iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** XI **Month of publication:** November 2025

DOI: https://doi.org/10.22214/ijraset.2025.75903

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue XI Nov 2025- Available at www.ijraset.com

AI-Powered Real-Time Face Recognition Attendance System for Organizations

Mrs. Radhika S N[1], Siddarth K G[2], Sudharshan Swamy M[2], Taruna B V[2], Tejas H S[2]
[1]*Assistant Professor,* [2]*UG Students, Department of Computer Science and Engineering, JNNCE, Visvesvaraya Technological University, Karnataka, India*

Abstract: ***This project focuses on developing a Face Recognition Attendance System using real-time image capture combined with machine learning and computer vision methods. The system streamlines the attendance marking process by using a CCTV camera to capture and recognize faces., eliminating the need for manual entry or biometric hardware. It leverages deep learning- based face recognition models for precise identification individuals and log their attendance in a connected database. The solution is designed to handle real-time inputs, manage user registration with live image capture, and ensure that each individual is marked only once per session. By automating the attendance process, this project improves operational efficiency, reduces time consumption, and minimizes human error. The system provides a scalable, accurate, and user-friendly alternative for institutions and organizations seeking a modern attendance solution.***

Index Terms: Artificial Intelligence, Automation, Biometric Authentication, CCTV-Based Recognition, Computer Vision, Contactless Attendance, Database Integration, Deep Learning, Employee Monitoring, Face Recognition, FaceNet, Facial Embedding, Flask, Intelligent Systems, MTCNN (Multi-Task Cascaded Convolutional Networks), MySQL, Organizational Efficiency, Python, Real-Time Attendance, Smart Attendance System

I. INTRODUCTION

Traditional attendance systems, which solely rely on the usage of manual signatures, RFID cards, or biometric fingerprint scanners, involve manual interaction; this, however, allows for contactless, automated logging through the discovery of people's faces in real time. Typically, these application utilize live video feeds, like the ones used by a webcam or CCTV camera, to record faces, to capture faces, compare them to a preregistered database using face encoding algorithms, and mark attendance upon assurance of a match. This desirably speeds up the processes, minimizes the possibility of proxy attendance, and guarantees hygiene and non-invasiveness in the process. Traditional mechanisms for attendance tend to be time-consuming, prone to errors, and susceptible to manipulation. In contrast, face recognition systems present an option that is flexible and can easily be scaled up for schools, businesses, and secure locations. By eliminating specialized hardware or manual intervention, these AI-powered face recognition systems foster efficiency and accuracy in operational tasks related to tracking attendance in diverse scenarios.

II. IMPORTANCE

In today's fast-paced and technology-driven environments, Automation of attendance systems is critical in enhancing efficiency, reducing errors and minimizing administrative workload. The face recognition-based attendance solutions offer a modern, intelligent alternative to conventional methods such as may appear as manual registers, punch cards, or biometric scanners. These Traditional systems usually involve time and are susceptible to proxy attendance, and limited in scalability. Automating Attendance using facial recognition ensures that organizations maintain Accurate, real-time tracking without physical contact or Specialized intervention saves not only duration but also resources. but also enhances security and transparency. The platform is highly scalable and adaptable to various environments. such as schools, colleges, offices, and factories.

III. SCOPE

The main goal is to develop an attendance system using real time face recognition technology, integrating machine learning and real-time image processing techniques. The primary objective is to enable educational institutions or organizations to capture attendance without manual intervention or physical hardware such as fingerprint scanners or ID card systems.

The system will:

1) Capture live images using a webcam or CCTV camera during specified time slots.
2) Detect and identify faces of individuals in real-time using deep learning models.

3) Record attendance data by matching recognized faces with the existing database.
4) Store and manage attendance logs with accurate timestamps in a structured and secure format.
5) Ensure each person is marked present only once per session to prevent duplicate entries.
6) Enable offline functionality by storing attendance locally and syncing with the server when a connection is re-established.
7) Include automatic face registration for new users, enabling administrators to add new students or employees directly through the interface by capturing and storing their facial data.
8) Implement a user-friendly dashboard for administrators to view, filter, and export attendance reports for individuals, departments, or specific dates.
9) Send real-time notifications (via email or app) to students or employees upon successful attendance marking, enhancing transparency.
10) Use advanced face detection algorithms such as FaceNet, DeepFace to increase accuracy even in the low lighting conditions or facial expressions.

IV. LITERATURE SURVEY

[1] Ngo Tung Son et al. developed a classroom attendance tracking framework using existing CCTV infrastructure and deep facial recognition models. The system utilizes pre-trained networks like ArcFace, SphereFace, and FaceNet with transfer learning to handle complex real-world scenarios such as lighting changes and camera angles. By deploying the system across five real classrooms at FPT Polytechnic College, the researchers demonstrated that accuracy remained consistently high across different conditions. Their modular design also enables group-wise classifiers for better scalability.

[2] Hao Yang and Xiaofeng Han designed a Face Recognition Attendance System Based on Real-Time Video Processing, aiming to improve attendance administrations in schools and secure locations through automated, efficient, and consistent ways. The setup integrates real-time video streams with advanced face sensing and identification modules. The authors compared traditional fingerprint and manual roll-call methods with their system, demonstrating a reduction in truancy rates and improved attendance accuracy (up to 82%) despite challenges like video blur and lighting. Their system architecture includes modules for login, recognition, check-in, and background management, built using Python, Java, C++, OpenCV, and MySQL. Experimental deployment across two universities showed improved stability, efficiency, and user experience, highlighting the feasibility of integrating real-time face recognition in attendance systems.

