

Face-Recognition Based Smart Attendance System for Classroom Monitoring

Bishwa Ranjan Behara¹, Tarun², and Tejaswini³

¹*B. Tech (ME), Manager at Learning Links Foundation (Technology & Innovation)*

RMOC (Regional Mentor of Change) at Atal Innovation Mission

²*Class 9th BBMP High School, Byraweshawaranagar*

³*Class 8th BBMP High School, Byraweshawaranagar*

Abstract— This paper presents a real-time face recognition-based attendance system designed for classroom environments. The proposed system uses a standard USB webcam or IP-based CCTV camera to capture live video of students, and employs OpenCV for face detection together with a facial-embedding based face recognition library to identify individual students in the frame. A small dataset of registered students is collected in advance, from which facial encodings are extracted and stored locally. During a session, the system detects faces in each frame, compares them with the known encodings, and automatically marks attendance for recognized students in a CSV file along with timestamp information, ensuring that each student is logged only once per session. The implementation is prototyped on a low-cost platform using Python and Raspberry Pi / personal computer hardware, making it suitable for resource-constrained educational settings. Experimental tests in a single-classroom scenario demonstrate that the system can operate in real time and achieve reliable recognition under reasonable lighting and camera placement. Limitations under poor lighting, occlusions, and increased class size are discussed, along with future integration using RTSP streams from CCTV cameras and central database connectivity.

Index Terms— Attendance system, Face recognition, Raspberry Pi, OpenCV, Computer vision, Classroom automation.

I. INTRODUCTION

In many schools and colleges, classroom attendance is still recorded manually using paper registers or simple spreadsheets. This process is time-consuming, error-prone, and difficult to audit at scale. Teachers lose valuable instructional time, and administrators often have limited visibility into attendance patterns and student engagement. With increasing class sizes and

multiple sections, there is a growing need for an automated, reliable, and scalable attendance solution. Biometric technologies have been widely explored for attendance management, including fingerprint, RFID cards, QR codes, and face recognition. Fingerprint and RFID systems require students to physically interact with a device or carry an identity card, which can cause queues, card loss, or hygiene concerns. QR-code based solutions often require smartphones or printed codes, which may not be suitable in all classroom contexts. Face recognition, in contrast, is a non-contact and passive method: students simply sit in their usual places while a camera captures their faces for identification.

This paper focuses on a face-recognition based smart attendance system tailored for classroom monitoring. A webcam or CCTV camera is used to capture the classroom view, and a vision algorithm detects all faces in the frame. A recognition module then identifies each student by comparing the detected face with a database of known faces. When a match is found, the system automatically marks the student present and records the timestamp. The goal is to minimize manual intervention while keeping the system affordable and easy to deploy in typical Indian school settings.

The main contributions of this work are:

1. Design and implementation of a real-time classroom attendance system using a combination of traditional face detection and modern face encodings.
2. A practical pipeline for registering students, storing face encodings, and logging attendance in CSV format that can be later imported into school management systems.

3. Validation of the system in a live classroom environment using low-cost hardware such as Raspberry Pi 4 Model B and a standard USB camera, along with a discussion of limitations and future improvements.

The rest of the paper is organized as follows. Section II presents related work and background concepts in face detection and recognition. Section III describes the proposed system architecture and methodology. Section IV explains the implementation details. Section V presents experimental results and discussion. Section VI concludes the paper and outlines future work.

This work was carried out as part of the CGI–Learning Links Foundation program at BBMP High School, Byraweshwaranagar, which focuses on fostering innovation and hands-on STEM learning among school students.

II. RELATED WORK AND BACKGROUND

Face-recognition based attendance systems have been widely studied in recent years as computing hardware and computer vision algorithms have become more accessible. Many works adopt a two-step approach: face detection followed by face recognition. Classical face detection methods include Haar Cascade classifiers and Histogram of Oriented Gradients (HOG) detectors, while modern systems increasingly use deep learning based detectors. For recognition, popular approaches include Eigenfaces, Fisherfaces, Local Binary Pattern Histograms (LBPH), and deep learning based face embeddings.

Several academic and hobby projects have demonstrated small-scale attendance systems using OpenCV for detection and recognition. Many of these systems are tailored for laboratory environments or small groups of users. However, practical classroom deployment requires robustness to variations in lighting, seating positions, and student movement. It also requires the ability to log attendance in a format that is easy for teachers and administrators to use, such as CSV files or direct integration with existing school information systems.

