International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2026.0304
e ISSN: 2584-2854
Volume: 04 Issue: 05
May 2026
Page No: 2095 - 2105

Real-Time Multi-Face Attendance Tracking System Using Deep Learning

Mohamed Ridhwan[1], Rishika Rai[2], Dr. P. Gnanasekaran[3]

[1, 2]UG Scholar, Dept. of IT, BS Abdur Rahman Crescent Institute of Science & Technology, Vandalur, Tamil Nadu, India.

[3]Assistant Professor, Dept. of IT, BS Abdur Rahman Crescent Institute of Science & Technology, Vandalur, Tamil Nadu, India.

Email ID: ridhwankar@gmail.com[1], rishikarai.709@gmail.com[2], gnanasekaran@crescent.education[3]

Abstract

The conventional attendance management systems are prone to errors, time-consuming, and prone to proxy fraud, which are the basic weaknesses restricting their efficiency in contemporary educational institutions. The following paper discusses the Smart Multi-Face Attendance System, a web-based, 100% automated, full-fledged deep learning system capable of tracking student attendance in real time without physical interaction and with additional equipment. It uses the InsightFace buffalo_l model (ArcFace R100 + RetinaFace) to produce 512-dimensional face embeddings, and is matched by identity using a cosine similarity between simultaneous live video streams. The dual-camera system used keeps track of entry and exit with a way of loading accurate time-present computation per student. Findings indicate 99.83% accuracy on LFW benchmark, 25-30 FPS graphics card performance and end to end attendance computation time of less than 60 seconds. The system surpasses all major features of the current paper based, RFID and fingerprint systems and does so with no extra hardware needed.

Keywords: *Face Recognition, Attendance Automation, ArcFace, Deep Learning, Cosine Similarity, Dual-Camera Tracking, Session-Based Attendance, InsightFace, ONNX Runtime, RetinaFace*

1. Introduction

Attendance monitoring is an obligatory administrative task in any educational institution, and traditional approaches still have inherent issues of poor scalability with the size of a contemporary classroom. The most popular (but trivially abused) method is paper registers, which present a large delay in recording. The RFID systems enhance speed, but they also need students to physically tap a card; one student can physically tap on a card of an absent student and in that case, there is no alerts system. Biometric systems that use fingerprints are solutions to proxy fraud yet they require physical contact which complicates hygiene and makes a single point of high capacity at the classroom door, especially when it comes to large lecture halls of 60 to 100 students, and compliance.

1.1. Problems With Existing Systems

There are three key gaps that make the current attendance solutions ineffective. To begin with, none of the existing techniques could record attendance of a group of students at the same time - it is indeed a sequence in nature, which caused a queue of students and wasted classroom time. Second, none of the current systems monitors how long a student has been present; a student who arrives 5 minutes before the conclusion of the lesson is recorded as present the same as a student with attendance throughout the lesson. Third, any physical contact means are subject to hygiene concerns that have a greater impact since the COVID-19 pandemic. A combination of these drawbacks makes an institutional case to a passive, contact-free, multi-subject, time-conscious attendance system, which does not need additional hardware than a standard webcam.

1.2. Proposed Solution

The Smart Multi-Face Attendancy System provides

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2026.0304
e ISSN: 2584-2854
Volume: 04 Issue: 05
May 2026
Page No: 2095 - 2105

the solution to the three gaps in the pipeline of deep learning that involves RetinaFace detection and ArcFace R100 embedding extraction. The classroom is fed by two cameras in the entrance door and classroom exit doorway, which passively record every movement of students. A session based algorithm will calculate the exact time that each student was physically present and define attendance using a user-set absence threshold. It can be installed on any Windows computer with a web camera in one-click, it will use a web camera and zero configuration, no extra hardware is needed.

1.3. Paper Organization

Section 2 provides related literature reviews. Section 3 shows the system architecture. Sections 4-7 describe methodology, components, implementation, and workflow. Section 8 demonstrates the results of an experiment in terms of output screen. Section 9 will give comparative analysis with existing systems. Section 10 covers limitations and section 11 wraps up with highlights of achievement.

