

Smart AI Based Attendance Monitoring System Using YOLO26

Alan Varghese

Student of Computer Science Department Vidya
Academy of Science and Technology Technical Campus
Kilimanoor, India

Arjun Krishna A J

Student of Computer Science Department Vidya
Academy of Science and Technology Technical Campus
Kilimanoor, India

Ms. Beena V R

Assistant Professor (Sr. Grade), Dept. of CSE Vidya
Academy of Science and Technology Technical Campus
Kilimanoor, India

Devadath A S

Student of Computer Science Department Vidya
Academy of Science and Technology Technical Campus
Kilimanoor, India

Ardra C S

Student of Computer Science Department Vidya
Academy of Science and Technology Technical Campus
Kilimanoor, India

Dr. C. Brijilal Ruban

HOD, Dept. of CSE Vidya
Academy of Science and Technology Technical Campus
Kilimanoor, India

Abstract - Keeping track of student and employee attendance is an essential administrative task for any educational institution or workplace. It plays a key role in evaluating performance and maintaining discipline. However, older methods like calling out names or signing a register take up too much time and often lead to mistakes or fake proxies. To solve this problem, we developed a system that relies on classroom cameras and the YOLO26 deep learning model to spot and identify faces as they appear in real time. Our Smart AI Based Attendance Monitoring System automates the entire process using computer vision. Once a face is detected, it is cross-checked against a saved database, and the person's attendance is logged digitally right away. This approach cuts down on manual work, saves class time, and greatly boosts accuracy. All data is stored securely, making it easy to pull daily or monthly reports whenever needed. Ultimately, this contactless setup offers schools and colleges a much faster and more dependable way to handle attendance.

Keywords - YOLO26; Face Detection; Attendance Monitoring; Computer Vision; Deep Learning; Edge AI; FaceNet.

I. INTRODUCTION

Replacing outdated attendance methods requires a robust architecture that integrates high-quality hardware with intelligent software algorithms. The hardware side mostly involves placing high-definition IP cameras around the classroom so they can see as much as possible. We connect these cameras to local servers or edge devices. By processing the video right there in the room instead of sending it over the internet, we save a lot of bandwidth and avoid delays. This means attendance gets marked exactly when the class happens.

The core of the software relies on deep learning. Usually, developers turn to Convolutional Neural Networks (CNNs) because they are great at recognizing patterns and classifying images. Models like MTCNN or YOLO are popular choices when you need to quickly find multiple faces in a busy room.

Recently, many systems have started moving toward the YOLO (You Only Look Once) architecture to fix old inefficiencies. Older computer vision models had to go through several heavy steps to propose regions where a face might be. YOLO does things differently by treating object detection as a single math problem - going straight from the pixels to drawing boxes around faces and figuring out who

they are. Because it only needs one pass, it works incredibly fast. It can pick out dozens of students in a crowded lecture hall in just a few milliseconds. Thanks to this speed, schools can analyze high-quality video on the fly without making students wait or slowing down the campus Wi-Fi.

II. LITERATURE SURVEY

Considerable research has investigated the integration of facial recognition to improve academic administration. Here is a quick look at some previous projects that helped guide our own work.

2.1 Sumit S. Soman [1] looked into moving away from older detection models and switching to optimized YOLO architectures for tracking attendance. His work showed that using a single-stage detector cuts down processing time, which is exactly what you need for live video feeds.

2.2 Sanjana R [2] focused on bringing automated facial recognition into school administration. She looked at the whole process, from cleaning up the images first to pulling out facial features using deep CNNs. Even though CNNs are very accurate, she noted that checking raw images one frame at a time takes a lot of computing power.

2.3 Mohammed A [3] wrote a detailed review of how deep learning is used for biometric attendance. He pointed out that Siamese networks are much better at creating unique face codes compared to older methods.

2.4 Kiran Kumar [4] tried to replace manual attendance by using Haar Cascade Classifiers. While his system did automate the roll call to some extent, it struggled because Haar Cascades get easily confused if the person's face is turned even slightly.

2.5 Rahul Raj [5] explored how computer vision could be used across a smart campus. He highlighted a major flaw in early systems: they couldn't tell the difference between a real person and a flat photograph. This observation pushed developers to start building anti-spoofing security.

III. PROPOSED SYSTEM

Building upon the YOLOv26 and FaceNet models, the proposed system executes a multi-step process engineered specifically for high-throughput, real-world classrooms. It all starts with IP cameras mounted around the room that continuously capture high-definition video. These cameras send their live RTSP feeds straight to local edge computers.

