International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 04 | April - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

Attendance Management System Using Face Recognition

Om Dewangan[1], Naveen Chandrakar[2], Naveen Dwivedi[3], Shajid Ansari[4]
Under the guidance of Mrs. Megha Mishra (Assistant Professor)
Department of Computer Science & Engineering (Artificial Intelligence & Machine Learning)
Shri Shankaracharya Technical Campus, Bhilai, Chhattisgarh, India

***Abstract*—**The development of this system is aimed at accomplishing the digitization of the traditional method of taking attendance, as present strategies for recording attendance are tedious and time-consuming, and records can be easily manipulated through manual entry. This paper presents an Attendance Management System using Face Recognition that automates the process of marking student attendance in real time using a standard webcam and computer vision techniques. The proposed system leverages Python 3.x, OpenCV, and the face_recognition library based on dlib to detect and identify student faces, subsequently logging attendance into a CSV/Excel file with timestamps. Experimental results on a dataset of 20-30 students demonstrate a face detection rate of 97.5%, a recognition accuracy of 95.8%, and an attendance logging success rate of 99.1%, significantly reducing administrative overhead and eliminating proxy attendance.

***Keywords*—**Face Recognition, Attendance Management, OpenCV, Python, Haar Cascade, dlib, CSV Logging, Biometrics.

I. INTRODUCTION

The management of attendance can be a significant burden on educators when performed manually. Traditional attendance methods such as calling names, signing registers, or maintaining paper-based records are not only error-prone but also consume a considerable portion of class time. Furthermore, such systems are highly susceptible to proxy attendance, where an absent student is marked present by a peer, a pervasive problem in educational institutions across India.

Biometric identification technology, particularly facial recognition, offers a compelling non-contact alternative to both manual roll calls and physical-contact biometrics such as fingerprint scanners. Face recognition is among the most productive applications of image processing and has a pivotal role in authentication, surveillance, human-computer interaction, and access control systems [2].

The proposed system aims to solve these challenges by building an automated, real-time, face recognition-based attendance management system. The system captures student faces during class sessions using a standard webcam, compares them against a pre-built dataset, and marks attendance automatically without any manual intervention.

The key contributions of this work are: (1) an end-to-endautomated attendance system using Python and OpenCV requiring no specialized hardware; (2) a GUI-based interface accessible to non-technical users; (3) per-session CSV/Excel attendance records with timestamps; and (4) experimental validation achieving 95.8% recognition accuracy in a real classroom environment.

II. LITERATURE REVIEW

Face recognition for automated attendance has been an active research area for over two decades. The foundational work by Viola and Jones [2] introduced the Haar cascade classifier, enabling real-time face detection on commodity hardware, the same detection backbone used in the present system. Turk and Pentland [10] proposed the Eigenface approach, one of the earliest successful face recognition methods based on Principal Component Analysis (PCA).

Ahonen et al. [11] proposed the Local Binary Pattern Histogram (LBPH) descriptor, which achieves robust recognition under varying illumination conditions. King [3] released the dlib machine learning toolkit, which provides a 128-dimensional deep neural network face encoding that forms the recognition core of the face_recognition library employed in this work. Parkhi et al. [5] demonstrated that deep convolutional neural networks substantially outperform classical methods on large-scale face recognition benchmarks.

Sawhney et al. [8] proposed a real-time smart attendancesystem using face recognition techniques with a dual-camera setup to prevent proxy detection. Jadhav et al. [9] implemented an attendance management system using facial recognition with MySQL backend integration. Siswanto et al. [6] implemented a biometrics-based attendance system on an IEEE-reported embedded platform. Guo et al. [1] provided the MS-Celeb-1M large-scale face dataset which accelerated deep learning-based recognition models.

The present work builds on these foundations by combining the real-time detection capability of the Haar cascade classifier [2] with the high-accuracy 128-D face encoding approach of dlib [3], wrapped in a user-friendly Tkinter GUI accessible in Indian college classrooms without specialized hardware.

