Smart AI-based CCTV attendance ledger using face recognition and liveness validation

Kushal VR[*1]
USN: 1EW23CS065
*[*1]Student, East West Institute of Technology, Off Magadi Main Road, 63, Near BEL Layout, Vishwaneedam Post, Anjana Nagar, Bengaluru, Karnataka, India*

Rahul P[2]
USN: 1EW23CS117
[2]Student, East West Institute of Technology, Off Magadi Main Road, 63, Near BEL Layout, Vishwaneedam Post, Anjana Nagar, Bengaluru, Karnataka, India

Pramod Gowda B.S[3]
USN: 1EW23CS109
[3]Student, East West Institute of Technology, Off Magadi Main Road, 63, Near BEL Layout, Vishwaneedam Post, Anjana Nagar, Bengaluru, Karnataka, India

Darshan S.R[4]
USN: 1EW23CS30
[4]Student, East West Institute of Technology, Off Magadi Main Road, 63, Near BEL Layout, Vishwaneedam Post, Anjana Nagar, Bengaluru, Karnataka, India

DOI: 10.5281/zenodo.23723588

Abstract:
Taking attendance is a small task that happens regularly, but it can take up valuable class time. Paper registers and manual roll calls also leave room for missed entries, typing mistakes, and proxy attendance. This paper describes a Smart AI-Based CCTV Attendance Ledger designed to record attendance from CCTV/IP-camera streams or USB webcams. The project repository documents a modular application using Next.js, React, TypeScript, FastAPI, Python, OpenCV, DeepFace, and PostgreSQL through Supabase, with SQLite available for local development. The proposed workflow combines face detection, identity matching, liveness or anti-spoofing checks, and persistent attendance records. For the system to be useful in practice, it also needs to handle poor-quality images, repeated detections, uncertain matches, database errors, access permissions, and the privacy risks that come with biometric data. This paper explains the problem, related work, system design, operational workflow, evaluation strategy, risks, and future improvements. This paper does not report accuracy or speed figures because no reproducible test results were available when it was prepared.

Keywords:
Facial recognition, CCTV, attendance automation, computer vision, liveness detection, anti-spoofing, audit ledger.

1. Introduction:
Schools and colleges use attendance records to track participation and support routine administration. With a traditional process, a teacher or staff member may need to call out names, circulate a register, or enter the results into a spreadsheet. These methods are easy to understand but become repetitive in large classes and may be vulnerable to mistakes, forgotten entries, and proxy attendance. A camera-based approach could reduce some of this repetitive work while keeping an electronic record for later review.
The project titled Smart AI-Based CCTV Attendance Ledger proposes to identify enrolled people from camera frames and create attendance events with timestamps. An attendance system needs to do more than

recognise a face. It must bring several steps together: detecting a face, assessing whether the image is usable, matching it to an enrolled identity, checking liveness where available, applying session rules, and recording the outcome. Each stage can fail independently, so the system should provide safe handling of uncertain or incomplete decisions.

1.1. Problem statement:

The core problem is how to reduce the manual effort of attendance recording without treating automated recognition as infallible. The system must operate under variable lighting, viewing angles, movement, partial occlusion, and different camera resolutions. It must also avoid logging the same person repeatedly from consecutive frames and preserve enough information to explain how a record was created or corrected.

1.2. Objectives and scope:

The objectives are to: (1) accept supported camera sources; (2) detect and align faces in video frames; (3) compare detected faces with enrolled identities; (4) use liveness or anti-spoofing checks to reduce simple presentation attacks; (5) validate the session and avoid duplicate attendance entries; (6) store events in a structured database; and (7) provide an authorised dashboard for monitoring and review. The intended scope is attendance support in a college or similar institution, not unrestricted surveillance or fully automated disciplinary decision-making.

1.3. Research contribution:

The project brings a computer-vision workflow together with a stored attendance ledger and a web interface that staff can use to review records. Its practical value depends on how reliably these components work together in the target environment. The paper therefore describes both the proposed design and an evaluation plan, while distinguishing repository-documented features from results that still require controlled testing.