[3] Shreyak Sawhney et al. presented a facial recognition-based attendance system employing Eigenfaces, Principal Component Analysis (PCA), and Convolutional Neural Networks (CNNs). The method includes image pre-processing steps such as grayscale conversion and histogram equalization to enhance recognition accuracy. The camera captures classroom images, and detected faces are compared to a central database for attendance marking. This system addresses the issue of proxy attendance and reduces the administrative load on faculty.

[4] Dr. V. Suresh et al. introduced a Python and OpenCV-based attendance model that records student presence via facial identification. It supports a full database management interface through a GUI and allows new students to enroll easily. The model automatically stores attendance logs and sends daily reports to faculty via email, with optional access for parents. This system resolves issues of proxy attendance, improves data accessibility, and supports fast attendance marking with minimal effort.

[5] Soumitra Chowdhury, Sudipta Nath, Ashim Dey, and Annesha Das developed an Automatic Class Attendance System using CNN-based Face Recognition, targeting live student attendance in educational institutions. The system captures live video streams, detects multiple faces, and recognizes them against a trained dataset using a CNN model. Key stages include data entry, dataset training with triplet loss, real-time face recognition, and attendance logging in spreadsheets. The user interface supports adding, training, and monitoring attendance efficiently. Experiments showed that increasing the number of training photos per student improved accuracy, reaching up to 92%. The system eliminates manual errors, works well in real-time classroom settings, and provides automated, accurate attendance records while saving instructors' time.

[6] Dr. M.V.B. Chandrasekhar, Ch. Vamsi, B. Yogeswara Rao, R. Meghana, Y. Premkumar proposed an intelligent facial identification attendance system by realtime camera access. The model integrates YOLO for face recognition and CNNs/Vision Transformers for recognition, supporting live streaming via RTSP. The system records attendance autonomously, sends real-time updates, and stores data in a secure SQLite database through a Flask web app. It is designed to adapt to lighting variations and facial changes, ensuring reliability in schools and secure locations

[7] The authors of Continuous Real-time Automated Attendance System using Robust C2D-CNN proposed a novel classroom attendance tracker that tracks not only student presence but also duration of attendance. The system uses MTCNN for reliable face

identification and a novel C2D-CNN architecture for face identification. A 3D morphable model is employed to regenerate facial images under various light conditions and pose conditions, while ResNet-50 regresses feature vectors. The model handles occlusions and misalignments effectively using hybrid losses and a skin-color-based attention mask. The system is highly reliable for real-time classroom scenarios with inconspicuous CCTV setups

[8] Aashish Rai, Rashmi Karnani, Vishal Chudasama, and Kishor Upla presented an End-to-End Real-Time Face Identification and Attendance System using CNNs. The proposed system utilizes MTCNN for face detection and FaceNet for embedding generation, achieving 96.02% accuracy. The end-toend software processes video footage in a single shot, saving attendance records in spreadsheets. It addresses challenges like occlusion, alignment, and varying lighting while maintaining a user-friendly interface.

V. DESIGN AND IMPLEMENTATION

This chapter mainly includes the system architecture, flow of the project, methodology being used in the project. It also explores the technologies and concepts crucial for face recognition and data maintenance.

A. System Architecture

The figure 5.1 shows the outline of the system architecture of the project.

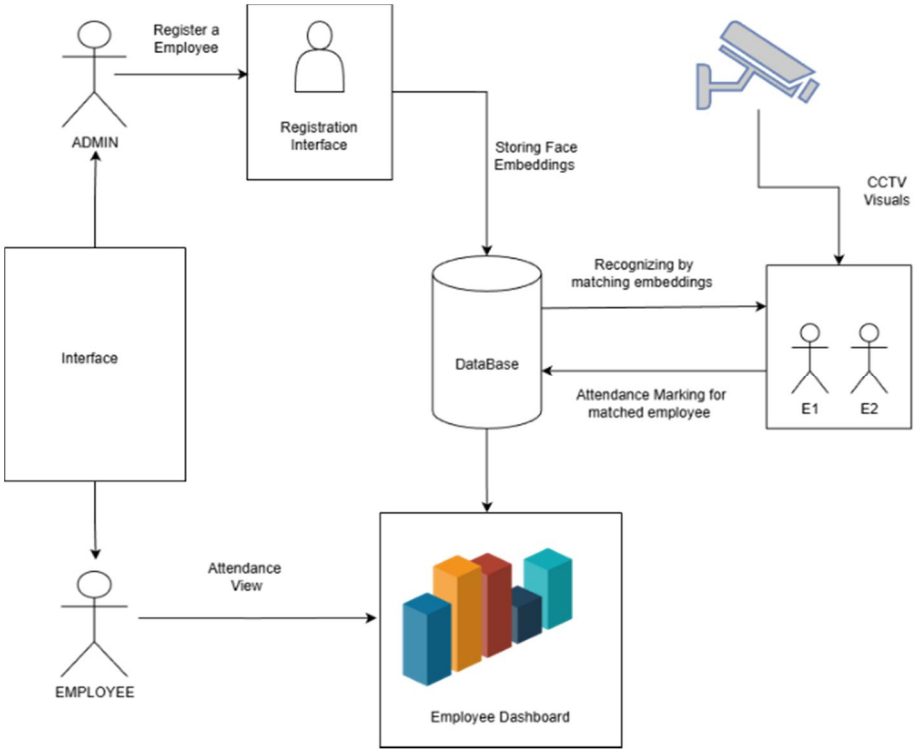


Figure 5.1 System architecture of the project

1) Registering the Employee

- The admin (Head of the organisation) by logging in can register a new employee.
- The new Employees Photos are captured and required details are taken.
- This step ensures that the details entered are correct and also take cares of duplicate entry.

