Automated Attendance System Using Face Recognition

Project submitted to the

SRM University – AP, Andhra Pradesh

for the partial fulfillment of the requirements to award the degree of

Bachelor of Technology

In

Computer Science and Engineering

School of Engineering and Sciences

Submitted By

Damera Aravind | AP23110011645

Vuyyuru Shruthi Deepika | AP23110011584

Peddoju Durga Sravanthi| AP23110011597

Kilari Joshitha | AP23110011641

Under the Guidance of

Ms. V. Veda Sri

Lecturer, Department of CSE

SRM University–AP

Neerukonda, Mangalagiri, Guntur

Andhra Pradesh – 522 240

[April, 2025]

Certificate

Date: _________

This is to certify that the work present in this Project entitled "**Automated Attendance System Using Face Recognition**" has been carried out by **[Damera Aravind,Vuyyuru Shruthi Deepika, Peddoju Durga Sravanthi, Kilari Joshitha]**under my/our supervision. The work is genuine, original, and suitable for submission to the SRM University – AP for the award of Bachelor of Technology/Master of Technology in **School of Engineering and Sciences**.

Supervisor

Kavitha Rani

Designation,

Affiliation.

Acknowledgement

The satisfaction that accompanies the successful completion of any task would be incomplete without introducing the people who made it possible and whose constant guidance and encouragement crowns all efforts with success.

I am extremely grateful and express my profound gratitude and indebtedness to my project guide, **Kavitha Rani**, Department of Computer Science & Engineering, SRM University, Andhra pradesh, for her kind help and for giving me the necessary guidance and valuable suggestions in completing this project work.

Damera Aravind - AP23110011645

Vuyyuru Shruthi Deepika - AP23110011584

Peddoju Durga Sravanthi - AP23110011597

Kilari Joshitha - AP23110011641

Table Of Contents

Abstract

An clever and creative way to transform conventional attendance tracking techniques in educational institutions is the Automated Attendance System Using Face Recognition. This solution gets rid of the errors and inefficiencies that come with manual operations by utilizing cloud computing, machine learning, and computer vision. Constructed with Python, the system incorporates essential libraries like SQLite3 for local data storage, NumPy and pandas for effective data processing and analysis, and OpenCV for real-time face detection and recognition. Supabase is also utilized as a cloud backend to safely and remotely monitor attendance information and store student photos.

Using a live camera stream, the system records faces, instantly recognizes students using facial recognition software, and logs their attendance with a timestamp. Faculty can view attendance records, update student data, and create reports using an intuitive Graphical User Interface (GUI). The automated email alert system, which informs students of their absences and promotes prompt attendance and openness, is one of the noteworthy characteristics.

This project demonstrates how many technologies can be practically integrated to address actual administrative issues in the field of education. In addition to saving time, it enhances the accuracy and integrity of data by reducing human intervention. The system is a strong foundation for next-generation attendance tracking since it is secure, scalable, and adjustable to institutions of any size. It provides a real-world illustration of how cloud services and AI might be integrated to produce intelligent, effective, and significant solutions.

1.Introduction

1.1 Context

Traditional attendance tracking methods, such as manual entry or RFID-based systems, are time-consuming, vulnerable to human error, and often ineffective at preventing dishonest practices like proxy attendance. These methods also lack real-time monitoring and integration with digital systems. As educational institutions move towards smart and data-driven solutions, there is a growing need for an **automated, secure, and cloud-integrated attendance system** that ensures accuracy, transparency, and efficiency.

1.1.1 Statement of the Problem

The existing methods of attendance management face several limitations:

- Inability to efficiently manage large groups of students.
- Lack of real-time updates and instant reporting.
- High risk of manipulation through proxy attendance or false entries.

To overcome these issues, this project introduces an **Automated Attendance System** that uses **facial recognition technology** and **cloud computing** to provide a reliable and scalable solution.

1.2 Goals

The primary objectives of this project are:

- To develop a **real-time face detection and recognition system** using the face_recognition library and **OpenCV**.
- To design a **graphical user interface (GUI)** for teachers to manage student data and view attendance records.
- To implement **automated daily reports** and **email alerts** for absent students.
- To securely store student face data and attendance logs in a **Supabase cloud database**.

1.3 Importance

This system offers several benefits over traditional methods:

- **Efficiency**: Reduces manual attendance time by up to 80%.
- **Security**: Minimizes the chances of proxy attendance through biometric validation.
- **Scalability**: Utilizes cloud storage to handle unlimited student data.
- **Accessibility**: Provides teachers and administrators access to all records through a unified dashboard.