2. Related work and technical background:

Face recognition generally involves detecting a face, extracting a numerical representation or embedding, and comparing that representation with stored templates. Deep learning models can learn features that remain useful across some variations in pose and illumination, but recognition remains sensitive to input quality and to differences between enrolment images and operational camera frames. In classroom or corridor settings, subjects may be far from the camera, look away from it, move quickly, or appear in groups. These conditions differ from a controlled frontal photograph and should be reflected in the system's evaluation.

Khan et al. described a CCTV-based attendance support system using deep face recognition in an educational setting [1]. Related work has also examined lightweight attendance systems that combine recognition with liveness detection, showing why identity matching and spoof detection should be considered distinct tasks [2]. The National Institute of Standards and Technology (NIST) has reported that face-recognition error rates can vary with image quality and demographic factors, so performance should not be assumed to be identical across all people or capture conditions [3].

2.1. Face detection and recognition:

Face detection locates the face region in an image; it does not establish the person's identity. Recognition then compares a detected face representation against enrolled templates. A threshold is used to decide whether a candidate match is sufficiently similar. A threshold that is too permissive may increase false acceptance, while a threshold that is too strict may increase false rejection. Thresholds should therefore be selected using representative validation data rather than chosen solely for convenience.

2.2. Liveness and presentation-attack detection:

A face-recognition system may be deceived by a printed image, a photograph displayed on a phone, or a replayed video. Liveness or presentation-attack detection attempts to identify evidence that the camera is observing a live subject rather than an artefact. No single check guarantees resistance to all attacks; performance depends on the method, camera, lighting, and attacker capability. ISO/IEC 30107-3:2023 provides testing and reporting guidance for biometric presentation-attack detection [5].

2.3. Attendance as a transaction:

Attendance is not merely a recognition label. It is a time-bound event associated with a person, a class or session, and a set of institutional rules. A system should decide what happens when a person is recognised twice, arrives outside a permitted window, appears in overlapping camera zones, or is recognised while the

database is unavailable. Treating attendance as a transaction—with validation, duplicate protection, and audit logging—helps prevent inconsistent records.

2.4. Research gap and design rationale:

A prototype may demonstrate face detection under ideal conditions but still lack safeguards needed for deployment. The design addressed here focuses on the complete workflow: image quality, recognition, liveness, session validation, database persistence, and human review. The aim is not to claim that these measures eliminate errors, but to define a practical architecture and a measurable plan for determining whether the system is suitable for its intended environment.

3. Proposed system architecture:

According to the project repository, the web interface uses Next.js 15, React 19, TypeScript, Tailwind CSS, Lucide React, and Recharts. The backend is documented as Python 3.13 with FastAPI, SQLAlchemy, Pydantic, and Uvicorn. OpenCV and DeepFace are listed for computer-vision tasks, alongside local facial-landmark and anti-spoofing components. PostgreSQL through Supabase is the primary database option, while SQLite is documented for local development [4]. Each feature should still be checked against the running application and its tests before it is described as fully implemented.

3.1. Frontend layer:

The frontend is the interaction layer for authorised users. It can present system status, camera or processing status, attendance summaries, and ledger records. A clear interface should distinguish confirmed records from uncertain matches, failed captures, and manual corrections. Dashboard visualisations can help staff inspect attendance by session or date, but charts must be derived from persisted records and should not conceal missing or incomplete data.

3.2. Backend and processing layer:

The FastAPI backend provides a boundary between the user interface, computer-vision processing, and data storage. Request validation helps ensure that incoming values have the expected structure, while SQLAlchemy can provide a consistent data-access layer. Video processing should be separated from ordinary dashboard requests so that long-running inference does not make the interface unresponsive. The implementation should also define how camera failures, timeouts, malformed frames, and model-loading errors are reported.

3.3. Data layer:

The database is expected to hold attendance events alongside the information needed to interpret them, such as enrolled people, class sessions, camera sources, and administrative actions. A robust schema should use stable identifiers rather than relying only on names, because names may be duplicated or changed. A uniqueness rule for person-and-session attendance can prevent repeated records, while transaction handling helps ensure that concurrent detections do not create duplicates. Access to biometric templates and attendance records should be limited to authorised roles.