2) Storing Face Embeddings in the Database

- From the photos captured (50 photos) during registration, embedding of each image is created and the average of all 50 images' embedding is created.
- This embedding is stored in the database table along with the other details.

3) Face Recognition from CCTV Live Stream

- Now From CCTV visuals a Face is recognised in real Time as the employee already registered enters the office.
- Recognition is based on the embeddings matching.

4) Attendance Marking in database

- Once the Employee is recognised this attendance is going to be noted for the recognised employee.
- Date and time shall be preserved in a database table.
- 2 times attendance is set to be taken in a day – Once in morning and once in the afternoon.

5) View Attendance Report

- An employee by logging in through their credentials can view their attendance report from any date to any other date.
- They can also download excel file of their attendance report.

B. Flow Chart

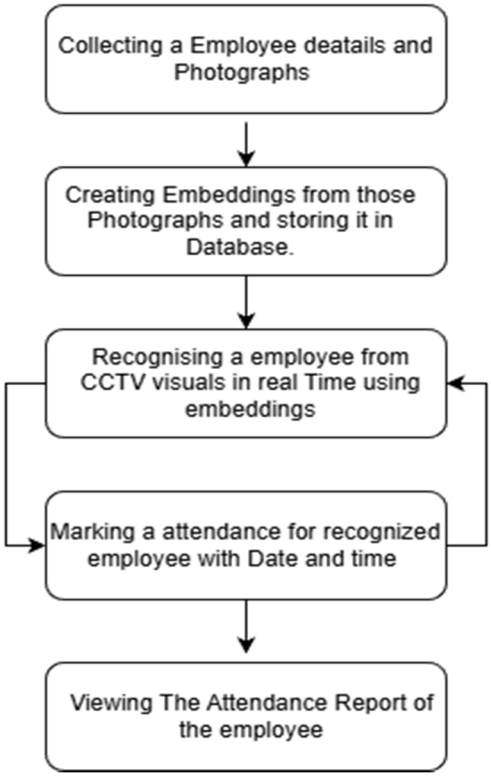


Figure 5.2 Flowchart of the project

Figure 5.2 shows the overview of the flow of project. The system starts by gathering an employee's basic details along with several photos during registration. The collected photos are then processed to create unique facial embedding, which is stored in the database for later use. From the CCTV live stream footage, the system identifies employees by comparing the detected face embedding with those stored earlier in real time.

If it matches to any of the employee's embedding, the system records that employee's attendance automatically with the exact date and time. Finally, employees and administrators can view attendance performance through logging in into the dashboard, making it easy to track and manage daily attendance activities.

C. Methodology

The methodology details the workflow adopted for implementing a real-time face-detection attendance system. The proposed framework encompasses six structured phases: data acquisition, preprocessing, feature extraction, identity verification, attendance recording, and user alert generation.

1) System Initialization and Database Configuration

The system begins by initializing the required facial recognition models and preparing the database. A structured relational database is used to maintain two core entities:

- Employee Records: Stores user information, secure password hashes, facial embeddings, and contact numbers.
- Attendance Records: Maintains daily attendance with four-time fields representing morning and afternoon entry/exit.

2) User Registration and Face Data Acquisition

During registration, employees provide their identification details, which are validated through rule-based checks for format, completeness, and consistency. A webcam then captures multiple facial images of the user under natural conditions. For each captured frame:

- The face is detected using a Haar Cascade classifier.
- The detected face region is cropped, normalized, and resized.
- A FaceNet deep learning model generates a numerical embedding representing the face.

A set of 50 embeddings is collected and averaged, producing a stable representative embedding for the user. This averaged vector is stored in the database as the user's unique facial signature. To prevent duplicate registrations, the system compares the new embedding against all existing embeddings. If similarity exceeds a predefined threshold, the registration is flagged as a duplicate.

3) Facial Embedding Storage

The final embedding is translated into a compact binary format and securely recorded in the database. This format allows faster fetching and comparison during real-time recognition while minimizing storage overhead.

4) Real-Time Face Recognition Pipeline

The real-time recognition module continuously processes video frames from a live camera feed. For each frame:

- Faces are detected using classical computer-vision techniques.
- Each detected face is cropped and pre-processed to match the input requirements of the embedding model.
- A facial embedding is generated using FaceNet.
- The embedding is compared with stored embeddings using cosine similarity.
- The identity with the highest similarity exceeding the threshold is classified as the recognized employee.
  To avoid repeated triggers, a cooldown period is enforced, ensuring recognition events for the same individual are spaced appropriately.