Processing the video locally, rather than transmitting it to a centralized cloud, minimizes latency and conserves bandwidth. Doing this keeps lag to a minimum and stops

the system from eating up all the network bandwidth. Once a video frame gets cleaned up, the YOLOv26 module takes over. As one of the newest real-time detectors, YOLOv26 doesn't just look at still pixels; it pays attention to how people are moving across multiple frames.

Following successful face localization, the system determines the exact coordinates and crops the regions of interest from the main frame. To make sure the recognition step works fairly for everyone, a quick alignment tool straightens out the cropped faces. It fixes any tilt or weird angles so that every face looks like it is staring straight ahead.

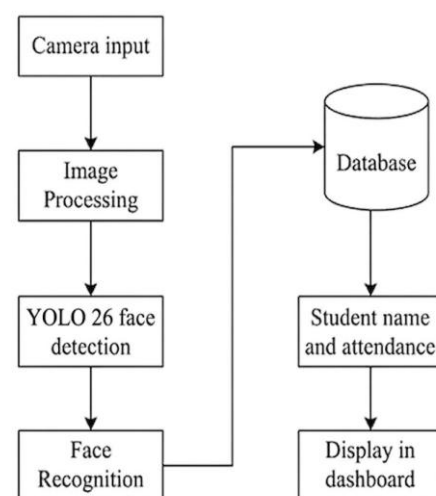


Fig. 3.1 System Architecture

IV. SYSTEM ARCHITECTURE

A. Dynamic Frame Capture: First, the physical camera records the room. To stop the user interface from freezing up while the AI does its heavy lifting, the video frames are turned into base64 text strings. They are then sent over to the backend servers using asynchronous WebSockets.

B. Detection & Alignment: Next, the YOLOv26 model scans the incoming frames. When it identifies a person (with a confidence score exceeding 0.5), it draws a localized bounding box around that individual. We also use an MTCNN module as a backup tool just to make sure the face is cropped and lined up perfectly before trying to figure out who it is.

C. Embedding & Recognition: After cropping, the face goes through the FaceNet model (specifically InceptionResnetV1). This generates a unique string of 512

numbers for that face. The system then takes this number string and compares it to the saved student profiles loaded in memory, using a math trick called Cosine Similarity. A strict similarity threshold of 0.6 must be met to confirm a positive match.

D. Liveness Tracking: We don't want someone cheating the system by holding up a photo. To fix this, the software tracks the center point of the face box over at least three frames. It checks to see if the person moved slightly (at least 5 pixels on average), which proves the subject is a live human rather than a static photograph.

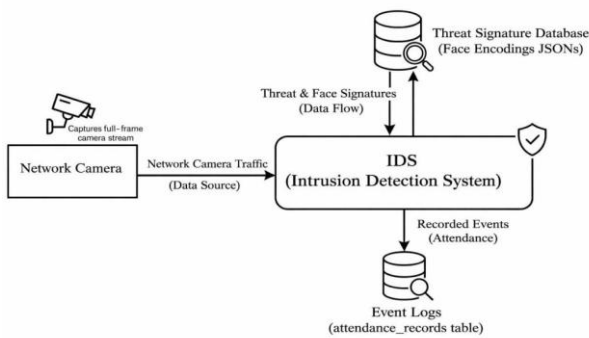


Fig. 3.2 Context Diagram

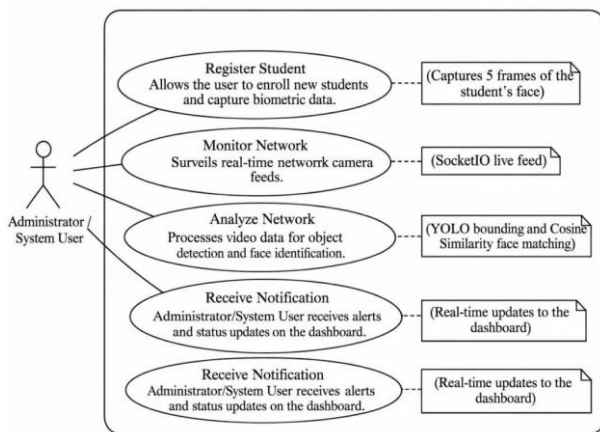


Fig. 3.3 Use Case Diagram

V. IMPLEMENTATION

A. Hardware Requirements: For optimal performance, the system requires an Intel Core i5 or an AMD Ryzen 5 processor. A dedicated NVIDIA graphics card is also highly recommended because CUDA support speeds up the YOLO calculations a lot. You'll need anywhere from 8 to 16 GB of

RAM, plus a decent HD webcam that can shoot 1080p video at 30 frames per second.