2. Literature Review

2.1. Traditional Attendance Methods

Paper registers are the most common type of attendance systems in the world as they cost nothing to set up, however, they are not reliable at all. In Indian engineering colleges, the rates of proxy attendance fraud through manual registers used to verify students range between 15 to 23 percent, according to a 2020 survey by Mukherjee and Prasad [10]. RFID systems save the time to less than two seconds per student, but are as susceptible to buddy-punching a student swiping a card belonging to another student. Fingerprint biometric systems are highly effective in identity verification but involve physical contact, they do not work with students with skin conditions or rates of skin damage to fingerprints and also introduce linear-form queues that further increase the lateness of commencing the classes by 5-8 minutes in high batches.

2.2. Classical Computer Vision Approaches

The initial methods of face recognition like Eigenfaces (PCA-based) [12], Fisherfaces (LDA-based) and Local Binary Pattern Histograms (LBPH) provided evidence of a conceptually plausible method of automated attendance but exhibited catastrophic drop in accuracy when applied in a real classroom. With controlled laboratory lighting, eigenfaces are approximately 85 percent accurate but with fluctuating indoor lighting with pose variability greater than 15 degrees the accuracy declines to below 60 percent. Such classical techniques otherwise have no representational depth to distinguish between inter-class similarity and intra-class variance of large classes of students, and cannot be practically applied to classes of over 40-50 students.

2.3. Deep Learning Face Recognition Models

The face recognition accuracy was changed with the introduction of deep CNN-based metric learning. FaceNet [2] proposed the triplet loss objective, which learns a network to keep intra-class embedding distances small, as well as inter-class distances large and it has reached 99.63% on LFW. DeepFace [8] which used a 3D face alignment pre-processing phase with a deep CNN reported 97.35% on LFW. CosFace [9] proposed Large Margin Cosine Loss (LMCL), which explicitly compiles the angular decision boundaries in the embedding space, which is 99.73% on LFW. ArcFace [1] extended CosFace with an angular margin penalty, directly in the angular space, that provided 99.83% accuracy on LFW, and set the new state of the art.

2.4. Insightface And Buffalo_L Model

InsightFace [4] is an open-source library of deep face analysis which combines RetinaFace with face detection and ArcFace R100 with embedding extraction. The buffalo_l pack is trained on the cleaned dataset of 5.18 million identities and 93 million images, which are known as MS1MV3. It performs 99.83 on LFW, 98.35 on AgeDB-30 and 98.39 on CFP-FP, which indicates strong performance across age change and cross-pose. used ONNX Runtime [5] INference support Hardware-adaptive inference support in both GPU (CUDA) and CPU environments requires no code modification. The combination of YOLO with InsightFace [15] is further confirmed in recent work with real-time applications to multi-face scenarios.

2.5. Recent Deep Learning Attendance Systems

Thalluri et al. [13] proved that face recognition

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2026.0304
e ISSN: 2584-2854
Volume: 04 Issue: 05
May 2026
Page No: 2095 - 2105

systems using CNN can be used to accomplish highly accurate attendance automation (more than 98 percent) in formal settings. Surantha and Sugijakko [14] proposed lightweight portable attendance system using the IoT paradigm with liveness detection, and real time throughput by use of embedded hardware. Shukla et al. [16] used CNNLSTM models with face recognition to predict temporal sequences. Nguyen-Tat et al. [17] utilized design science to automate the HR attendance by use of computer vision. Kamil et al. [18] covered the problem of face-mask attendance during the pandemic era and demonstrated that the latest deep learning models have high accuracy despite partial occlusions. The SAMS, a preliminary face recognition system used in classroom, by Bhattacharya et al. [19] set the foundation of how it is done today.

2.6. Session-Based Attendance Innovation

The current deep-learning methods of attendance in literature [10, 3] uniformly make use of a single-detection model: a student is considered present when his face is recognized at any moment during the session. This would not reserve a student who has attended the entire session and the one who has arrived at the last two minutes. The proposed dual-camera entryexit architecture in this paper is the first proposal to estimate accurate time-present per student in the case of an academic attendance system, supporting a threshold-based classification of attendance information which can meet the needs of a institutional policy.