5) Attendance Marking Logic

Attendance is recorded according to defined temporal rules:

- Time-Window Restriction: Attendance can only be marked within a predefined working interval (e.g., 08:00–18:30). Any recognition outside this window is ignored to maintain data integrity.
- Two-Session Attendance Model
  The system divides the workday into morning and afternoon sessions. Based on the current time:
- Session 1: Morning (Entry: IN1, Exit: OUT1)
- Session 2: Afternoon (Entry: IN2, Exit: OUT2)

For each recognized employee, the system determines which field to update based on existing entries and the period of day. Time entries are kept with precise timestamps.

6) Automated Notification System

- Upon successful attendance marking, the system sends a real-time notification to the employee through a messaging API. The message includes: The employee's name and the timestamp and date.
- Notifications provide transparency and immediate confirmation to the employee.

7) Attendance Retrieval and User Dashboard

Authenticated employees can access a dashboard to review their attendance. The system:

- Retrieves attendance over a selected date range
- Calculates total working period for each day
- Presents the record in a structured tabular format

- Allows downloading the records as an Excel file

This ensures clear visibility of attendance history and supports organizational reporting.

VI. TOOLS AND TECHNOLOGIES USED

The tools and technologies used in the project, emphasizing Flask for backend development ,VS code for environment, package management and easier efficient installation of libraries and MySql a DBMS for data storage. Their integration ensures a secure, scalable, and efficient development framework. These tools form the foundation for the seamless functioning of the smart attendance system, providing a robust platform for development and deployment.

A. Python and Flask

Flask is a lightweight Python web framework designed to make building web applications straightforward. It provides only the essential components, keeping its core minimal and easy to extend. Because it follows a “micro-framework” approach, it does not ship with built-in tools like an Object Relational Mapper (ORM), leaving developers free to choose their own database solutions and additional libraries as needed.

Despite its simplicity, Flask offers useful features such as URL routing and a powerful templating system. It is built on top of the Werkzeug WSGI toolkit and uses the Jinja2 template engine—both created by members of the Pocoo community led by Armin Ronacher, who is also the original author of Flask. Figure 6.1 illustrates the relationship between Python and the Flask framework. The term “micro” reflects that Flask provides flexibility, allowing developers to decide how to structure their applications and which extensions or components to integrate.

Figure 6.1 Python and Flask

It is a micro framework, but that doesn’t mean your whole app should be inside one single Python file. You can and should use many files for larger programs, to handle complexity.

Flask is one among the top-rated web frameworks, meaning it’s up-to-date and modern. You can easily enlarge its functionality. You can scale it up for complex applications.

B. MySQL Workbench

MySQL is a widely used open-sourced RDBMS that forms the backbone of the records storage layer in this project. It is responsible for securely managing employee records, serialized facial embeddings, and time-stamped attendance logs in a structured tabular format. MySQL ensures data consistency, fast query execution, and reliable transaction handling, which are essential for a real-time face-recognition attendance system.

Figure 6.2 MySQL

MySQL Workbench is user interface used for designing, visualizing, and managing the database. It provides an intuitive environment for creating tables, inspecting stored data, executing SQL queries, and monitoring the database structure. Together, MySQL and MySQL Workbench streamline database development and administration, ensuring efficient data organization and smooth integration with the Python-based backend.

C. FaceNet Model

FaceNet is a deep-learning–based face recognition model designed to convert a face image into a compact numerical representation known as an embedding. Instead of classifying faces directly, FaceNet learns to position visuals of the same person close together in a high-feature space while pushing photos of different people far apart. This idea makes FaceNet extremely efficient and flexible: once embeddings are generated, tasks like recognition, verification, or clustering can be performed using simple distance calculations.

FaceNet is widely adopted in modern biometric systems due to its accuracy, robustness, and ability to work reliably even under real-world variations such as lighting changes, camera angles, or facial expressions.

1) Properties of FaceNet Model

- Embedding-Based Face Representation: Each face is transformed into a fixed-length vector (commonly 128 dimensions). This vector mathematically represents the characteristics that are unique to the face.
- Discriminative Feature Learning: FaceNet uses a specialized loss function known as Triplet Loss, which trains the network to pull similar faces together and push dissimilar ones apart.
- Invariance to External Factors: The model maintains high accuracy despite variations in pose, illumination, background, and facial expression.
- High Scalability: Since recognition depends only on vector comparison, the apllication can scale to thousands or millions of clients without retraining the model.
- Compact and Efficient: The model produces small but highly informative embeddings, making storage, retrieval, and comparison extremely efficient.

2) Working of FaceNet

FaceNet workflow consists of a sequence of steps involving face detection, alignment, embedding extraction, and similarity comparison.

- Face Detection: The starting point is identifying the position of a face within an image. Classical methods such as Haar Cascades or more advanced detectors like MTCNN may be designed to extract the face region.
- Preprocessing: In this step the detected face is standardized for uniformity by doing the following operations:
    - Converting it to RGB.
    - Scaling it to a fixed resolution. For example 160 x 160 pixels.
    - Normalizing it to match FaceNet's input expectations.

This improves the consistency of embeddings generated by the visuals, and also helps in improving the model's accuracy.