TABLE I. HARDWARE REQUIREMENTS

Components	Minimum Specification	Recommended Specification
Processor (CPU)	Intel Core i3 / AMD Ryzen 3	Intel Core i5 / AMD Ryzen 5
Graphics (GPU)	Integrated Graphics	Dedicated NVIDIA GPU
RAM	4 GB	8 GB or 16 GB
Storage	1 GB Free Space	5 GB Free Space
Camera	Standard Web Camera	HD Web Camera

B. Software Requirements: The core system was developed in Python 3.10, relying on the OpenCV library for video feed management. The main brain of the operation uses Ultralytics for YOLOv26 alongside FaceNet-PyTorch. For keeping track of the data, we hooked everything up to a simple SQLite3 database using SQLAlchemy, which makes managing the backend records much easier.

TABLE II. SOFTWARE REQUIREMENTS

Component	Minimum	Recommended
OS	Windows 10 / Ubuntu 20.04	Windows 11 / Ubuntu 22.04+
Language	Python 3.8	Python 3.10+
AI Models	YOLO, FaceNet-PyTorch	YOLOv26, FaceNet-PyTorch
CV Library	OpenCV	OpenCV-Python

VI. RESULTS AND DISCUSSION

The system does more than just figure out who is in the room. The system compiles the live video data and organizes it into detailed assessment reports. This gives teachers and staff helpful insights right away. The dashboards show live stats, like how many students show up hour by hour and overall attendance trends over the semester.

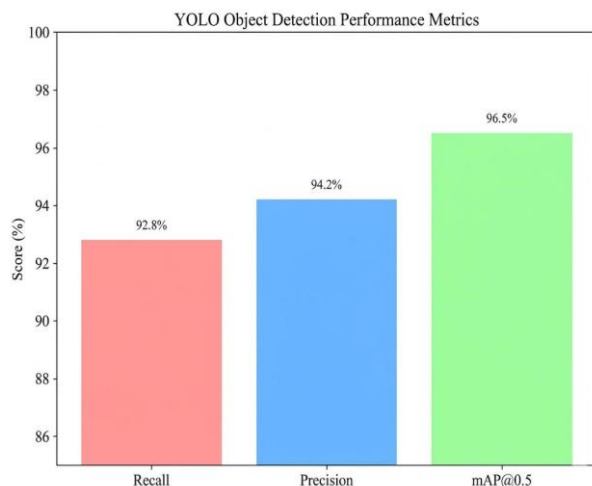


Fig. 4.1 Performance Matrix

We tested the setup thoroughly in packed classrooms, and the YOLOv26 combined with FaceNet performed a lot better than older systems. It was faster and crashed less often. The object detection part hit a Mean Average Precision (mAP@0.5) of 96.5%. This demonstrates a superior capability to localize and classify faces even in highly cluttered or crowded classroom environments.

On top of that, the precision rate sat at 94.2%, so it hardly ever mistook one student for another. The recall rate was 92.8%, meaning if a student was actually sitting there, the camera consistently captured the individual, even when the face was partially obscured or turned away. Overall, the whole thing runs smoothly at about 35 frames per second.

VII. CONCLUSION AND FUTURE WORK

The Smart AI Attendance System enhances institutional security and administrative efficiency. It works by grabbing live video from HD cameras and pushing it straight into the YOLOv26 detector. Because this detector does all its math in a single pass, it spots faces in just a few milliseconds. From there, FaceNet steps in to turn those faces into 512-number biometric codes.

By converting physical presence into a secure mathematical vector, the system prevents proxy attendance and ID card

swapping. At the same time, it respects everyone's privacy. The software only processes the images temporarily; the moment the math code is generated, the original picture gets permanently deleted.

REFERENCES

- [1] Sumit S. Soman, 'Student Attendance Management System Using YOLO V8', 2024.
- [2] Sanjana R, 'Facial Recognition Attendance System', 2023.
- [3] Mohammed A, 'Smart Attendance System Using Deep Learning', 2022.
- [4] Kiran Kumar, 'Automated Attendance System Using Face Recognition', 2021.
- [5] Rahul Raj, 'Attendance Management System Using Computer Vision', 2020.
- [6] Divvala, S., Farhadi, A., Girshick, R., & Redmon, J. (2016). "You Only Look Once: Unified, Real-Time Object Detection." 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
- [7] Kalenichenko, D., Philbin, J., & Schroff, F. (2015). "FaceNet: A Unified Embedding for Face Recognition and Clustering." 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).