2.Methodology

2.1 Frontend: Real-Time GUI and Camera Feed

The frontend component of the system is responsible for handling the real-time video stream from the webcam and managing user interactions through a graphical user interface (GUI). It plays a crucial role in delivering an interactive and user-friendly experience for educators.

Live Video Stream

- The camera continuously captures video frames, which are processed in real-time using OpenCV.
- Each frame is downscaled to improve system performance and reduce latency.
- Face detection and recognition are applied on each frame to identify students.

Graphical User Interface (GUI)

- The GUI is developed using Tkinter, Python's standard GUI library.
- It provides an intuitive interface for teachers and administrators to:
    - Add new students and upload their photographs for training the recognition model.
    - View and manage attendance records, with options to sort or filter by date or student.
    - Start or stop attendance recording during class sessions.
    - Export attendance data in readable formats such as CSV or Excel.

Visual Feedback

- For better usability and transparency, the system provides real-time visual feedback by:
    - Overlaying bounding boxes around detected faces.
    - Displaying the recognized student's name directly on the live camera feed.
    - Highlighting unknown or unregistered faces for review.

2.2 Backend: Email Logic, Face Encoding, Detection, and Matching

Face encoding, real-time face identification, matching logic, updating attendance records, and email notifications are all handled by the backend, which is the central component of the system. Accurate processing, effective data management, and safe connectivity with the cloud database are all guaranteed by this component.

Pipeline for Face Recognition

1. Acquisition of Images

- Face photos are submitted to Supabase Storage during the student registration process and arranged in an images bucket.
- To maintain uniformity and facilitate retrieval, each photograph is given a unique name based on the student's registration number.

2. Generation of Encoding

- OpenCV is used to read images and transform them from BGR to RGB color representation.
- The Histogram of Oriented Gradients (HOG) model is used to identify facial structures.
- Face_recognition.face_encodings() is used to create a 128-dimensional facial embedding for every image.
- Python's pickle module is used to serialize and save these encodings locally in a file called EncodeFile.p for quicker runtime loading.

3. Instantaneous Matching

- To save computation and increase speed, webcam live frames are shrunk to 25% of their original size.
- Face_recognition.face_locations() is used to find the locations of faces in the current frame.
- The Euclidean distance determined by face_recognition is used to encode each detected face and compare it with the stored encodings. face_distance().
- To balance accuracy and false positives, a face is deemed a match if the distance is less than a predetermined threshold, usually 0.6.

Logic for Attendance Marking After a successful match:

- To avoid duplicate entries, the system verifies the student's most recent attendance timestamp.
- The student's attendance is re-recorded if sufficient time has elapsed (for example, more than five seconds or a new day).
- The Supabase database is updated in real-time with the most recent attendance time as well as the overall number of attendees.

2.3. Integration of Databases with Supabase

Supabase, which provides safe access, scalability, and real-time updates, serves as the cloud-based backend for storing student data and photographs.

Students table in supabase:

- Name, major, registration number, total attendance, and last attendance time are fields in the Students Table (Supabase).
- A comprehensive record is created by linking each student's facial image with their registration_number.
- makes it possible to handle attendance logs and retrieve data efficiently.

Images Storage (pictures bucket)

- uses the registration_number as a unique identifier to store facial photographs.
- permits remote access and facilitates validation or retraining of the recognition model in the future.

2.4 Email Notification System

To ensure efficient communication between instructors and students, the system has an integrated email service.

Daily Attendance Reporting (report_sender.py):

- Pandas is used to retrieve attendance data, which is subsequently transformed into Excel (.xlsx) format.
- Timestamps are formatted into date and time that can be read by humans.
- After being locally saved, the Excel file is attached to an email.

Email Delivery:

Gmail's SMTP and app passwords are used in conjunction with the smtplib module to securely send emails. Faculty members receive daily attendance data. Students who miss more than three days in a row will receive an email alert from the system in subsequent updates.

3. Implementation

3.1 Tools and Libraries:

Tool	Purpose
OpenCV	Video capture, frame processing, and GUI rendering.
face_recognition	Face detection, encoding generation, and matching.
Supabase	Cloud storage for student data and images.
Pandas	Generating Excel reports for email attachments.
smtplib	Sending emails with attendance reports.

3.2 Process

Phase of Registration
Students' photos are uploaded by teachers to the pictures folder. Upload.py uploads the student's metadata (name, major, etc.) to the Supabase students table after encodegenator.py processes these photos, creates 128-dimensional face encodings, and maps them to each student's registration_number.
Attendance Phase main1.py uses the webcam to record live video, identify faces, and compare them to EncodeFile.p's saved encodings. The GUI shows the student's information and Supabase updates the attendance if a match is identified and 60 seconds have elapsed since the last mark. "No Data Found" is displayed if there is no match.
Phase of Reporting
Following attendance, report_sender.py retrieves Supabase's attendance information, uses pandas to generate an Excel report, then uses smtplib to email the report to the faculty.