- Embedding Generation: The face after preprocessing in the second step is now passed through FaceNet's deep convolutional network.
  Instead of producing a class label, the network outputs a 128-dimensional embedding vector. This vector numerically encodes the person's identity, which we have stored in the record system

- Triplet Loss Optimization: FaceNet uses three types visuals to train the model as mentioned below:
  - Anchor (A): reference image, a image of a face of a person.
  - Positive (P): Same person's (Anchor) image.
  - Negative (N): Image of a face of different person

The Model tries to satisfy the below equation:
Distance (A, P) + margin < Distance (A, N)
By this the model makes sure that embeddings from the same person cluster tightly.

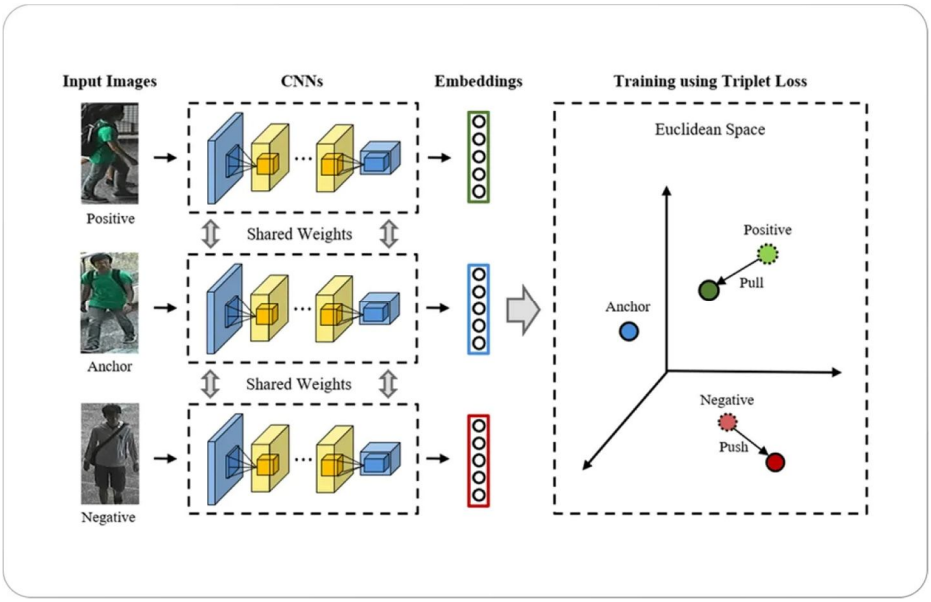


Figure 6.3.2 Triplet Loss Optimization

- Similarity Measurement

During recognition, embeddings are compared using distance metrics such as:

- Euclidean distance
- Cosine similarity

If the distance between two embeddings is below a threshold, then they are considered to be the embeddings of the same individual.

VII. RESULTS

A. Role Select

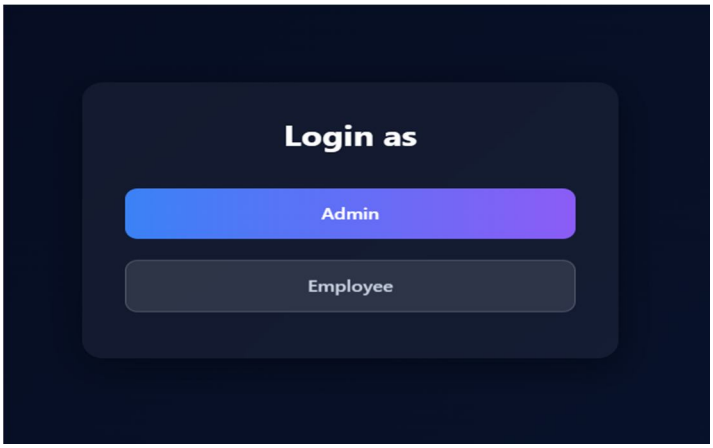


Figure 7.1 Role Select Interface

Here, the user can select their role as Admin or Employee and login.

B. Admin Login Credentials

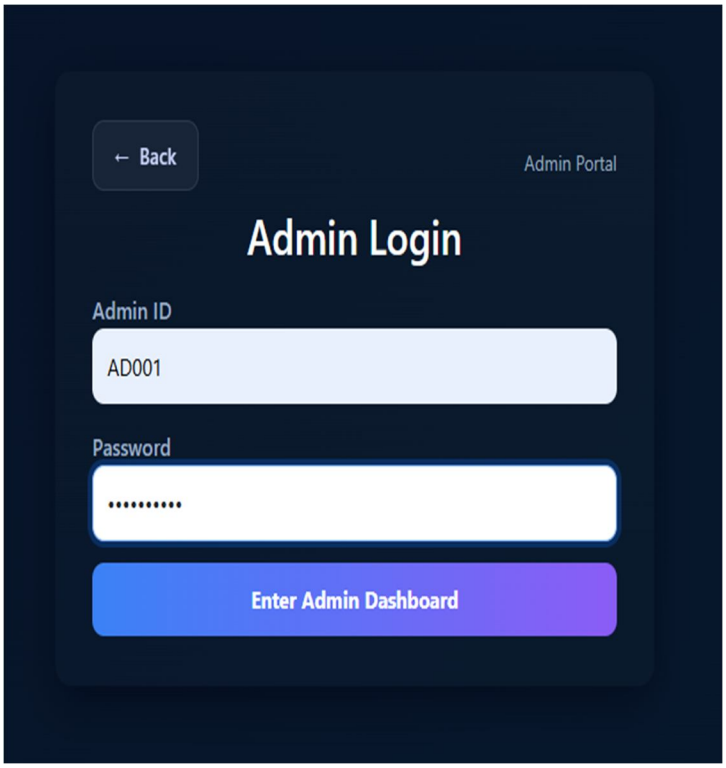


Figure 7.2 Admin Login Credentials

Figure 7.2 give the Admin Login page, where the admin can enter their credentials (Admin ID and password) to login.