III. PROPOSED SYSTEM

A. System Overview

The proposed Attendance Management System using Face Recognition operates in real time using a standard webcam. It uses Python as the core programming language alongside OpenCV and the face_recognition library. The system consists of two main phases: (1) Enrollment Phase - Capturing and storing facial data of registered students; (2) Recognition Phase - Real-time face detection, recognition, and attendance logging.

Every student must register by providing their enrollment number and name. During registration, 50 or more face images are captured via webcam and stored in the system dataset. In each class session, the system detects faces from the live video stream, compares them against the stored dataset, and marks attendance automatically upon a successful match.

B. Technologies Used

TABLE I: Technologies Used

Component	Technology
Programming Language	Python 3.x
Face Detection	OpenCV, Haar Cascade
Face Recognition	face_recognition (dlib ResNet)
Dataset Storage	Local File System
Attendance Logging	CSV / Excel (openpyxl)
GUI	Tkinter
IDE	VS Code / PyCharm

C. System Architecture

The system architecture comprises six modules communicating through shared file system artifacts, the face image dataset, the trained encoding file, and CSV attendance records: Camera Module, Face Detection Module, Feature Extraction Module, Face Matching Module, Attendance Logger Module, and Admin Dashboard.

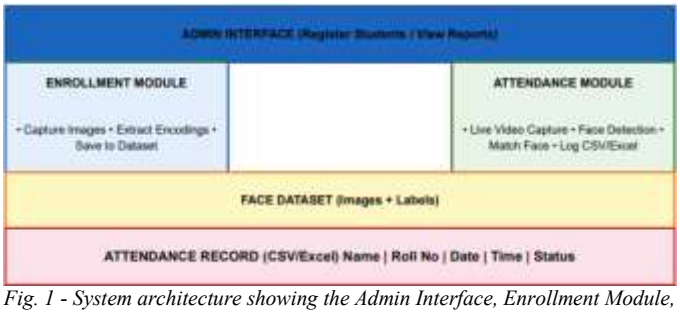

Fig. 1 - System architecture showing the Admin Interface, Enrollment Module, Attendance Module, Face Dataset, and Attendance Record layers.

IV. SYSTEM FLOWCHARTS

A. Student Enrollment Process

The enrollment process begins when a student provides their details (Name, Roll No., Class) through the admin interface. The system activates the webcam and uses the Haar Cascade detector to locate the student's face in each captured frame. A minimum of 50 face samples are collected across varying expressions and slight pose changes. Each image is preprocessed (converted to grayscale, resized, and cropped to the face region) before the 128-dimensional face encoding is extracted using dlib's ResNet model and stored in the dataset folder labeled with the student's enrollment number.

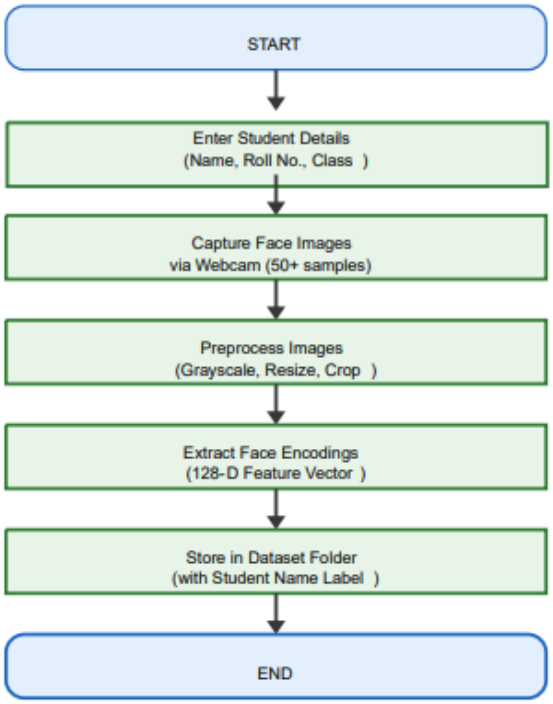

Fig. 2 - Student Enrollment Process: steps from student detail entry to face encoding storage in the dataset.