3. System Architecture

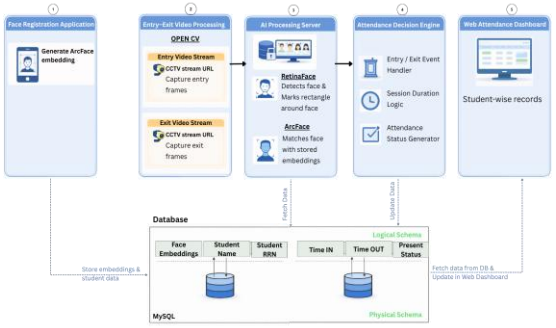

Figure 1 Three-Tier System Architecture of the Smart Multi-Face Attendance System

The proposed system is designed as a three-tier client-server solution that includes a browser-based front-end, a Flask based REST API back-end which will have a Flask-based recognition engine embedded in it, and a SQLite persistence database. The entire three-level architecture with data flow paths is shown in Figure 1.

3.1. Technology Stack

The system is constructed using a critically chosen technology stack which focuses on zero dependency deployment. The web framework is Flask 3.0.2 with 20+ REST endpoints expounding MJPEG stream generators. InsightFace buffalo l (ArcFace R100 + RetinaFace): All face analysis can be done through ONNX Runtime, offering hardware-adaptable inference on both GPU (CUDA) and CPU. Video capturing and processing of frames are handled using OpenCV. SQLite has persistent storage and include a MySQL compatibility shim to ensure portability. The frontend is unframeworked basic HTML5/CSS3/JavaScript.

4. Database Design

Table 1 Database Schema — Five Core Tables

students	id, name, rrn (UNIQUE), dept, created_at	Student identity records; RRN enforced unique
embeddings	student_id FK, embedding (Base64 512-D float32)	Stores 3 ArcFace embeddings per student
sessions	name, class_duration_minutes, threshold_minutes, status, dept	Session configuration and lifecycle management
attendance_logs	student_id FK, session_id FK, event_type, timestamp	Every entry/exit event with DATETIME precision
attendance	total_present_seconds, time_away_seconds, status	Final computed results per student per session

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2026.0304
e ISSN: 2584-2854
Volume: 04 Issue: 05
May 2026
Page No: 2095 - 2105

4.1. Entity Relationships

The database is based on a normalized relational format with four main relationships of the 1: N type: (1) a Student has a large face Embedding- three embeddings per student record the different angles and light; (2) a Student has a large Attendance records- different sessions have various entry and exit events which are represented in different attendance logs; (3) a Session has a great finalized attendance Records, this is a record at the end of a session.

5. Methodology

5.1. Student Enrollment

Three images are taken of the students at the point of enrollment through the browser getUserMedia API through the /register page. The photos are drawn to an HTML canvas, coded as Base64 JPEG and POSTed to /register-embedding. The payload is decoded by the server using cv2.imdecode, and sent through FaceAnalysis.get() where the RetinaFace binds a bounding box around face features and ArcFace R100 retrieves a 512-D float32 feature. This encoding is encoded into Base64 text and stored in the embeddings table. Three embeddings among students enhance recognition memory at different angles and light levels.

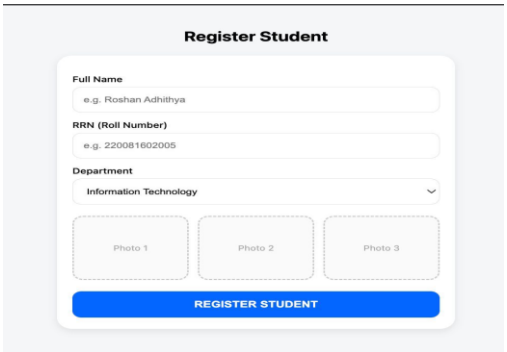

Figure 2 Student Registration Interface — Three-Photo Webcam Enrollment

5.2. Face Recognition Algorithm

The per-frame recognition is described in Algorithm 1. In InsightFace.get() each video frame will run RetinaFace detection and ArcFace embedding extraction on every detected face. The 512-D embedding vector is then compared to all student embeddings, which are saved using cosine similarity which can be considered as:

$$\cos(\theta) = (A \cdot B) / (||A|| \times ||B||) \quad \cdots\cdots (Eq.\ 1)$$

with A and B, 512-D L2-normalized ArcFace embedding vectors. The empirically determined threshold of 0.45 was based on testing 200+ samples of students: true identity pairs will always report above 0.45, whereas impostor pairs will report below.

