

Vision-Based Intelligent Classroom Management and Monitoring System

Karthik S, Tejas Gowda N R, Bhuvan Shankar G, Bharath Kumar H P
Department of Information Science and Engineering
Global Academy of Technology
Bangalore, India

Deepak G
Professor, Department of Information Science and Engineering
Global Academy of Technology
Bangalore, India

Abstract—This work presents a smart classroom system designed to automate attendance and improve classroom monitoring using facial recognition. The model uses a webcam to capture live video, and each frame is then processed to detect and identify student faces. The extracted facial features are compared with stored data to recognize students, and record their attendance along with timestamps. A dashboard interface is used to display attendance records and provide basic monitoring features for instructors. The model was tested under different conditions such as lighting changes and partial face visibility, where it showed stable performance. By reducing manual effort and improving reliability, the model offers a practical solution for modern classroom environments.

Keywords—*Facial Recognition, Real-Time Attendance, Machine Learning, Automation, Face Encoding.*

I. INTRODUCTION

In many institutions, attendance is still recorded manually using roll calls or paper registers. While simple, these methods take time and are prone to errors, including proxy attendance [2], [4], [8], [15].

With the growth of artificial intelligence and computer vision, more efficient alternatives are now possible [6], [11]. Face recognition is widely used today since it does not require any physical interaction and can work under different conditions such as varying lighting or slight changes in face orientation. In recent years, deep learning models have improved recognition accuracy to a great extent, making real-time applications feasible [1], [7], [20]. However, most existing systems focus only on attendance tracking and do not provide additional insights about classroom activity. In this work, an attempt is made to combine attendance automation with basic classroom monitoring through a single system supported by a centralized dashboard [6], [11], [12].

II. LITERATURE REVIEW

Previous studies have explored different approaches for automating attendance. Joshi et al.[1] developed a system using OpenCV and a face recognition library, where facial features were encoded and matched with live video input. Although the system performed well, its accuracy decreased under poor lighting and motion conditions.

Hemavathi[20] proposed a system that combines attendance with security features using YOLOv8 and convolutional neural networks. It was capable of detecting unauthorized individuals and sending

alerts, making it suitable for controlled environments.

Another study by Sirivarshitha et al. [13] used traditional methods such as Haar Cascade and PCA. While the system worked for basic applications, its performance was lower compared to deep learning

approaches, especially when handling multiple faces.

From these studies, it is clear that deep learning methods provide better accuracy and robustness. However, most existing systems are limited to attendance marking and do not focus on classroom monitoring, which is addressed in this work.

