Smart Attendance System

Dhamal Vaishnavi [1], Kanthe Pranita [2], Narute Ankita[3], Nimbhorkar Dyaneshwar[4]
[1,2,3,4] *Svpm Coe Malegaon (Bk), Baramati.*

***Abstract*—Tracking student attendance in educational institutions is often tedious, inaccurate, and susceptible to manipulation when done manually. To address this, we propose a smart attendance system that leverages advanced deep learning methods for automated, real-time monitoring. The system combines FaceNet for robust facial feature extraction and YOLOv8 for detecting mobile phones during class sessions, thereby maintaining student discipline and minimizing distractions. If a mobile phone is detected, the system instantly sends a Pushbullet notification to designate authorities. Attendance is marked based on the cosine similarity between live-captured face embeddings and pre-stored feature vectors, removing the need for manual checking. An SQLite database manages student profiles, attendance logs, and class timetables efficiently. Additionally, a Flask-powered web interface provides user-friendly modules for student registration, schedule management, and visual attendance tracking. The solution also includes automated entry and exit time tracking, mapped to real-time class durations. This contactless and intelligent system significantly boosts administrative productivity and classroom discipline.**

***Index Terms*—Face Recognition, FaceNet, YOLOv8, Cosine Similarity, Attendance System, Deep Learning, Push Notifications, SQLite, Flask Web App**

I. INTRODUCTION

In recent years, artificial intelligence and computer vision have significantly improved traditional attendance systems, making them smarter, faster, and more reliable. Manual attendance tracking is often time-consuming, error-prone, and vulnerable to impersonation. To address these challenges, this paper presents an intelligent smart attendance system that uses deep learning for face recognition and real-time mobile phone detection to ensure accurate tracking and maintain discipline.

Developed using Python and Flask, the system integrates FaceNet (InceptionResNetV1) for high-accuracy facial recognition and YOLOv8 to detect mobile phone usage within classrooms. When a mobile device is detected, Pushbullet is used to instantly notify authorities such as the principal. The system also features a class configuration module where departments can define class schedules and durations, automatically calculating each student's attendance based on timestamped entry and exit data.

Attendance is recorded automatically using cosine similarity to compare live face embeddings with pre-registered ones. To avoid duplicate alerts, a cooldown mechanism is included for mobile detection. All data is securely managed in an SQLite database.

The system is scalable, supports multiple class configurations, and provides a real-time video stream interface. It offers a contactless, efficient, and reliable alternative to traditional systems, while also enhancing classroom discipline through integrated device monitoring.

II. RELATED WORKDONE

1.Face Recognition with FaceNet

Schroff et al. [1] introduced FaceNet, a deep learning model that maps facial features into a Euclidean embedding space, achieving high face verification accuracy. Its robustness to variations in lighting, pose, and occlusion has made it the backbone of many facial recognition systems. Several attendance solutions have adopted FaceNet for its ability to identify students quickly and contactless, increasing both efficiency and user satisfaction.

2. Mobile Phone Detection Using YOLOv8

The YOLO (You Only Look Once) series, particularly YOLOv8, has set benchmarks in object detection for speed and accuracy. Redmon et al. [2] pioneered YOLO for real-time detection tasks, and the newer versions such as YOLOv8 have improved precision significantly. Integrating YOLO into classroom settings enables detection of unauthorized mobile phone usage, a key feature missing in most existing attendance systems.

III. METHODOLOGY

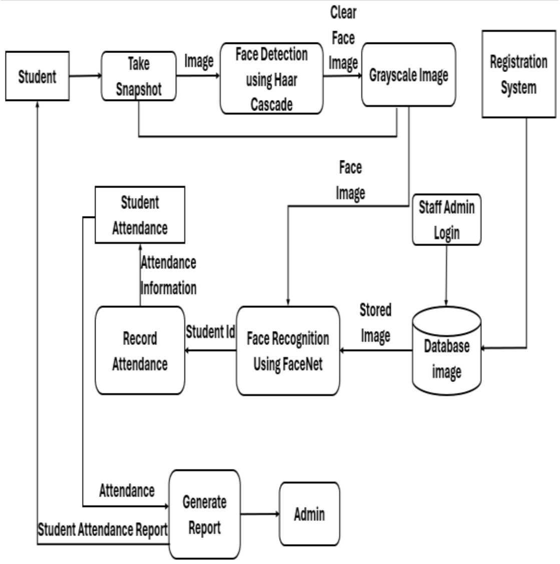


Fig. System Architecture.