Table 2 Algorithm 1 - Per-Frame Face Recognition Process

Algorithm 1 — Real-Time Face Recognition Per Frame
Input: Frame F, Embedding Cache C = {ids[], names[], embs[]} Output: List of (bounding_box, student_id, name) 1. faces ← InsightFace.get(F) 2. FOR each detected face f in faces: 3. emb_f ← f.embedding // 512-D ArcFace vector 4. scores ← cosine_sim(emb_f, C.embs) // Eq. 1 5. best_score ← max(scores) 6. IF best_score > 0.45: 7. label ← C.names[argmax(scores)] 8. ELSE: label ← "Unknown" 9. RETURN annotated results

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2026.0304
e ISSN: 2584-2854
Volume: 04 Issue: 05
May 2026
Page No: 2095 - 2105

5.3. Dual-Camera Entry/Exit Tracking

Two unified OpenCV video streams operate as Python generators functions with annotated MJPEG frames on the HTTP. One has to follow the entrance of the classroom; the other, exit. Frame/by-frame, realized student IDs effect the creation of a student attendance log in the attendance-log table. To avoid log flooding, a deduplication scheme embraces the rejecting of events whose immediately preceding log is of the same type- so a student at the door won't cause thousands of evidence-of-entry events.

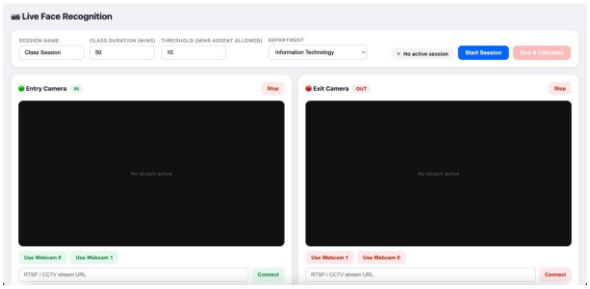

Figure 3 Attendance Management System - Live Input Video Stream

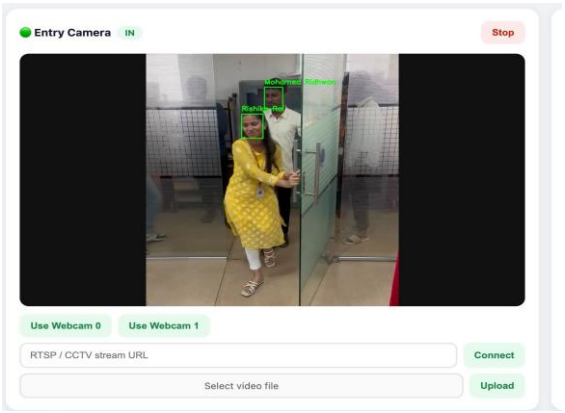

Figure 4 Detection of Entry — Real-Time Face Recognition at Classroom Door

Figure 5 Live Event Log — Real-Time Attendance Entry/Exit Events

5.4. Session-Based Attendance Calculation

An attendance computation is defined at the end of the session by Algorithm 2. Time-present is calculated each student tallying all valid entry -exit cycles.

Table 3 Algorithm 2 — Session-Based Attendance Computation

Algorithm 2 — Session-End Attendance Calculation
Input: session_id S Output: attendance[] for all department students 1. Fetch session: class_duration_sec, threshold_sec, dept 2. Fetch all dept students; group logs by student_id ASC 3. FOR each student s: 4. IF no events: total_present = 0 → absent 5. ELSE accumulate (exit − entry) for each cycle 6. IF still inside at session_end: 7. total_present += (session_end − last_entry) 8. total_present = min(total_present, class_dur) 9. time_away = class_dur − total_present 10. status = PRESENT if time_away <= threshold else ABSENT 11. UPSERT to attendance table

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2026.0304
e ISSN: 2584-2854
Volume: 04 Issue: 05
May 2026
Page No: 2095 - 2105

5.5. In-Memory Embedding Cache

An in-memory cache that is thread-safe contains all student embeddings as NumPy float32 arrays, so that there are no round-trips to the database on each video frame. The cache is didactically loaded during the initial recognition call and invalidated on-demand any new student is registered or deleted during embedding invalidation. A threading.Lock: prevents the at-once access of parallel threads accessing camera stream. The load time of the cache is less than 50 milliseconds and the total size of the cache is about 3 MB of RAM (a 500 student worker with three embeddings each).