III. METHODOLOGY

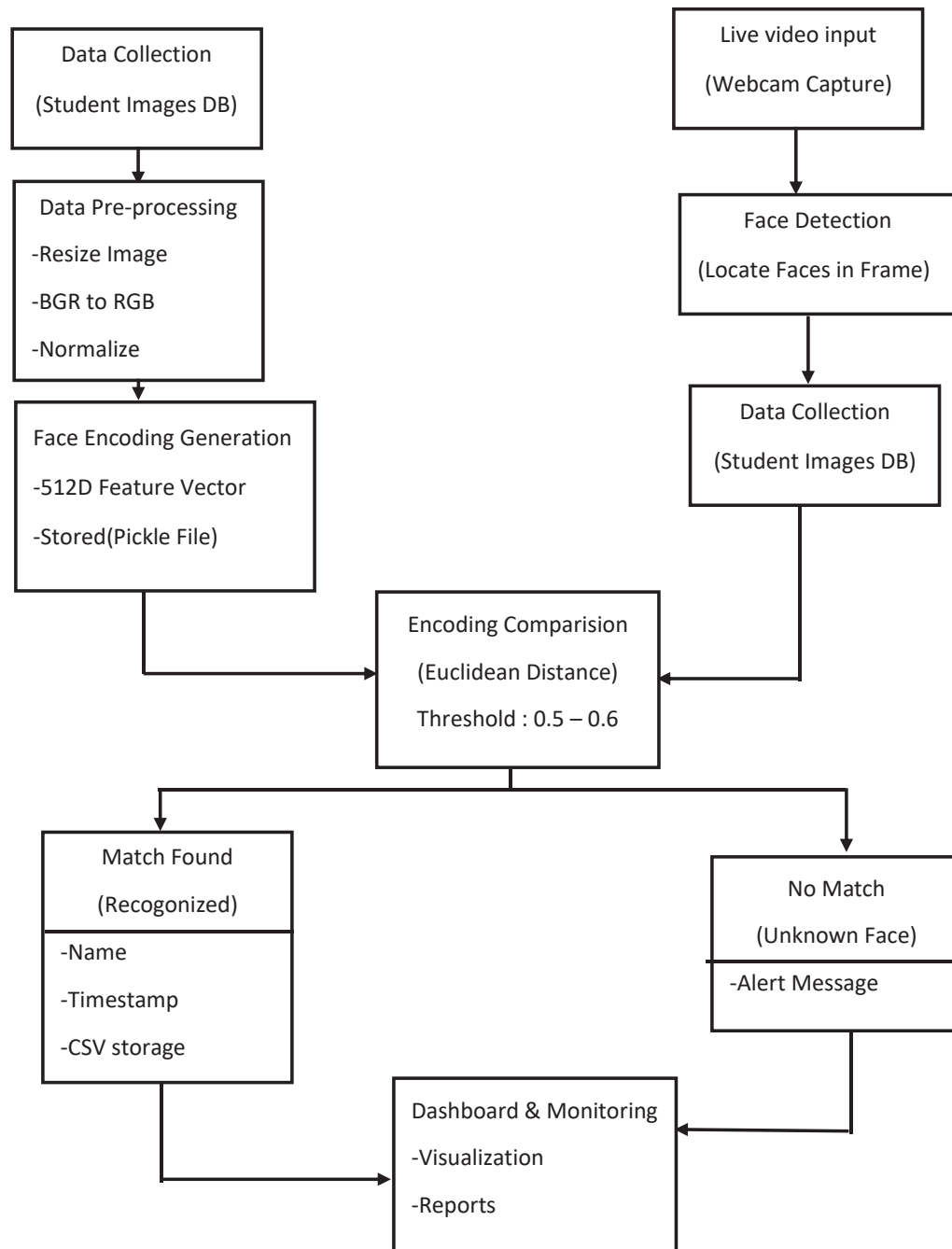


Fig. 1. Block Diagram of Proposed Model.

Fig. 1 shows the overall workflow of the system. It begins by collecting multiple images of each student under different lighting conditions and face orientations. These images are stored in separate folders for each student to maintain organization.

Before processing, the images are resized so that the computation time can be reduced. and converted from BGR to RGB format to match the requirements of the recognition model. After that, facial features are extracted and converted into numerical form (encodings), which are stored for later use.

During execution, live video is captured through a webcam. Each frame is processed to detect faces, and encodings are generated for the detected faces. These encodings are then compared with stored data using Euclidean distance.

If the similarity score falls within a predefined threshold, the student is identified, and attendance is recorded automatically along with the current timestamp. If no match is found, the system treats the face as unknown and can trigger an alert.

A. Data Collection

The methodology begins with collecting multiple facial images of each student under varying orientations and lighting conditions to improve model robustness. Student's images are collected during the data collection process which contains set of Individual Folders Which Contains Name of the Student as Folder Name and each Folder Contains Their Respective images for this system to recognise face [1], [14].

$$D = \{I_{i,j} | i=1,2,\dots,N; j=1,2,\dots,M_i\} \quad (1)$$

where:

- $I_{i,j} = j^{th}$ image of i^{th} student
- $N =$ total number of students
- $M_i =$ number of images per student

B. Data Pre processing

Before initiating face detection, images undergo preprocessing to extract face encodings. Newly

detected data is appended to the known student dataset, and a new CSV file is generated for the current date to record attendance [1], [13].