1.Student Entry and Image Capture:
The process initiates when a student interacts with the system interface. Upon triggering, the "Take Snapshot" feature captures the student's face using a webcam in real-time.
2.FaceDetectionUsingHaarClassifier:
The captured photo is then analysed using the Haar Cascade algorithm, which scans the image to detect and locate facial features.
3.Preprocessing the Detected Face:
Once the face is detected, it undergoes an image enhancement phase. This step improves clarity and quality through alignment, resizing, or denoising to ensure accurate recognition results.
4.Conversion to Grayscale Format:
To minimize processing complexity, the enhanced image is transformed into a grayscale version. This helps streamline further analysis and boosts model performance.
5.Embedding Generation via FaceNet:
The grayscale image is passed to the FaceNet model, which extracts unique facial embeddings. These embeddings act as numerical representations of the student's identity.
6.Image Database Comparison:
The newly generated embedding is compared with existing embeddings in the student database to find a match. Each stored profile includes a face vector linked to the student's identity.
7.Attendance Entry and Logging:
On successful identification, the system records the attendance along with a timestamp and class details. This eliminates the need for manual entry.
8.Admin Panel for System Control:
The application includes a login-secured admin panel for faculty and staff. Through this, authorized users can manage student records, review logs, and update the image database as needed.
9.Automated Report Generation:
Attendance data collected throughout sessions is used to generate detailed attendance reports. These reports can be exported for further administrative use.
10.Connection to Student Registration System:
The system is integrated with an external registration database, allowing for automatic syncing of student details during the enrolment phase.

IV. PROPOSED SOLUTION

This paper proposes a robust, real-time smart attendance system leveraging advanced face recognition and object detection technologies integrated into a web-based client-server architecture using Python and Flask. The system is designed to automate and enhance the traditional manual attendance process with improved accuracy, efficiency, and scalability.

A. Face Recognition-Based Attendance
The solution utilizes the FaceNet model (InceptionResNetV1 pretrained on VGGFace2) to extract high-dimensional facial embeddings for accurate identification. Haar cascade classifiers are first used for real-time face detection from live video streams. Each detected face is then resized and passed through the FaceNet model to generate a 512-dimensional facial embedding. These embeddings are stored in a SQLite database during registration and later matched during recognition using cosine similarity, allowing precise verification even with minor variations in facial appearance.

Haar cascade
The mathematical expression for a single Haar feature can be generalized as:

$$F = i \in Rwhite \sum I(i) - j \in Rblack \sum I(j)$$

FaceNet

FaceNet is a deep learning algorithm developed by Google for face recognition. FaceNet uses deep convolutional neural networks (CNNs) to learn a mapping of faces directly into a 128-dimensional embedding space where similar faces are grouped closer together, and dissimilar faces are further apart.

1.Embedding Function: For an input image x, the FaceNet model maps it to a 128-dimensional embedding f(x):

$$f(x) \in R^{128}$$

2.Triplet Loss: The Triplet Loss is the core function used to train FaceNet. It ensures that the distance between an anchor image A and a positive image P (same person) is smaller than the distance between the anchor and a negative image N (different person) by a margin α\alphaα.

$$\text{Loss} = \max(0, \| f(A) - f(P) \|^2 - \| f(A) - f(N) \|^2 + \alpha)$$

3.Euclidean Distance: The Euclidean distance between two embeddings f(x1) and f(x2) is used to measure similarity between two faces:

$$d(x1, x2) = ||f(x1) - f(x2)||^2$$

B. YOLOv8-Based Mobile Detection

To ensure discipline and prevent misuse of mobile devices during class hours, the proposed system integrates YOLOv8 (You Only Look Once) for object detection. YOLO is trained on the COCO dataset and identifies mobile phones (class ID 67). When a mobile phone is detected with confidence greater than 70%, an alert is sent to the admin/principal in real-time using the Pushbullet notification service, helping monitor in-class behaviour.