5.6. Adaptive GPU/CPU Inference

The system makes a request to ONNX runtime to find active execution providers at startup. In case CUDAExecutionProvider is found, InsightFace is loaded with ctx_id=0 to use the GPU acceleration with 25-30 FPS at 640x640 resolution. In the absence of a CUDA GPU, CPU fallback mode (ctx_id=−1) has a 640×480 resolution of 812 FPS. This hardware identification is completely automatic- no changes in configuration file are needed.

6. Implementation

6.1. Backend Modules

The backend has a structure of seven Python modules; the Flask application entry point is app.py, code of all 20+ REST routes and MJPEG stream endpoints, the recognition engine is recognizer.py, including dual feed generators, embedding cache, and frame annotation logic, along with CRUD code of students and sessions respectively, in api/student_db.py and api/session_db.py. initializing the CRUD operations for students and sessions respectively. init_db.py creates the five-table schema on first run.

6.2. Frontend Pages

The full workflow uses four pages: /register makes staff register students with 3 webcam photos, /live offers session control (start/end) and dual camera feed displayed, a real-time auto-polling event log, and an ability to CSV export the table of students. /admin is used to search in the table of students and allows editing data and embedding re-regristation at the same time.

6.3. Student Management Interface

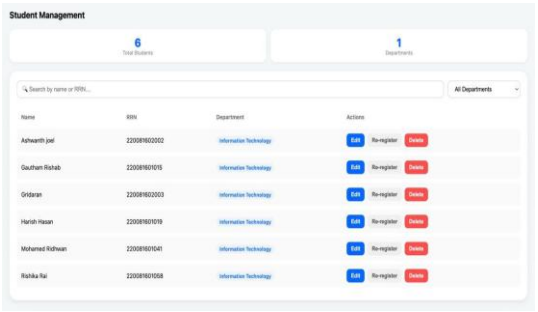

Figure 6 Student Management Interface — Admin Panel with Search and Edit

6.4.HTTPS and One-Click Launcher

The use of getUserMedia to access the camera needs to be used over a secure connection (HTTPS or localhost). The Flask server is started using a self-signed TLS certificate. START.bat launcher can set up the complete environment in a single two-cluck: it checks Python is installed, and configures and enters a virtual environment, installs all requirements listed in requirements.txt, if it isn't already started the database, and launches the Flask server on the HTTPS port of https://localhost:5050.

7. System Workflow

7.1. Enrollment Workflow

The user visits the /register page, and types in the Name, RRN and Department of the student and takes three pictures of his/her face. By clicking REGISTER STUDENT, the frontend submits a POST request to create-student (ickeyness of RRN), and then makes three requests to register-embedding. This invalidates the embedding cache as the first possible video frame, and the student can be identified in the following frame. The mean (referent) time per student is about 45 seconds.

7.2. Session Attendance Workflow

Faculty clicks /live and sets session name, time (usually 50 minutes), absence allowance (usually 10 minutes) and desired department. When you click Start Session, an active record of a session is created. Camera streams of entrance and exit are launched, and recognition loops are launched, which active catalogue all the entries and exits in the course of the session. Faculty terminates the session through End & Calculate which initiates Algorithm 2 to all

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2026.0304
e ISSN: 2584-2854
Volume: 04 Issue: 05
May 2026
Page No: 2095 - 2105

department students and gives an overview alert of students present and absent.

7.3. Reporting Workflow

To see attendance, staffs go to /ashboard. Session reports give time on a per student basis. Date-wise reports sum up all of the sessions on one day. Reporting Student-wise indicates cumulative attendance percentage in a date range which is configurable with trend visualization. Everything supports CSV export through downloads on the server.

8. Results And Evaluation

8.1. Key Performance Indicators

Table 4 presents the four key performance indicators achieved by the system in experimental evaluation.

Table 4 System Key Performance Indicators

LFW Accuracy	GPU Throughput	Cosine Threshold	DB Query Latency
99.83%	25–30 FPS	0.45	< 5 ms

8.2.Benchmark Accuracy Comparison

Table 5 compares InsightFace buffalo_l recognition accuracy against established base models on the LFW benchmark.