- 1) Image Pre-processing: Initially in pre-processing phase images are loaded into memory using the OpenCV's cv2.imread() $I(x,y) = \text{LoadImage}(\text{path})$ (2)

function. This function reads the images from the specified directory. After loading each image, it's resized for efficient processing. The resizing factor used here 0.25 of the original size given by the following formula:

$$\text{Resized Image} = \text{Original Image size} * 0.25 \quad (3)$$

Once resized, the image is converted from the BGR color space (OpenCV reads file into BGR Format) to the RGB color space (understood by face recognition library) by the following formula:

$$IRGB(x,y) = [R(x,y), G(x,y), B(x,y)] \quad (4)$$

- 2) Face Encoding Generations and Management: Facial features are extracted and represented as a 512-dimensional encoding vector, which uniquely identifies each face. These encodings are stored using pickle serialization for efficient storage and retrieval.

$$E_i = [f_1, f_2, f_3, \dots, f_{512}] \quad (5)$$

where:

- $f(\cdot) =$ deep learning feature extractor
- 3) New Image Identification: The system checks for newly added images by comparing filenames with previously processed records. Only new images are encoded and added to the existing database to avoid duplicate processing.
 - 4) Attendance File Management: A date-wise CSV file is generated to store attendance details. Each recognized student's information and timestamp are updated automatically.

$$A = \{(s_i, t_i)\} \quad (6)$$

where:

- s_i = student identity
- t_i = timestamp

C. Face Capturing

The system captures live video frames from the webcam after preprocessing. Each frame is resized and converted from BGR to RGB format. Faces are then detected in the frame, and facial encodings are generated for the detected faces [2], [20].

$$V = \{F_t | t = 1, 2, 3, \dots\} \quad (7)$$

where:

- F_t = frame at time t

D. Face Encoding Comparison

The generated face encodings from the live feed are compared with stored encodings using Euclidean distance [9], [13]. If the distance is below a predefined threshold (0.5–0.6), the face is recognized as a match and attendance is marked.

$$d(E_{live}, E_k) = \sqrt{\sum_{i=1}^{512} (E_{live_i} - E_{k_i})^2} \quad (8)$$

Where:

- E_{live} → Encoding of the detected face (live camera)
- E_k → Encoding of a stored face (database)
- i → Feature index (from 1 to 512)
- $E_{live,i}, E_{k,i}$ → Individual feature values
- 512 → Number of features in face encoding

E. Attendance Marking

Once a match between a detected face and the stored encodings is confirmed, the system proceeds to mark the student's attendance. The student's details are recorded in a CSV file along with the current time [1], [8], [15].

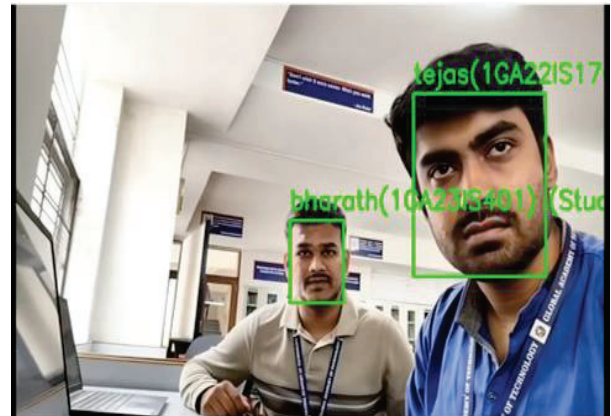


Fig. 2. Face Recognition

Fig. 2. Demonstrates how system's camera interface appears after all the processing and starts capturing frames for student's image to identify and mark attendance it showcases a bounding box as soon as it detects face and labels it with its data stored in the encodings.pkl file.