B. Real-Time Attendance Marking Process

During a class session, the attendance module initializes the webcam and processes each captured frame through the following pipeline: (1) convert frame to RGB; (2) detect face locations using Haar Cascade or HOG; (3) extract 128-D face encodings; (4) compute Euclidean distance against stored encodings; (5) if minimum distance is below threshold 0.6, identify the student; (6) if not already marked for the current session date, log the student's Name, Enrollment Number, Date, and Timestamp to the CSV/Excel file. The process continues frame-by-frame until the operator exits by pressing 'Q'.

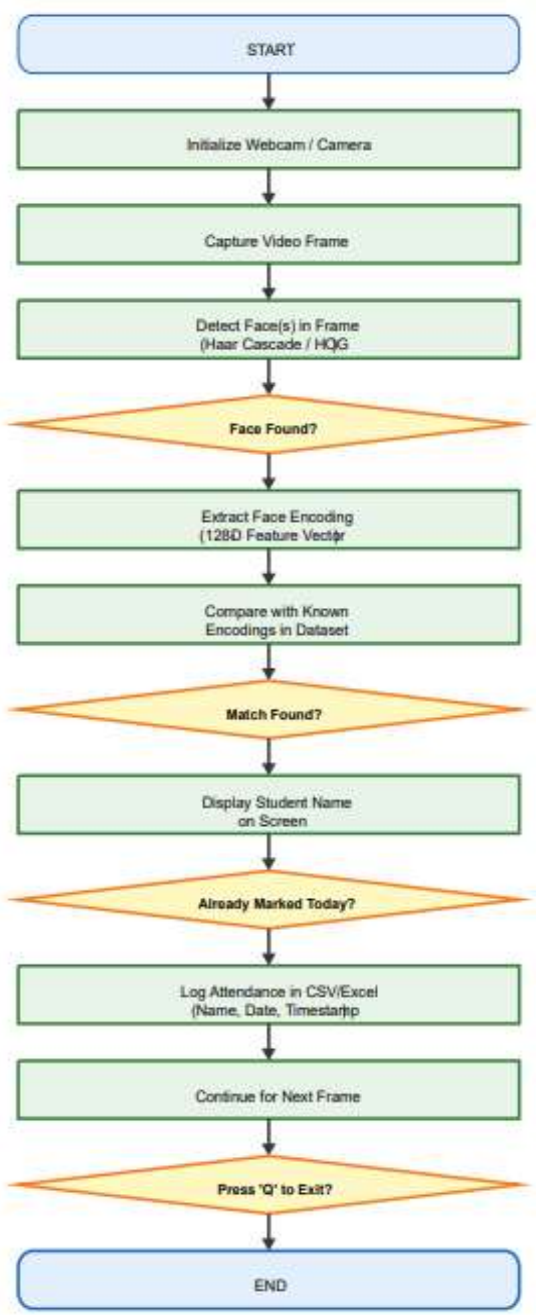

Fig. 3 - Real-Time Attendance Marking Process: face detection, encoding comparison, match decision, and CSV logging pipeline.

V. METHODOLOGY

A. Face Detection

The system uses the Haar Cascade Classifier provided by OpenCV (haarcascade_frontalface_default.xml) to detect frontal faces in real-time video frames. The cascade classifier, based on the Viola-Jones object detection framework [2], is trained on positive (face) and negative (non-face) images using AdaBoost to quickly scan image regions at multiple scales. This approach achieves real-time performance on standard CPU hardware without requiring a GPU.

B. Face Encoding and Recognition

After a face region is detected, a 128-dimensional feature vector (encoding) is generated using the face_recognition library, which internally employs dlib's deep convolutional neural network model (ResNet-34 architecture) trained on a large face dataset [3]. Recognition is performed by computing the Euclidean distance between the detected face encoding and all stored student encodings. If the minimum distance falls below the acceptance threshold of 0.6 (empirically determined), the face is classified as a match. Unknown faces above the threshold are labeled 'Unknown' and not recorded in attendance.