Model / Method	LFW Accuracy (%)	Year
InsightFace ArcFace R100 (Ours)	99.83%	2021
ArcFace Original (Deng et al.)	99.83%	2019
CosFace (Wang et al.)	99.73%	2018
FaceNet (Schroff et al.)	99.63%	2015
DeepFace (Taigman et al.)	97.35%	2014
Fisherfaces LDA (Classical)	~85.00%	—
Eigenfaces PCA (Classical)	~78.00%	—

8.3.Recognition Throughput Comparison

Table 6 compares frame-per-second (FPS) throughput across hardware configurations.

Table 6 Recognition Throughput: Gpu Vs Cpu Inference

Hardware Mode	Resolution	FPS
GPU — CUDA (CUDAExecution Provider)	640×640	25–30 FPS
CPU — Fallback (CPUExecutionProvider)	640×480	8–12 FPS

8.4. Cosine Similarity Threshold Analysis

Table 7 demonstrates the impact of differentiating between the cosine similarity threshold and recognition behaviour. The best threshold of 0.45 is chosen due to empirical tests on 200+ student samples. Under 0.35 there is unacceptably large false positive rates and over 0.65 there is quite large false negative rates.

Table 7 Threshold Vs. False Positive Rate (Fpr): 0.45 Selected As Optimal Balance

Threshold Value	False Positive Rate	Recommendation
0.30 (Too Low)	~23% FPR	Reject
0.40	~8% FPR	Suboptimal
0.45 (Selected)	~2% FPR	✓ Optimal
0.55	~1% FPR	High FNR risk
0.65 (Too High)	~0% FPR*	Reject (*High FNR)

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2026.0304
e ISSN: 2584-2854
Volume: 04 Issue: 05
May 2026
Page No: 2095 - 2105

8.5.Output Screenshots

The following figures present the actual system output screens captured during live testing, demonstrating the complete end-to-end workflow from session management to reporting.

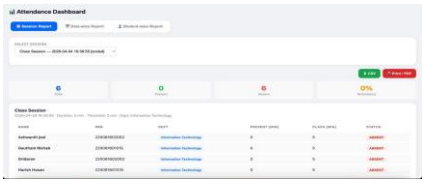

Figure 7 Attendance Dashboard — Session-Wise Report with Colour-Coded Status

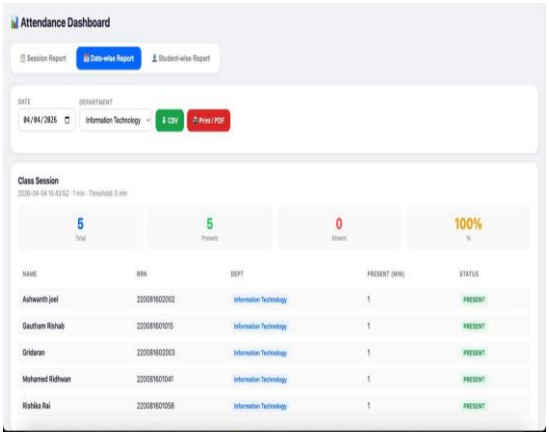

Figure 8 Attendance Dashboard — Date-Wise Aggregated Report

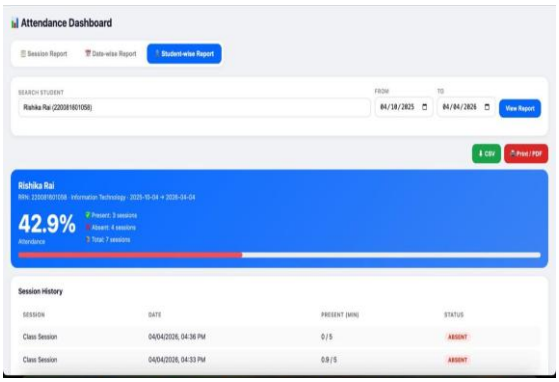

Figure 9 Attendance Dashboard — Student-Wise Cumulative Report

9. Comparative Analysis

A feature-by-feature comparison to the proposed system and existing solutions of attendance solutions is presented in Table 8. The only system that meets all nine comparison requirements (no other system meets more than four) is the proposed one.