1	Date	Time	ID	Name	Subject	Faculty
2	07-11-2025	08:21:17	GAT007	darshan	Blockchain	Kumar
3	07-11-2025	08:21:17	GAT001	abhi	Blockchain	Kumar
4	07-11-2025	08:21:17	GAT002	aditya	Blockchain	Kumar
5	07-11-2025	08:21:17	GAT003	arya	Blockchain	Kumar
6	07-11-2025	08:21:17	GAT004	bharath(1ga23is401)	Blockchain	Kumar
7	07-11-2025	08:21:17	GAT005	bhuvan(1ga23is402)	Blockchain	Kumar
8	07-11-2025	08:21:17	GAT006	charan	Blockchain	Kumar
9	07-11-2025	08:21:17	GAT007	darshan	Blockchain	Kumar
10	07-11-2025	08:21:17	GAT008	deepak	Blockchain	Kumar
11	07-11-2025	08:21:17	GAT009	gopi	Blockchain	Kumar
12	07-11-2025	08:21:17	GAT010	karthik(1ga23is406)	Blockchain	Kumar
13	07-11-2025	08:21:17	GAT011	kiran	Blockchain	Kumar
14	07-11-2025	08:21:17	GAT012	kumar	Blockchain	Kumar
15	07-11-2025	08:21:17	GAT013	muni	Blockchain	Kumar
16	07-11-2025	08:21:17	GAT014	pavan	Blockchain	Kumar
17	07-11-2025	08:21:17	GAT015	prajwal	Blockchain	Kumar
18	07-11-2025	08:21:18	GAT016	rakesh	Blockchain	Kumar
19	07-11-2025	08:21:18	GAT033	thippesh	Blockchain	Kumar
20	07-11-2025	08:21:18	GAT034	venkat	Blockchain	Kumar
21	10-11-2025	10:02:03	GAT032	tejas(1ga22is170)	Data Science	Mrs. Priya
22	10-11-2025	10:02:03	GAT033	thippesh	Data Science	Mrs. Priya
23	10-11-2025	10:02:03	GAT034	venkat	Data Science	Mrs. Priya
24	10-11-2025	10:02:03	GAT004	bharath(1ga23is401)	Data Science	Mrs. Priya
25	10-11-2025	10:02:03	GAT005	bhuvan(1ga23is402)	Data Science	Mrs. Priya
26	10-11-2025	10:02:03	GAT006	charan	Data Science	Mrs. Priya
27	10-11-2025	10:02:03	GAT007	darshan	Data Science	Mrs. Priya
28	10-11-2025	10:02:03	GAT008	deepak	Data Science	Mrs. Priya

Fig. 3. Attendance marked in CSV file.

Fig. 3. shows that student details are stored in the csv file which maintains a Quick and easy to access records of the students.

IV. Algorithm

Input: Video Stream from Camera

Output: Updated Attendance CSV

1. Initialize Video Source
2. Load Known Encodings from *encodings.pkl*
3. **while** (Stream Active) **do**:
4. Capture Frame F
5. Rescale F & Convert to RGB
6. Detect Faces using YOLOv8
7. **for** each detected face **do**:
8. Compute Encoding E_live
9. Match E_live with Database Encodings
10. **if** match found (Distance < 0.6):
11. Identify Student ID
12. Log Name and Timestamp to CSV
13. Display Label on Frame
14. **end for**
15. Render Processed Frame to Dashboard
16. **end while**