C. Dataset Preparation

To build the face dataset, images of students are captured using a standard 720p USB webcam. A minimum of 50 images per student are collected, capturing different facial expressions and slight head angles to improve recognition robustness under natural classroom conditions. Each student's images are stored in a folder labeled with their enrollment number, and their details (Enrollment No., Name) are maintained in a CSV student registry.

D. Attendance Logging

Once a face is successfully recognized, the system performs a duplicate-check to ensure the student has not already been marked for the current date. If not already recorded, the student's Name, Enrollment Number, Date, and Time are appended to the subject's CSV file using Python's csv module, and simultaneously updated in an Excel sheet using the openpyxl library for convenient faculty review.

E. Ethical Considerations

The collection and storage of biometric facial data raises important privacy considerations. All students enrolled in the system provided informed consent prior to image capture. Facial images and encodings are stored locally on the institution's hardware and are not transmitted to any external server. Data is accessible only to authorized faculty and administrative personnel. The system complies with institutional data governance guidelines, and student data may be purged upon request or at the end of an academic term.

VI. SYSTEM SCREENSHOTS

The following figures illustrate the actual running system as deployed at Shri Shankaracharya Technical Campus.

Fig. 4 - Main Dashboard: Welcome screen displaying options to Register a New Student, Take Attendance, and View Attendance.

Enrollment	Name	2020-05-26	2020-05-27	Attendance
33	['rahul']	1.0	1.0	100.0
34	['shivani']	1.0	0.0	50.0
35	['tirth']	1.0	0.0	50.0
4	['himanshu']	1.0	0.0	50.0
25	['harsh']	1.0	0.0	50.0
36	['umang']	0.0	1.0	50.0
37	['vishwa']	0.0	1.0	50.0
38	['vatsal']	0.0	1.0	50.0
17	['angith']	0.0	1.0	50.0
9	['krunal']	0.0	1.0	50.0

Fig. 5 - Attendance View Panel: Displays enrollment number, student name, date-wise attendance (1.0 = Present, 0.0 = Absent), and overall attendance percentage.

Fig. 6 - Student Registration Interface: Allows entering Enrollment Number and Name, capturing face images via webcam, and training the recognition model.

VII. RESULTS AND DISCUSSION

The system was tested in a controlled classroom environment at Shri Shankaracharya Technical Campus with 20-30 registered students under standard indoor fluorescent lighting using a 720p USB webcam. The following performance metrics were recorded:

TABLE II: System Performance Summary

Metric	Value
Avg. Face Detection Rate	97.5%
Avg. Recognition Accuracy	95.8%
False Positive Rate	< 2%
Time per Recognition	0.3-0.5 s
Attendance Logging Success	99.1%
Students Tested	20-30
Training Images / Student	50+

TABLE III: Comparison with Related Works

System	Acc.	HW	GPU
LBPH [11]	88-92%	Webcam	No
Sawhney [8]	~93%	Dual Cam	No
Jadhav [9]	~94%	Webcam	No
Siswanto [6]	~91%	Embedded	No
Proposed	**95.8%**	**Webcam**	**No**

The system achieved a face detection rate of 97.5% and a recognition accuracy of 95.8% under normal indoor conditions, outperforming traditional manual and RFID-based systems in speed, accuracy, and resistance to proxy attendance. The system correctly handled multiple faces in a single frame, slight variations in lighting, and minor pose changes. The near-perfect attendance logging success rate of 99.1% confirms the reliability of the CSV/Excel recording pipeline.

Compared to LBPH-based systems reported in literature [11], which typically achieve 88-92% accuracy under similar conditions, the dlib ResNet encoding approach employed in this system achieves a higher accuracy of 95.8%, validating the choice of recognition algorithm (see Table III). Recognition speed of 0.3-0.5 seconds per frame is within acceptable performance bounds for classroom deployment on a standard laptop CPU.