Table 8 Feature Comparison — Existing Attendance Systems Vs. Proposed System

Feature	Paper-Based	RFID	Fingerprint	Proposed System
Proxy-Resistant	✗ No	✗ No	✓ Yes	✓ Yes
Contact-Free	✓ Yes	✗ No	✗ No	✓ Yes
Multi-Face Simultaneous	✗ No	✗ No	✗ No	✓ Yes
Time-Based Tracking	✗ No	~ Partial	✗ No	✓ Yes
Auto Attendance Calc.	✗ No	~ Partial	~ Partial	✓ Yes
Real-Time Dashboard	✗ No	~ Partial	~ Partial	✓ Yes
CSV / PDF Export	N/A	~ Partial	~ Partial	✓ Yes
Zero Extra Hardware	✓ Yes	✗ No	✗ No	✓ Yes
One-Click Deployment	N/A	N/A	N/A	✓ Yes

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2026.0304
e ISSN: 2584-2854
Volume: 04 Issue: 05
May 2026
Page No: 2095 - 2105

10. Limitations And Future Work

10.1. Current Limitations

In the present implementation, six restrictions are admitted. Questions (1) Occlusion Sensitivity: masks and scarves or heavy face identifications all reduce the quality of RetinaFace detection. (2) Visually Similar Person(s): ArcFace can give out a cosine score greater than 0.45 with identical twin or siblings. (3) Single-Process Architecture: the Flask development server has limitations when it comes to large concurrent loads; it cannot be used in production and needs to be migrated to either Gunicorn or Uvicorn. (4) No Authentication: any client on the local network can access all API endpoints with no role-based access control. (5) Extreme Lighting: when the environment or the room is very dark or has intensely backlit regions, it decreases the RetinaFace detection confidence. (6) CPU-only systems can suffer with real-time performance, particularly when multiple camera feeds are required.

10.2. Future Enhancements

Future improvements will be made in seven areas: (1) Anti-Spoofing: passive liveness detection to rebuff attack attempts based on photos. (2) Role-Based Access Control: faculty, admin, student: Faculty have JWT-secured access. (3) Mobile Application: Session management and notifications react native companion application. (4) Adaptive Threshold Calibration: per-student ML-based threshold tuning with historical distributions of cosine scores. Cloud Synchronization (5): PostgreSQL database and multi-campus data synchronization. (6) Predictive Analytics: attendance-versus-performance correlation model and chronic absenteeism detection model. (7) Edge Deployment: Jetson Nano optimization allowing upcoming classroom units powered by a battery.

Conclusion

The design, implementation and evaluation of the Smart Multi-Face Attendance System, a full-stack deep learning system, which addresses the three key failures of current academic attendance solutions, sequential single-student processing, lack of time-based presence monitoring and physical contact needs was described. Compared to paper-based registers, the proposed system completely avoids proxy fraud and minimizes the effort spent per-session registering attendance to none, and offers far more detailed time-based presence information that paper techniques cannot afford. Compared to the RFID systems, it removes cost of hardware installation (usually Rs.). 15,000 -50,000 per-classroom, eliminates the buddy-punch vulnerability and provides 4x more feature cover with no extra expense. Compared to fingerprint biometric systems, it eliminates all physical training, servicing of 30-40 students at a time compared to one at a time, and is resistant to degradation of fingerprint quality.

Table 9 Quantitative Performance Summary: Proposed System Vs. Existing Attendance Solutions

Performance Metric	Paper-Based	RFID	Fingerprint	Proposed System (Ours)
Recognition Accuracy	N/A	N/A	~95%	99.83%
Proxy Fraud Prevention	No	No	Yes	Yes
Contact-Free Operation	Yes	No	No	Yes
Multi-Face Simultaneous	No	No	No	Yes
Time-Based Presence	No	Partial	No	Yes
Auto Attendance	No	Partial	Partial	Yes

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2026.0304
e ISSN: 2584-2854
Volume: 04 Issue: 05
May 2026
Page No: 2095 - 2105

Calc.				
Real-Time Dashboard	No	Partial	Partial	Yes
CSV / PDF Export	N/A	Partial	Partial	Yes
Extra Hardware Cost	Zero	High	High	Zero
One-Click Deployment	N/A	N/A	N/A	Yes