3.4. High-level processing workflow:

Camera input → Frame sampling → Image-quality screening → Face detection/alignment → Liveness check → Identity matching → Session and duplicate validation → Database write → Dashboard and audit review

3.5. Core data entities:

A practical data model may include Person, EnrollmentTemplate, Session, CameraSource, AttendanceEvent, and AuditEvent. Person stores a unique institutional identifier and minimal profile information. EnrollmentTemplate stores protected biometric representations and model metadata. Session defines the attendance window and relevant class. AttendanceEvent records the person, session, timestamp, decision state, and source. AuditEvent captures authorised edits and system-level decisions. This is a conceptual model; the final schema should be checked against the repository's migrations before being described as implemented.

3.6. Deployment considerations:

A deployment may run the frontend and API on a local machine or approved server, with the database hosted locally or through Supabase. Camera access, network connectivity, and compute availability affect end-to-end performance. Credentials must be stored outside source code, production endpoints should use encrypted connections, and logs should avoid storing unnecessary face images or secrets. If cloud services are used, institutions should understand where biometric-related data is processed and stored.

4. Operational methodology:

4.1. Enrolment:

Before attendance can be recorded, each participant must be enrolled through an authorised process. Enrolment should confirm the person's identity using institutional procedures and capture images under conditions similar to the expected operating environment. Where multiple enrolment samples are supported, they should represent reasonable variation without collecting more data than necessary. Templates must be associated with the correct institutional identifier, protected from unauthorised access, and updated or deleted according to an established retention policy.

4.2. Camera capture and image quality:

The capture stage obtains frames from a supported USB webcam or CCTV/IP-camera source. Sampling every available frame may be unnecessarily expensive, so the system can process selected frames or use a configurable interval. A quality gate can defer recognition when the face is too small, blurred, occluded, or poorly illuminated. Camera placement should minimise extreme angles and backlighting, and the system should be tested at the actual distance between the camera and participants.

4.3. Recognition and decision thresholds:

After detection and alignment, the recognition model generates a representation or similarity score. The system compares the result against enrolled templates and considers a match only when it satisfies a validated threshold. It should not automatically assign the nearest identity if all candidates are weak matches. Low-confidence detections should remain unresolved or be routed for review. Threshold calibration must balance false acceptance and false rejection and should use a separate validation set rather than the same images used for enrolment.

4.4. Liveness and anti-spoofing:

Liveness screening is a second decision stage, not a replacement for identity recognition. The system should define what happens when liveness cannot be determined—for example, due to low resolution or a failed landmark calculation. A conservative design should avoid marking attendance as confirmed when required checks fail. Testing should include legitimate live presentations and authorised test artefacts such as printed photos and screen replays, under documented conditions. Attack tests should be performed ethically and should not involve collecting biometric data without consent.

4.5. Session rules and duplicate prevention:

A successful identity match should be associated with an active attendance session. Business rules may define an arrival window, late status, permitted camera zones, and whether a correction requires staff approval. Consecutive frames of the same person should not create multiple records. Duplicate prevention should be enforced at the database level as well as in application logic, since simultaneous requests can bypass a simple in-memory check. Failed database writes should be retried safely or surfaced as an error rather than silently lost.

4.6. Ledger review and corrections:

The ledger should display the time, session, status, and relevant source information needed to review an event. Corrections should preserve an audit trail containing the prior value, new value, actor, time, and reason where appropriate. A student or staff member should have a clear process for disputing an incorrect record. Automated records that are uncertain, incomplete, or manually corrected should be distinguishable from records confirmed by the normal pipeline.

4.7. Error handling and recovery:

Operational tests should cover camera disconnection, API timeouts, unavailable models, database outages, duplicate requests, and application restarts. The system should expose understandable status messages while avoiding disclosure of sensitive implementation details to ordinary users. Backups and recovery procedures should be tested, not merely configured. Monitoring should focus on service health and error rates while minimising retention of raw biometric data.