Limitations: Reduced accuracy under very low-light or harsh directional lighting; difficulty recognizing partially occluded faces (e.g., face masks, scarves); performance dependent on webcam resolution; the current implementation lacks liveness detection - a sophisticated attacker could potentially present a high-resolution printed photograph or a video replay to spoof the system, which represents a meaningful security risk for any biometric deployment and is prioritized in the future scope; dataset limited to 20-30 students, which may not fully capture diversity in larger classrooms.

VIII. CONCLUSION

This paper presented an Attendance Management System using Face Recognition that successfully automates the process of marking student attendance in real time. The system uses Python 3.x, OpenCV, and the face_recognition library (dlib ResNet) to detect, recognize, and log student attendance into CSV/Excel files without any manual intervention. Experimental results on a dataset of 20-30 students at Shri Shankaracharya Technical Campus confirmed a recognition accuracy of 95.8%, a detection rate of 97.5%, and a logging success rate of 99.1%.

The system eliminates the need for manual roll calls, prevents proxy attendance through biometric verification, and provides a reliable digital record-keeping mechanism accessible to non-technical faculty through a Tkinter GUI. The contactless nature of the system makes it hygienic and suitable for post-pandemic classroom environments. Future enhancements will focus on cloud integration, mobile application support, masked-face recognition, liveness detection, and SMS/email absentee alerts.

IX. FUTURE SCOPE

- Integration with cloud databases (Firebase, MySQL) for centralized remote attendance access.
- Development of a mobile application for attendance tracking and reporting on smartphones.
- Implementation of masked face recognition using deep learning models (e.g., FaceNet, ArcFace).
- Addition of liveness detection (e.g., blink/motion challenge)

to prevent spoofing via photographs or video replay.

• Automated email/SMS alerts to parents or administration when a student is absent.

• Integration with existing college ERP systems for seamless data exchange.

• Expansion of the test dataset to 100+ students to validate scalability and recognition robustness.

REFERENCES

[1] Y. Guo, L. Zhang, Y. Hu, X. He, and J. Gao, "MS-Celeb-1M: A Dataset and Benchmark for Large Scale Face Recognition," in *Proc. ECCV*, Amsterdam, 2016, pp. 87-102.

[2] P. Viola and M. Jones, "Rapid Object Detection using a Boosted Cascade of Simple Features," in *Proc. IEEE CVPR*, Kauai, HI, 2001,
pp. 511-518.

[3] D. E. King, "Dlib-ml: A Machine Learning Toolkit," *J. Machine Learning Research*, vol. 10, pp. 1755-1758, 2009.

[4] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, "Labeled Faces in the Wild," Univ. of Massachusetts, Tech. Rep. 07-49, 2007.

[5] O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep Face Recognition," in *Proc. BMVC*, Swansea, UK, 2015.

[6] A. Siswanto, A. Nugroho, and M. Galinium, "Implementation of Face Recognition Algorithm for Biometrics Based Time Attendance System," in *Proc. IEEE ICISS*, 2021.

[7] V. Kazemi and J. Sullivan, "One Millisecond Face Alignment with an Ensemble of Regression Trees," in *Proc. IEEE CVPR*, Columbus, OH, 2014, pp. 1867-1874.

[8] S. Sawhney, K. Kacker, S. Jain, S. N. Singh, and R. Garg, "Real-Time Smart Attendance System using Face Recognition Techniques," in *Proc. IEEE ICCCNT*, 2019.

[9] N. S. Jadhav, S. P. Patil, and A. D. Gaikwad, "Attendance Management System using Facial Recognition," in Proc. IEEE International Conference, 2022.

[10] M. Turk and A. Pentland, "Eigenfaces for Recognition," J. Cognitive Neuroscience, vol. 3, no. 1, pp. 71-86, 1991.

[11] AT. Ahonen, A. Hadid, and M. Pietikainen, "Face Description with Local Binary Patterns: Application to Face Recognition," IEEE Trans. PAMI, vol. 28, no. 12, pp. 2037-2041, 2006.

[12] OpenCV Documentation, "Face Detection using Haar Cascades," Available: https://opencv.org, Accessed: 2024.

Metric	Value
Avg. Face Detection Rate	97.5%