The system reached 99.83 percent face recognition accuracy on LFW benchmark and 25-30 FPS recognition throughput on a single set of GPU hardware, latency of database query under 5ms and full attendance computation in 60 seconds at the end of a session in a single department. The one-click START.bat launcher saves and removes time to do one more double-click, which is not present at present in any academic attendance system. The proposed system is the only solution that fulfils all ten assessment parameters that have been compared in the comparative analysis. These findings affirm that the Smart Multi-Face Attendance System is a feasible, implementable, and scalable platform to the contemporary academic attendance management that is institutional deployment-ready with clear avenues of future improvement.

Acknowledgements

The authors are humbled to acknowledge the Department of Information Technology, BS Abdur Rahman Crescent Institute of Science and Technology as the provider of the Laboratory facilities and the computing facilities required in the course of this study. True appreciation is given to the InsightFace and ONNX runtime as open-source communities whose tools have become the technical backbone of this system, and to the student volunteers who helped with enrolment and testing as part of system testing.

References

[1]. Deng, J., Guo, J., Xue, N., & Zafeiriou, S. (2019). ArcFace: Additive Angular Margin Loss for Deep Face Recognition. Proc. IEEE/CVF CVPR. https://arxiv.org/abs/1801.07698

[2]. Schroff, F., Kalenichenko, D., & Philbin, J. (2015). FaceNet: A Unified Embedding for Face Recognition and Clustering. Proc. CVPR 2015. https://arxiv.org/abs/1503.03832

[3]. Zhang, K., Zhang, Z., Li, Z., & Qiao, Y. (2016). Joint Face Detection and Alignment Using Multitask Cascaded CNNs. IEEE Signal Processing Letters, 23(10), 1499–1503.

[4]. InsightFace Team. (2021). InsightFace: An Open Source Deep Face Analysis Library. https://github.com/deepinsight/insightface

[5]. ONNX Runtime Team. (2021). ONNX Runtime: Cross-Platform ML Inferencing Accelerator. https://onnxruntime.ai

[6]. Bradski, G. (2000). The OpenCV Library. Dr. Dobb's Journal of Software Tools.

[7]. Grinberg, M. (2018). Flask Web Development (2nd ed.). O'Reilly Media.

[8]. Taigman, Y., Yang, M., Ranzato, M., & Wolf, L. (2014). DeepFace: Closing the Gap to Human-Level Performance in Face Verification. Proc. CVPR 2014.

[9]. Wang, H. et al. (2018). CosFace: Large Margin Cosine Loss for Deep Face Recognition. Proc. CVPR 2018.

[10]. [10] Mukherjee, P., & Prasad, R. (2020). Automated Attendance Using Face Recognition: A Survey. IJCA, 175(18), 12–18.

[11]. Viola, P., & Jones, M. (2001). Rapid Object Detection Using a Boosted Cascade of Simple Features. Proc. CVPR 2001.

[12]. Turk, M., & Pentland, A. (1991). Eigenfaces for Recognition. Journal of Cognitive Neuroscience, 3(1), 71–86.

[13]. Thalluri, L.N., Babburu, K., Madam, A.K., et

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2026.0304
e ISSN: 2584-2854
Volume: 04 Issue: 05
May 2026
Page No: 2095 - 2105

al. (2024). Automated face recognition system for smart attendance application using convolutional neural networks. Int. Journal of Intelligent Robotics and Applications, 8, 162–178. https://doi.org/10.1007/s41315-023-00310-1

[14]. Surantha, N., & Sugijakko, B. (2024). Lightweight face recognition-based portable attendance system with liveness detection. Internet of Things, 25, 101089. https://doi.org/10.1016/j.iot.2024.101089

[15]. N., A., & Anusudha, K. (2024). Real-time face recognition system based on YOLO and InsightFace. Multimedia Tools and Applications, 83, 31893–31910. https://doi.org/10.1007/s11042-023-16831-7

[16]. Shukla, A.K., Shukla, A., & Singh, R. (2024). Automatic attendance system based on CNN–LSTM and face recognition. Int. Journal of Information Technology, 16, 1293–1301. https://doi.org/10.1007/s41870-023-01495-1