V. RESULTS AND DISCUSSION

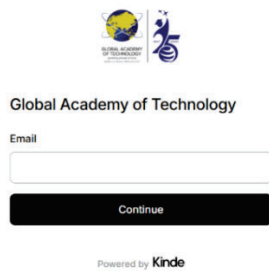


Fig. 4. Login Interface for Smart Class Management System

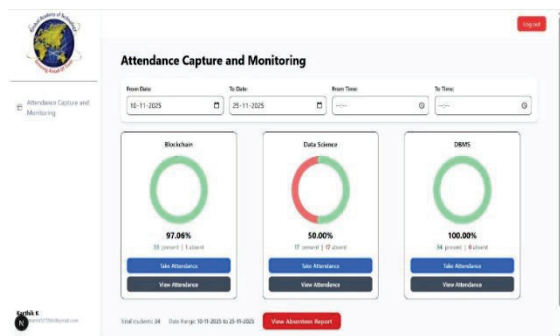


Fig. 5. Dashboard of All Subject Attendance

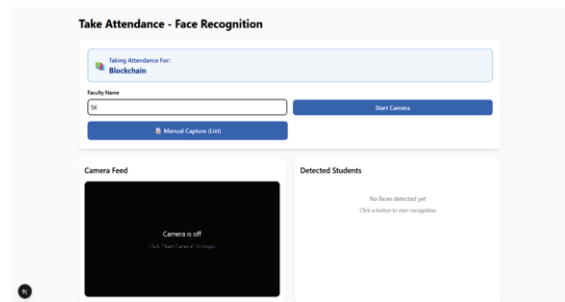


Fig. 6. Face Recognition-Based Attendance Capture Interface

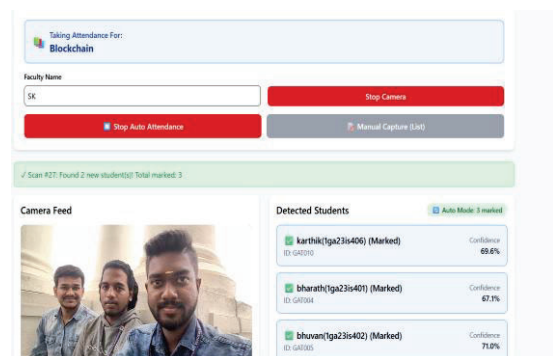


Fig. 7. Real-Time Student Detection and Automatic Attendance Marking

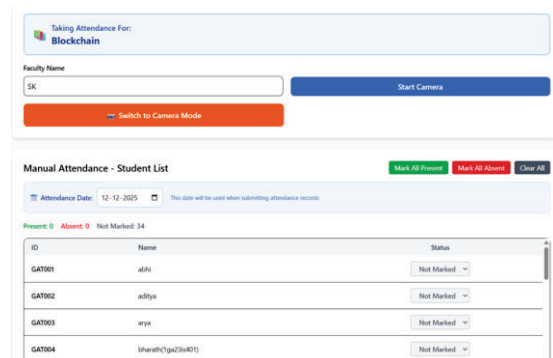


Fig. 8. Manual Attendance Entry Interface

TABLE I. COMPARISON BETWEEN PROPOSED YOLO MODEL AND TRADITIONAL METHODS

Model	Detection	Recognition	Accuracy	Complexity
Proposed	YOLOv8	ResNet-512	98.4%	Moderate - High
Siddiqui [9]	Haar Cascade	LBPH	94.0%	Low
Alfattama [16]	MTCNN	VGG-Face	90.0%	High

Note: Accuracy values are reported from the respective referenced studies and are provided for comparative analysis.

In this system, YOLOv8 is used for face detection, while ResNet-512 is used for recognition. Together, these models give reliable performance in real-time conditions. In our observations, the model achieved an accuracy of around 98.4% under normal classroom conditions, while still maintaining manageable computational requirements.

A similar attendance approach using traditional methods was presented by Siddiqui [9], where Haar Cascade was used for detection and LBPH for recognition. Although this method is simpler and computationally less demanding, it reported a lower accuracy of about 94% and showed higher sensitivity to variations such as lighting changes and face orientation.

On the other hand, Alfattama [16] explored a deep learning-based solution using MTCNN for face detection and VGG-Face for recognition. While this approach improves feature extraction capability, it introduces higher computational complexity, making it less suitable for real-time deployment on systems with limited resources.

From these comparisons, we can see that the proposed system provides a better balance between accuracy and efficiency when compared to both traditional and more computationally intensive deep learning approaches.

VII. CONCLUSION

In this work, a system is developed to automate attendance using facial recognition. By integrating computer vision techniques with a simple dashboard interface, this system reduces manual effort and improves accuracy. The use of deep learning models helps the system perform reliably under different conditions, including lighting variations and minor pose changes. Overall, this system works efficiently and can be scaled for use in different classroom environments. [6], [11], [12], [3], [20].