C. Automated Entry and Exit Tracking

The system dynamically handles entry and exit tracking based on face recognition events. When a face is detected and matched, the system checks the SQLite database for an ongoing session. If none is found, a new attendance entry is created with the current timestamp. If an ongoing session exists, the system calculates the duration of presence and automatically determines the number of classes attended based on a pre-configured timetable. This eliminates the need for manual class duration inputs and enables real-time calculation of attendance status (e.g., "Present" or "Leave Early").

D. Web-Based Interface with Flask

The backend is developed using Flask, providing a simple and intuitive web interface for registration, class configuration, and real-time video stream monitoring. Users can register themselves by uploading a face image along with their name, roll number, and department. Admins can configure class schedules dynamically. The frontend includes routes for registering new users (/register page), configuring classes (/configure classes), and viewing real-time recognition via /video feed.

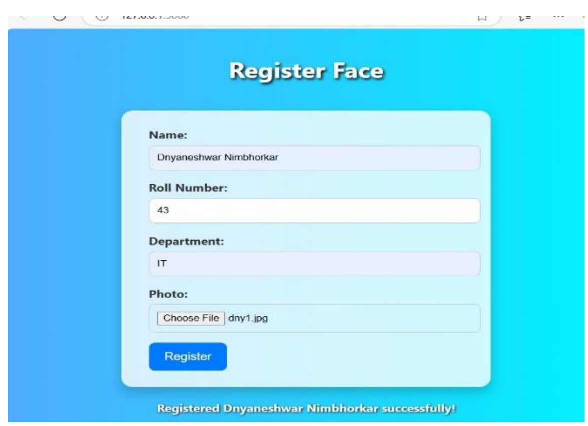
E. SQLite-Based Storage System

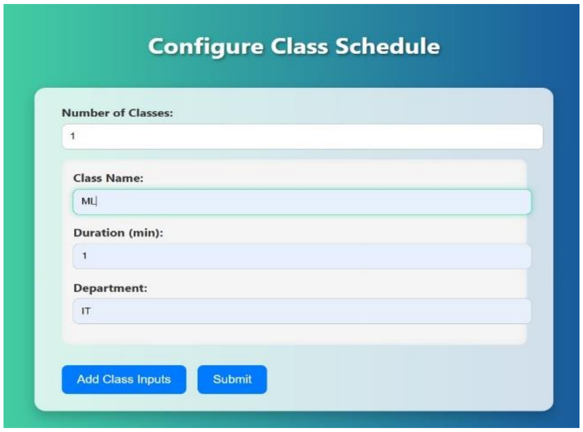
For data persistence and easy portability, the system employs SQLite as the backend database. Three main tables are maintained: users (storing name, roll number, department, and facial embeddings), attendance (logging entry, exit, and status), and class schedule (storing department-wise class names and durations). This enables precise attendance analysis and reporting while being lightweight and easy to deploy.

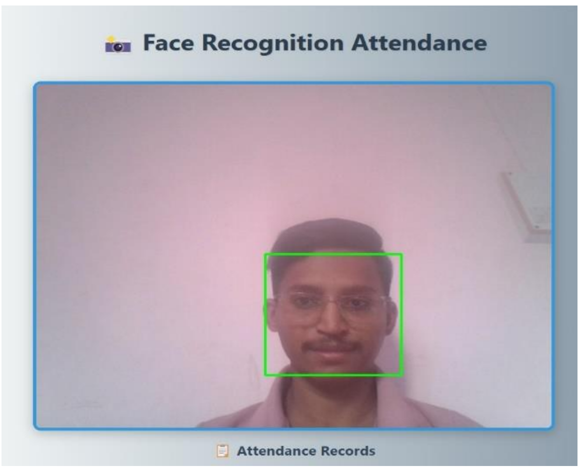
F. Multithreaded Performance Optimization