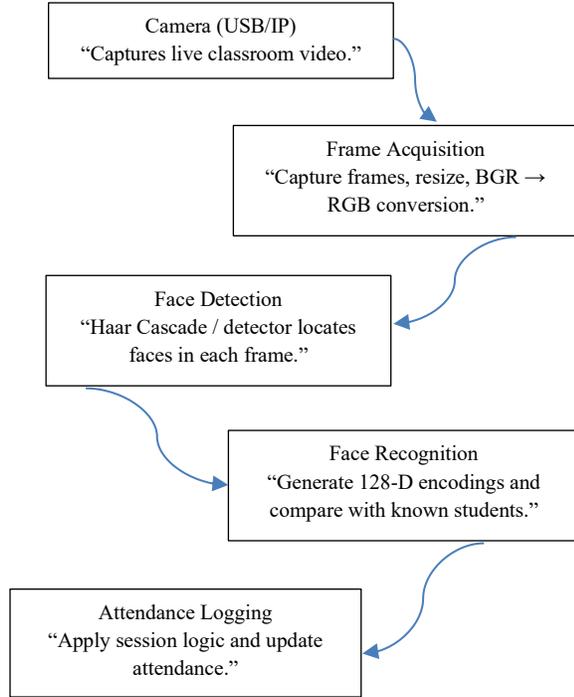
In this work, we adopt a pragmatic approach that balances accuracy and computational cost. For face detection, we use a Haar Cascade detector for frontal faces, which is well-established and widely available in OpenCV. For recognition, we use a facial-embedding based library that generates 128-dimensional encodings for each face. Matching is performed by computing distances between encodings of detected faces and stored encodings of registered students. This combination allows the system to run in real time on both a laptop and a Raspberry Pi 4 Model B, which is important for low-cost deployment.

III. SYSTEM ARCHITECTURE AND METHODOLOGY

The proposed system follows a modular architecture. At a high level, it consists of four main components: image acquisition, face detection, face recognition, and attendance logging.

1. **Image Acquisition:** A camera is placed at the front of the classroom to capture video of the students. For the prototype, a USB webcam connected to a laptop or Raspberry Pi is used. In future deployments, this can be replaced with an IP-based CCTV camera that provides an RTSP video stream.
2. **Face Detection:** Each frame from the camera is processed by a face detection module. The frame is converted to grayscale, and a pre-trained Haar Cascade classifier is applied to locate regions containing faces. For each detected face, the system obtains its position in the image.
3. **Face Recognition:** Detected face regions are passed to a recognition module. Before the main system is used, a registration phase is carried out in which images of known students are collected. From these images, face encodings are generated and stored. During operation, for each detected face in the live video, an encoding is computed and compared with the stored encodings using a distance metric. If the distance is below a predefined threshold, the corresponding student is considered recognized.
4. **Attendance Logging and Interface:** Recognized students are marked present in a CSV file that includes their name, date, and time of first recognition in that session. Each student is logged only once per session to avoid duplicate entries.

The live video feed is overlaid with bounding boxes around faces, name labels, and statistics such as the number of detected and recognized students and the current time. This serves as a simple user interface for the teacher.



IV. IMPLEMENTATION

The system is implemented using Python. The main libraries used are OpenCV for image processing and display, a face-recognition library for computing face encodings and matching, NumPy for numerical operations, and Pandas for CSV-based attendance logging. The code is organized into three main scripts: a camera test script, an encoding script, and the main attendance script.

A. Hardware and Software Setup

The prototype uses a standard USB webcam connected either to a Windows or Linux laptop, or to a Raspberry Pi 4 Model B with at least 2 GB RAM. The Raspberry Pi runs a Linux-based operating system and has Python and the required libraries installed through package managers and pip. The camera is verified using a simple script that captures frames and displays them in a window so that the user can confirm correct installation.

B. Student Registration and Encoding

A folder named “known_faces” is created in the project directory. For each student, a subfolder with the student’s name is created, and 1–3 clear images of that student’s face are stored in that subfolder. These images should be captured in good lighting, with the face mostly frontal and fully visible.

An encoding script is used to process all student images and generate face encodings. The script traverses each subfolder, loads each image, detects the face, and computes a 128-dimensional face encoding. These encodings are stored in a list along with the corresponding student names. Finally, the encodings and names are serialized to a file (for example, a pickle file) for later use. This registration procedure only needs to be repeated when new students are added or existing images are updated.

C. Real-Time Detection and Recognition

The main attendance script opens the camera using the default video capture interface. For each frame, the image is optionally resized to reduce computational load, which is especially important when running on a Raspberry Pi. The frame is converted from BGR color space (used by OpenCV) to RGB (required by the face-recognition library).

Face locations are detected in the current frame, and for each location a face encoding is computed. Each encoding is compared with all stored encodings using a distance function. If the closest match is below a certain tolerance threshold, the system assigns the corresponding student name to that face; otherwise, it labels the face as “Unknown”. The threshold can be tuned to trade-off between false positives and false negatives.

The system draws a bounding box around each detected face and overlays the recognized name below the box. It also counts the total number of detected faces and the number of recognized students. These numbers, along with the current time, are displayed as text at the top of the video window. This provides real-time feedback to the user.

D. Attendance Logging and Session Logic

To manage attendance, the system maintains a set of names for which attendance has already been marked in the current session. When a recognized face is

detected and its name is not “Unknown”, the system checks this set. If the name is not present, it appends a new row to a CSV file with the student’s name, the current date, and the current time, and adds the name to the set. If the name is already present, no new entry is added. This ensures that each student is marked present only once per session, even if the face is repeatedly detected in subsequent frames.

The CSV file serves as the attendance log and can be opened directly in spreadsheet software or imported into a school management system. Each run of the system corresponds to a session; if the program is restarted, the in-memory set is cleared, and a new session can begin.