C. Admin Dashboard

Figure 7.3 Admin Dashboard

Figure 7.3 gives the interface of the admin dashboard. Here Admin can view any Employee's attendance by clicking on the "view" button. The admin can search the Employee in the search box. Admin can also load the attendance details of a particular employee in any date range and download the excel report.

D. Employee Registration

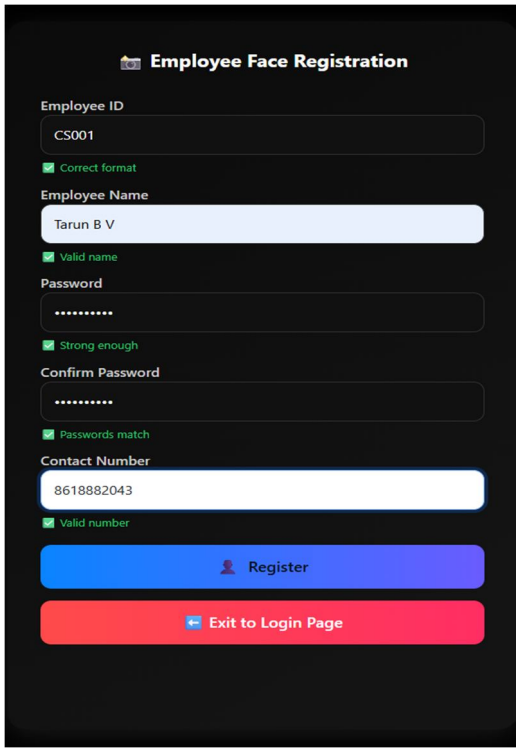


Figure 7.4 Employee Registration

On clicking on the "Register Employee" button in the admin dashboard, Employee Face Registration is opened, where the admin can register a new employee by entering the information of the new employee. After successfully entering the details, then the platform will start to capture images of new employee, create embedding and stores the data in the database.

E. Registration Successful

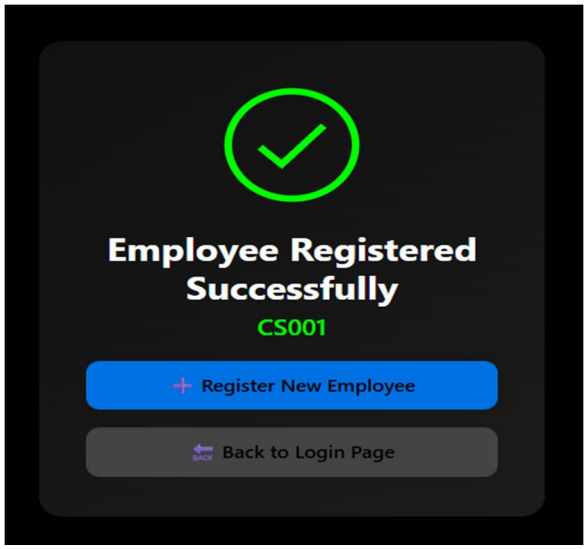


Figure 7.5 Registration Successful

Figure 7.5 shows the successful registration of the employee, where employee can register a new employee or he/she can go back to the login page.

F. Storing in the Database

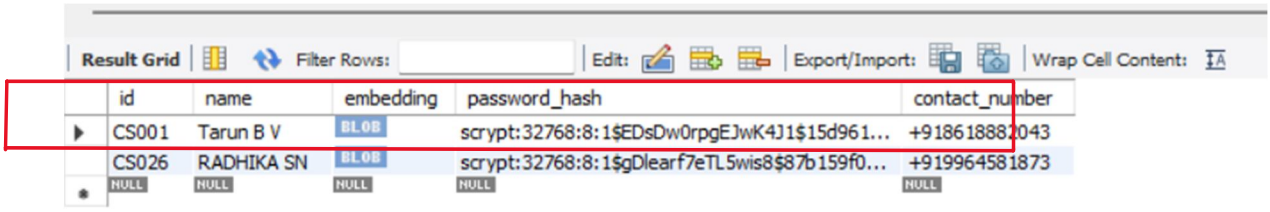


Figure 7.6 Storing in the Database

Figure 7.6 shows how the data which is taken during the employee registration is preserved in the database.