REFERENCES

- [1] [1] Madhur Joshi, Sunil Sikka, Manoj Pandey, Sweta Tripathi, "Advance Attendance management Through Face Recognition", "IEEE"., pp.954-959,2025, doi:10.1109/ISPCC66872.2025.11039556
- [2] [2] S. Krishna, "Automatic attendance system," *ResearchGate*, pp. 1–9, 2025.
- [3] [3] A. Kamble, V. Bisen, and A. Maind, "Student attendance system using blockchain: secure and automated attendance system using blockchain and facial recognition," *ResearchGate*, pp. 1–13, 2025.
- [4] [4] O. Oshin, J. Amenaghawon, F. Moninuola, and B. Idowu-Bismark, "Class attendance system using facial recognition," *ResearchGate*, pp. 1589–1595, 2025.
- [5] [5] H. A. Fernand, "Enhancing primary schools attendance systems with facial recognition," *ResearchGate*, pp. 1–140, 2025.
- [6] [6] P. Riyan and K. A. Kumar, "Smart classroom management: AI-driven attendance, resource optimization, and analytics," *IRJET*, pp. 383–388, 2025.
- [7] [7] M. Kim, Y. Su, F. Liu, A. Jain, and X. Liu, "KeyPoint relative position encoding for face recognition," *IEEE*, pp. 244–255, 2024.
- [8] [8] M. Mahboob, "Attendance management system using face recognition," *ResearchGate*, pp. 69–74, 2024.

- [10] [9] M. H. Siddiqui, "Facial recognition-based attendance system," *ResearchGate*, pp. 1–10, 2024.
- [11] [10] S. B. Wanjare *et al.*, "Contactless smart attendance system using facial recognition...", *IJEAT*, vol. 13, no. 2, pp. 13–16, 2024.
- [12] [11] E. Figueroa, E. Batista, R. Palau, O. Unciti, M. Ferre, and A. Martinez-Balleste, "The use of artificial intelligence techniques in smart classrooms in its infancy," *IEEE Access*, vol. 12, pp. 125179–125193, 2024.
- [13] [12] M. M. Ahmed, A. A. Ali, and A. A. Afwah, "Smart classroom automation system," *ResearchGate*, pp. 131–138, 2024.
- [14] [13] A. K. Sirivarshitha, K. Sravani, K. S. Priya, and V. Bhavani, "An approach for face detection and face recognition using OpenCV and face recognition libraries in Python," *IEEE*, pp. 1274–1278, 2023.
- [15] [14] P. B. Jha, A. Basnet, B. Pokhrel, B. Pokhrel, G. K. Thakur, and S. Chhetri, "An automated attendance system using facial detection and recognition technology," *ResearchGate*, pp. 103–120, 2023.
- [16] [15] E. Mgbeahuruike and B. Adeoti, "Student attendance monitoring system using facial recognition," *ResearchGate*, pp. 48–59, 2023.
- [17] [16] S. Alfattama, P. Kanungo, and S. K. Bisoy, "Face recognition from partial face data," *IEEE*, pp. 20–28, 2021.
- [18] [17] A. Sherpa, A. R. Subba, J. A. Rai, N. Tamang, P. K. Bhutia, S. Gurung, Y. Lepcha, and S. Pradhan, "Automated attendance system using facial structure," *SpringerLink*, pp. 1–19, 2020.
- [19] [18] G. G. Shingan, P. S. Kadam, A. M. Mahadik, D. A. Ghadage, and N. R. Suradkar, "Automated supervision allocation system," *IRJETS*, pp. 1611–1625, 2020.
- [20] [19] S. Haji and A. Varol, "Real time face recognition system (RTFRS)," *ResearchGate*, pp. 107–111, 2016.
- [21] [20] Ms. S Hemavathi, "AI-Powered Security and Attendance Management System Using Deep Learning and Facial Recognition" *ResearchGate*, pp. 1–7, 2024.