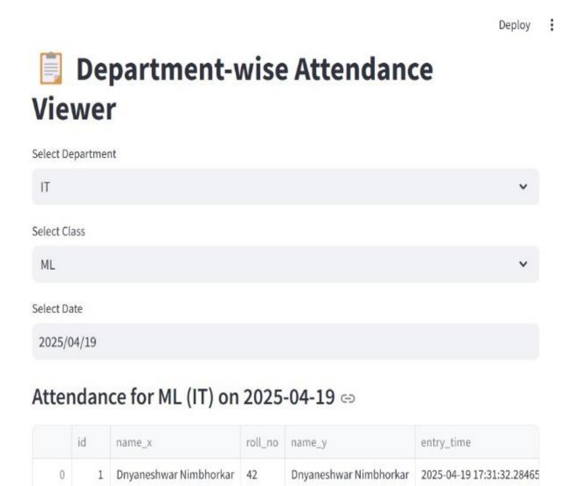
To ensure real-time responsiveness and reduce lag, the system employs multithreading. Face recognition and attendance marking are executed in background threads, allowing continuous video streaming and UI responsiveness. Additionally, a cooldown mechanism is implemented to avoid redundant alerts for mobile phone detection.

V. RESULTS









VIII. CONCLUSION

This Flask-based smart attendance system leverages FaceNet for face recognition and YOLO for mobile phone detection to automate and monitor classroom attendance efficiently. It allows students to register with their face data, and during live video streaming, it detects and recognizes faces in real time, marking entry and exit while calculating class attendance based on predefined schedules stored in a SQLite database. Additionally, it uses Pushbullet to send alerts when mobile phones are detected in the classroom, promoting discipline. The system provides an intuitive web interface for registration, class configuration, and attendance tracking, making it a comprehensive and intelligent solution for educational institutions.

ACKNOWLEDGMENT

The authors would like to express their sincere gratitude to the faculty and project guides of SVPM College Of Engineering Malegaon (BK) Baramati. for their continuous support, valuable insights, and encouragement throughout the development of this project. Special thanks to the Department of Information Technology for providing the necessary resources and infrastructure. The authors also acknowledge the use of open-source libraries such as Flask, FaceNet, YOLO and SQLite, which played a crucial role in building this system.

REFERENCES

[1] P. Viola and M. Jones, “Rapid Object Detection using a Boosted Cascade of Simple Features,” Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2001, pp. 511–518.

[2] F. Schroff, D. Kalenichenko, and J. Philbin, “FaceNet: A Unified Embedding for Face Recognition and Clustering,” 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 815–823.

[3] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," arXiv preprint arXiv:1804.02767, 2018.
[4] G. Jocher et al., "YOLOv5: Implementation of YOLO Object Detection in PyTorch," Ultralytics, 2020. [Online]. Available: https://github.com/ultralytics/yolov5
[5] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 4510–4520.
[6] D. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," International Conference on Learning Representations (ICLR), 2015.
[7] A. Krizhevsky, I. Sutskever, and G.E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," Communications of the ACM, vol. 60, no. 6, 2017, pp. 84–90.
[8] R. Kaur and P. Singh, "Face Recognition Based Attendance System using Machine Learning Algorithms," International Journal of Computer Applications, vol. 182, no. 33, 2018, pp. 18–22.
[9] M. M. Hasan, M. S. Rahman, and S. M. Rahman, "Smart Attendance Management System Using Face Recognition," International Journal of Computer Science and Network Security (IJCSNS), vol. 21, no. 6, pp. 112–117, June 2021.
[10] J. He, W. Wu, Z. Yang, and X. Liang, "YOLOv5 Face Recognition for Classroom Attendance Monitoring," 2022 IEEE 6th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), 2022, pp. 1345–1350.
[11] A. A. Parvez, M. A. Hossain, and S. M. Ali, "Development of a Real-Time Face Recognition System Using YOLOv5 and Face Net," International Journal of Machine Learning and Computing, vol. 12, no. 5, pp. 413–420, Oct. 2022.