3.3 Graphical User Interface Design

The GUI has two panels and is constructed using Tkinter. While the right panel reveals student information such as name, registration number, major, total attendance, and image, the left panel shows the live camera stream. A mode switcher provides unambiguous user input while managing the various system states (loading, success, and error).

3.4 Supabase Integration

Database and storage management is done by Supabase. Complete access to tables and storage is granted via a service role key (SUPABASE_KEY). The registration_number is used to obtain photographs from the "images" bucket. All data is safely delivered via HTTPS, and row-level security (RLS) prevents unwanted access.

4. Result and Analysis

4.1 Performance Metrics:

Metric	Result
Face Matching Speed	15 FPS (640x480 resolution)
Encoding Accuracy	92% (tested on 50 students)
Database Latency	<500 ms for CRUD operations
Email Delivery	100% success rate (Gmail SMTP)

4.2 Test of Real-Time Recognition

Using the live camera feed, the algorithm was able to detect and identify several student faces at once. In order to improve performance, real-time frame resizing was used; during testing, the average speed was more than 20 frames per second (FPS). Accuracy was increased and false positives were decreased by only activating face recognition when a face stayed still in the frame. Even under various lighting scenarios, recognition was responsive and seamless.

4.3 Recording Attendance

Every face that was successfully identified led to an instant attendance record in the Supabase database. The system made sure that at least 60 seconds had elapsed since the student's last attendance was logged in order to avoid duplicate entries. This prevented repetition and guaranteed correct logging. All attendance data was validated using a verification script, which ensured accurate and consistent entries throughout several sessions.

4.4 Evaluation of Email Reporting

At the conclusion of each session, attendance reports were generated and sent by the dependable email reporting system. Excel reports with pertinent student data and timestamps were formatted correctly using the pandas package. Using app credentials, emails were sent safely via Gmail's SMTP server, and there were no delivery problems. With the least amount of manual labor, this automatic reporting function made sure that faculty members received timely attendance updates.

5. Conclusion and Discussion

The Automated Attendance System effectively substitutes a quick, precise, and real-time face recognition solution for human attendance methods. It effectively detects faces using OpenCV and face_recognition, and Supabase guarantees dependable cloud storage and simple data access.

By checking each student's face before marking them present, the technique minimizes human mistake and stops proxy attendance. Teachers find it easy to use because of email reports and a straightforward GUI.

However, when pupils wear masks or hats or in low light, recognition accuracy may decrease. Future iterations of this can employ deep learning models, such as ResNet, to increase accuracy and resilience in a range of settings.

All things considered, the project demonstrates how cloud integration and facial recognition may improve the security, scalability, and efficiency of attendance tracking.

Conclusion

By preventing proxy attendance and minimizing manual errors, the Automated Attendance System with Face Recognition efficiently automates the attendance process. The solution is scalable and effective since it uses Supabase for cloud storage and OpenCV for real-time face identification. The method shows great promise for expediting attendance in educational settings, despite ongoing issues including illumination and facial occlusions. Accuracy and robustness will be further improved in the future by using deep learning models, which will provide a more dependable answer under a variety of real-world circumstances.

6. Future Scope

1. Multiple Face Recognition:

- Improve the system so that it can identify and record several students' attendance at once in a single frame.
- For improved multi-facial recognition in a variety of lighting and angle scenarios, use sophisticated face detection models such as MTCNN or RetinaFace.

2. Mode Offline

- Create an offline mode that uses SQLite3 to locally cache attendance data during internet disruptions.
- After the internet connection is restored, make sure no data is lost by syncing the cached data to the Supabase database.

3. Advanced Information Analysis

- To visualize attendance trends, including absenteeism patterns over time, combine Matplotlib with pandas.
- To assist teachers in identifying and resolving problems, provide thorough reports on attendance rates by student, subject, or class.

4. Integration of Mobile

- Create a companion smartphone app that will allow students to see their attendance history and get real-time absence alerts.
- Provide tools such as push notifications to inform students when they have been declared absent or to remind them to come to class.

5. Increased Security

- To stop spoofing attempts, use liveness detection to make sure that only real students are listed as present.
- Use methods such as 3D depth analysis or eye movement tracking to confirm the legitimacy of faces the system has identified.

References

- OpenCV: https://opencv.org

- Supabase (2023). SDK for Supabase Python. Reference: https://supabase.com/docs/python

- Pandas Development Team. (2023). pandas: Powerful Data Analysis Tools. https://pandas.pydata.org

- Python Software Foundation. (2023). smtplib Documentation. https://docs.python.org/3/library/smtplib.html