5. Evaluation plan and discussion:

5.1. Experimental setup:

Evaluation should use a consented, labelled dataset collected under the intended deployment conditions. Enrolment and test images must be separated to avoid measuring performance on images already used to create templates. The test plan should include different lighting levels, camera distances, viewing angles, partial occlusion, multiple people in a frame, and legitimate variations in appearance. Ground-truth attendance should be recorded independently so that automated events can be compared with a trusted reference.

5.2. Metrics:

To judge whether the system works well enough for its intended setting, the evaluation should report: recognition precision and recall; false acceptance rate (FAR), which measures incorrect identity acceptances; false rejection rate (FRR), which measures genuine users rejected by the system; presentation-attack detection performance; average and percentile processing latency; duplicate-event rate; successful database-write rate; and service availability. Report the number of test cases and the test conditions alongside each metric. A single overall accuracy figure can hide important failure modes, so results should be broken down by condition and interpreted carefully.

5.3. Expected benefits and limitations:

If it performs reliably, the system could reduce time spent on roll calls, create timestamped records, make reporting easier, and leave a trace when someone corrects an entry. These benefits remain conditional on the system's reliability and on staff having a workable exception-handling process. Low-resolution or off-angle footage, occlusion, network failures, and recognition bias can still cause errors. Liveness checks reduce certain presentation attacks but do not guarantee complete protection. Until the implementation has been tested using a repeatable protocol, numerical performance and real-world effectiveness should be treated as unknown.

6. Privacy, security, and responsible use:

Face images and derived templates are sensitive biometric information. The institution should provide clear notice about the purpose of processing, establish an appropriate legal basis, define retention and deletion schedules, and restrict access by role. Data should be protected in transit and at rest, backups should be secured, and administrative access should be logged. Raw camera frames should not be retained longer than necessary for the stated purpose. The system should avoid collecting unrelated information and should provide a correction or appeal process for disputed attendance.

Security controls should include authentication, authorisation, input validation, secret management, dependency updates, and review of API permissions. Database credentials and service keys must not be exposed in client-side code or committed to the repository. An institution should complete a privacy and security assessment before deployment and verify applicable Indian data-protection requirements and institutional policies. Where appropriate, a non-biometric attendance alternative should be available. Automated recognition should support administration, not serve as the sole basis for serious disciplinary decisions.

7. Conclusion and future work:

This paper has outlined a Smart AI-Based CCTV Attendance Ledger that links camera input, face recognition, liveness checks, session rules, and stored attendance records. The repository documents a modular technology stack with a Next.js frontend, FastAPI backend, computer-vision libraries, and PostgreSQL/Supabase storage [4]. The main engineering challenge is to make attendance easier without losing sight of recognition errors, spoofing risks, record traceability, or privacy. Future work should verify each documented feature against the source code, benchmark the running system, calibrate decision

thresholds, test presentation attacks, measure latency, evaluate performance across capture conditions, and audit security and data-retention practices. Only reproducible experiments should be used to support claims about accuracy or deployment readiness.

8. References:

(1) M. H. Raza Khan et al., "Implementing CCTV-Based Attendance Taking Support System Using Deep Face Recognition: A Case Study at FPT Polytechnic College," Symmetry, vol. 12, no. 2, art. 307, 2020, doi: 10.3390/sym12020307.

(2) "Lightweight face recognition-based portable attendance system with liveness detection," Internet of Things, vol. 25, art. 101089, Apr. 2024, doi: 10.1016/j.iot.2024.101089.

(3) P. J. Grother, M. L. Ngan, and K. K. Hanaoka, "Face Recognition Vendor Test (FRVT), Part 3: Demographic Effects," NIST Interagency Report 8280, 2019, doi: 10.6028/NIST.IR.8280.

(4) K. VR, "Smart AI-Based CCTV Attendance Ledger," project repository and README, GitHub. Available: https://github.com/Kushal-VR/College-Project (accessed Oct. 21, 2026).

(5) ISO/IEC 30107-3:2023, Information technology—Biometric presentation attack detection—Part 3: Testing and reporting, International Organization for Standardization, 2023.