G. Employee Login Credentials

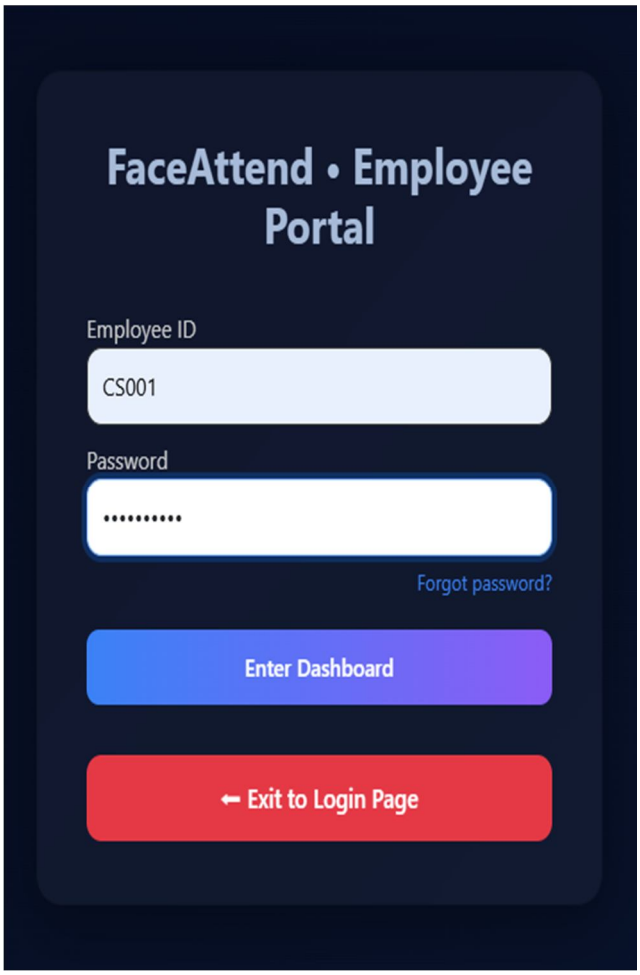


Figure 7.7 Employee login credentials

After successful registration, the employee can now login to their dashboard. Figure 7.7 give the employee login page, where the employee can enter their credentials (Employee ID and password) to login.

H. Employee Dashboard

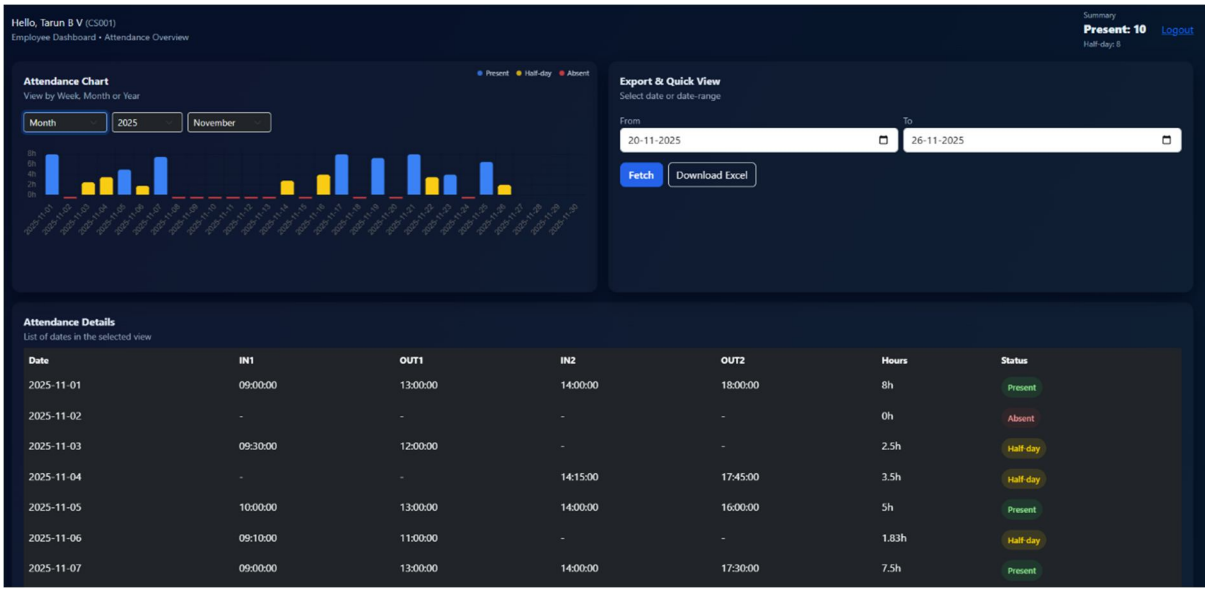


Figure 7.8 Employee Dashboard

Figure 7.8 gives the interface of Employee dashboard, where the employee can view the attendance report from any date range and can download the excel report also.

