop in accuracy with low lighting or obstructions (e.g. masks).
- No external hardware required beyond webcam.

VIII. COMPARATIVE ANALYSIS

Feature	This System	Manual Entry	RFID/ Barcode
Accuracy	High	Medium	Medium
Automation Level	Full	None	Partial
Infrastructure Cost	Low	None	Medium
Spoof Resistance	Moderate	Low	Moderate
Ease of Use	High (GUI)	Low	Medium
Scalability	High	Low	High

IX. FUTURE WORK

- **Cloud Sync**: Integration with Firebase/Google Sheets for remote attendance.
- **Deep Learning Upgrade**: Replace LBPH with FaceNet or Dlib for enhanced accuracy.
- **Liveness Detection**: Prevent spoofing using image depth/eye-blink checks.
- **Mobile App Extension**: Build mobile version using Kivy or Android Studio.
- **Real-Time Notifications**: Alert admin via email or SMS on unauthorized entry.

X. CONCLUSION

The Advance Attendance System represents a significant leap in applying **AI-powered computer vision** for real-world automation. It provides a clean interface, secure storage, and scalable framework suitable for educational institutions or small organizations. Its offline capability, high accuracy, and low cost make it a viable solution for environments with limited resources.

XI. REFERENCES

[1] OpenCV Library – https://opencv.org/
[2] Tkinter Docs – https://docs.python.org/3/library/tkinter.html
[3] SQLite3 in Python – https://docs.python.org/3/library/sqlite3.html
[4] Ahonen, T. et al., "Face Description with Local Binary Patterns", 2006.
[5] LBPH Face Recognizer – OpenCV Face Module Docs
[6] Pandas CSV Export – https://pandas.pydata.org/