V. RESULTS AND DISCUSSION

The system was evaluated in a classroom-like environment with a small group of students. Images for 3–5 students were collected and used to build the initial database of known faces. The camera was placed at the front of the room, slightly above eye level, pointing towards the students. Tests were conducted using both a laptop and a Raspberry Pi 4 Model B to assess performance across different hardware.

In well-lit conditions, with students facing towards the camera, the system was able to detect and recognize faces in real time. The frame rate on a laptop was higher than on the Raspberry Pi, but even on the Raspberry Pi the processing speed was sufficient for classroom monitoring when the image was resized to a lower resolution. The system successfully logged each student to the attendance CSV only once per session, and the on-screen overlay helped to verify which students were recognized.

However, several limitations were observed. When lighting was poor, or when strong backlight was present (for example, windows behind students), detection and recognition accuracy decreased. Occlusions such as masks, hands covering the face, or students turning to the side also caused occasional missed detections or misrecognitions. In addition, as the number of students increases and faces become smaller in the image, the difficulty of reliable recognition grows, especially on lower-resolution cameras.

To mitigate these issues, multiple images per student were collected under slightly different conditions (with and without glasses, slight head rotations), which improved robustness. The tolerance parameter used for matching was tuned to reduce false positives. Camera placement and angle also played an important role: placing the camera centrally and ensuring that faces occupied a reasonable number of pixels significantly improved performance.

Overall, the results suggest that the proposed system is feasible for small to medium-sized classrooms and can be used as a supportive tool for teachers. It performs best under controlled lighting and when students are relatively static for at least a few seconds at the start of the session.

VI. FUTURE WORK AND CCTV INTEGRATION

An important future extension of this work is integration with IP-based CCTV cameras. Many schools already have CCTV infrastructure for security and monitoring. Instead of a directly connected USB webcam, the system can be configured to receive a video stream via the RTSP protocol from an IP camera. In the implementation, this requires only a small modification to the video capture code, replacing the camera index with an RTSP URL. The rest of the pipeline—detection, recognition, and logging—remains unchanged.

For larger deployments, multiple cameras may be needed to cover different parts of a classroom or multiple classrooms. Each camera can run an instance of the attendance system, either on separate devices or on a more powerful central server. The attendance logs from each instance can then be merged into a centralized database, enabling school-wide attendance analytics and reporting.

Other future improvements include replacing the traditional Haar Cascade detector with a more robust deep learning based detector to handle non-frontal faces and varying lighting conditions; incorporating anti-spoofing methods to prevent misuse of photographs or videos; and developing a

web-based dashboard for administrators to view attendance in real time.

VII. CONCLUSION

This paper has presented a face-recognition based smart attendance system for classroom monitoring. By combining a camera, a face detection module, a face recognition module, and a CSV-based logging mechanism, the system can automatically identify students and mark their attendance in real time. The implementation using Python, OpenCV, and Raspberry Pi demonstrates that such a system can be built with low-cost hardware and open-source software, making it accessible for schools with limited resources.

Experimental tests indicate that the system works effectively in small to medium-sized classrooms under reasonable lighting and camera placement. While limitations remain, especially in challenging visual conditions and crowded scenes, the proposed system offers a practical step towards automated attendance and can be extended to support CCTV integration and centralized data management. With further refinement and field testing, this approach has the potential to reduce teacher workload, improve data accuracy, and provide richer insights into student attendance patterns.

ACKNOWLEDGMENT

The authors would like to thank CGI and Learning Links Foundation for their continuous mentorship and support through the CGI–Learning Links Foundation program at BBMP High School, Byraweshwaranagar. The authors also thank the school management, teachers, and students for providing the environment and cooperation needed to develop and test the proposed attendance system.

REFERENCES

- [1] G. Bradski, “The OpenCV Library,” *Dr. Dobb’s Journal of Software Tools*, 2000.
- [2] A. Geitgey, “face_recognition: Simple Facial Recognition Library for Python,” GitHub repository, available: https://github.com/ageitgey/face_recognition, accessed Feb. 2026.
- [3] P. Viola and M. Jones, “Rapid Object Detection Using a Boosted Cascade of Simple Features,” in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2001, pp. 511–518.
- [4] A. K. Jain, A. Ross, and S. Prabhakar, “An Introduction to Biometric Recognition,” *IEEE Trans. Circuits and Systems for Video Technology*, vol. 14, no. 1, pp. 4–20, Jan. 2004.
- [5] M. Turk and A. Pentland, “Eigenfaces for Recognition,” *Journal of Cognitive Neuroscience*, vol. 3, no. 1, pp. 71–86, 1991.
- [6] S. L. Happy and A. Routray, “Automatic Facial Expression Recognition Using Features of Salient Facial Patches,” *IEEE Trans. Affective Computing*, vol. 6, no. 1, pp. 1–12, Jan.–Mar. 2015.
- [7] “Capture RTSP Stream from IP Camera using OpenCV,” Online article, available: <https://lindevs.com/capture-rtsp-stream-from-ip-camera-using-opencv>, accessed Feb. 2026.