I. Excel Report

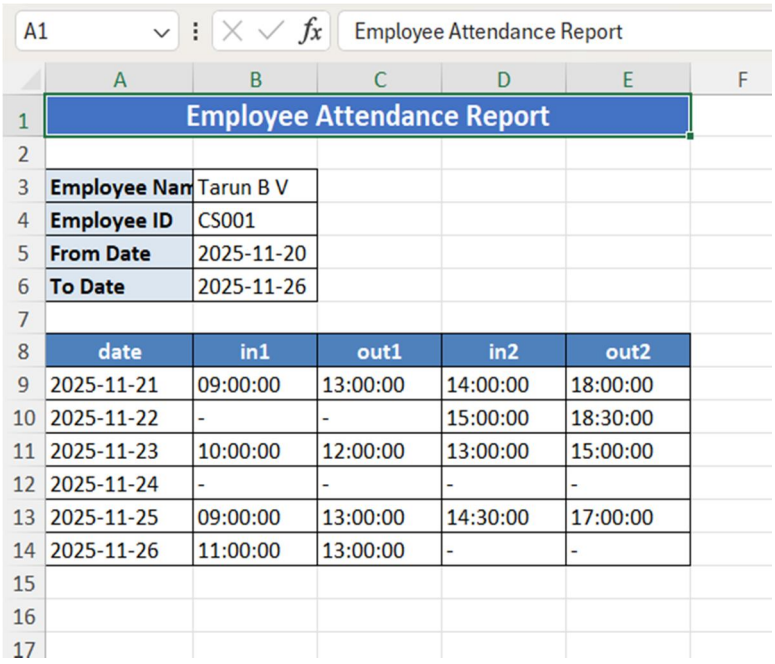


Figure 7.9 Employee Attendance Report

Figure 7.9 shows how the downloaded excel report looks like. It will show the date range as the employee given in the dashboard while downloading the report.

J. Face Recognition

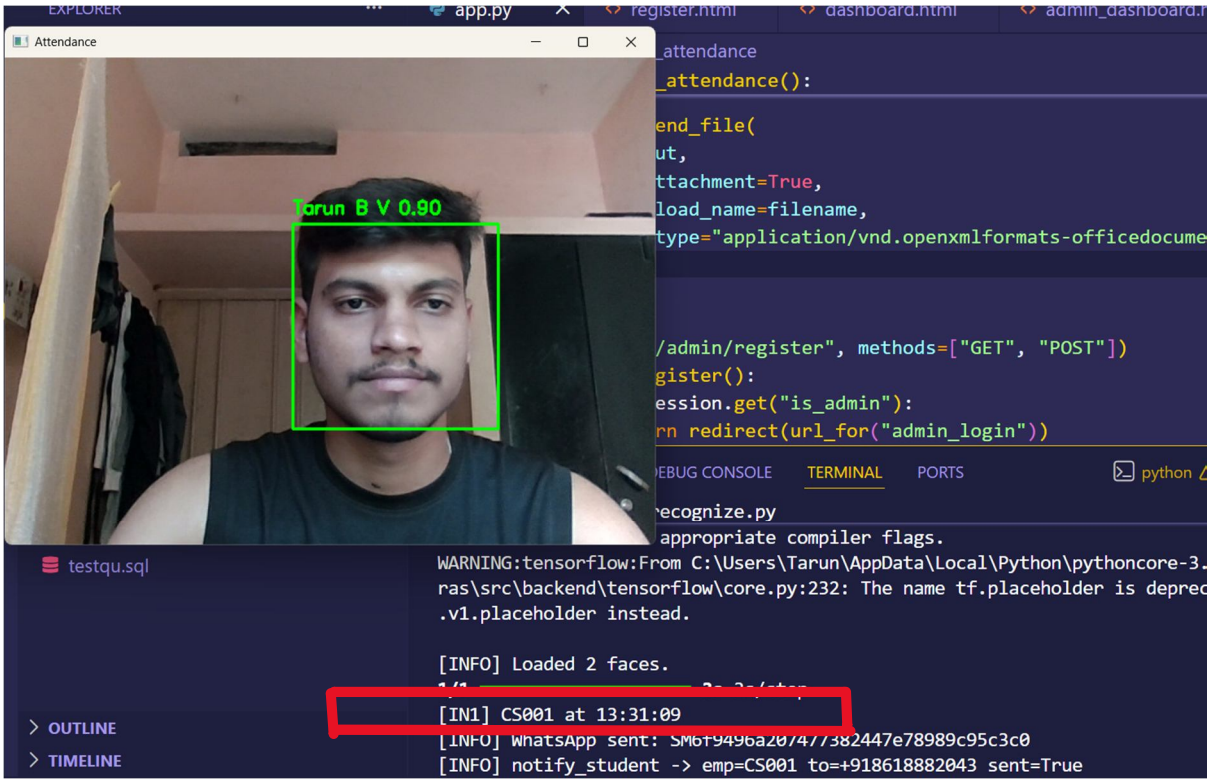


Figure 7.10 Face Recognition

K. Marking Attendance

id	emp_id	date	in1	out1	in2	out2
234	CS001	2025-10-13	14:00:00	17:00:00	NULL	NULL
235	CS001	2025-10-15	09:00:00	12:30:00	NULL	NULL
236	CS001	2025-10-16	NULL	NULL	14:00:00	17:30:00
237	CS001	2025-10-19	10:00:00	12:00:00	NULL	NULL
238	CS001	2025-10-23	NULL	NULL	15:00:00	17:00:00
239	CS001	2025-10-27	09:00:00	12:00:00	NULL	NULL
240	CS001	2025-10-21	NULL	NULL	NULL	NULL
241	CS001	2025-10-25	NULL	NULL	NULL	NULL
242	CS001	2025-10-29	NULL	NULL	NULL	NULL
243	CS001	2025-10-31				
					NULL	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Figure 7.11 Marking Attendance

VIII. CONCLUSION

Facial detection tech keeps shifting the way we track attendance in schools and offices. It moves fast in that area. Old manual methods, such as calling out names or passing around sign-in sheets, eat up a lot of time. They open the door to mistakes too. Attendance is sometimes marked on behalf of individuals who are absent. Researchers stepped up to tackle those issues. They turned to automated options powered by AI. These setups run quicker and hit higher accuracy rates. They resist tampering in ways the old systems never could.

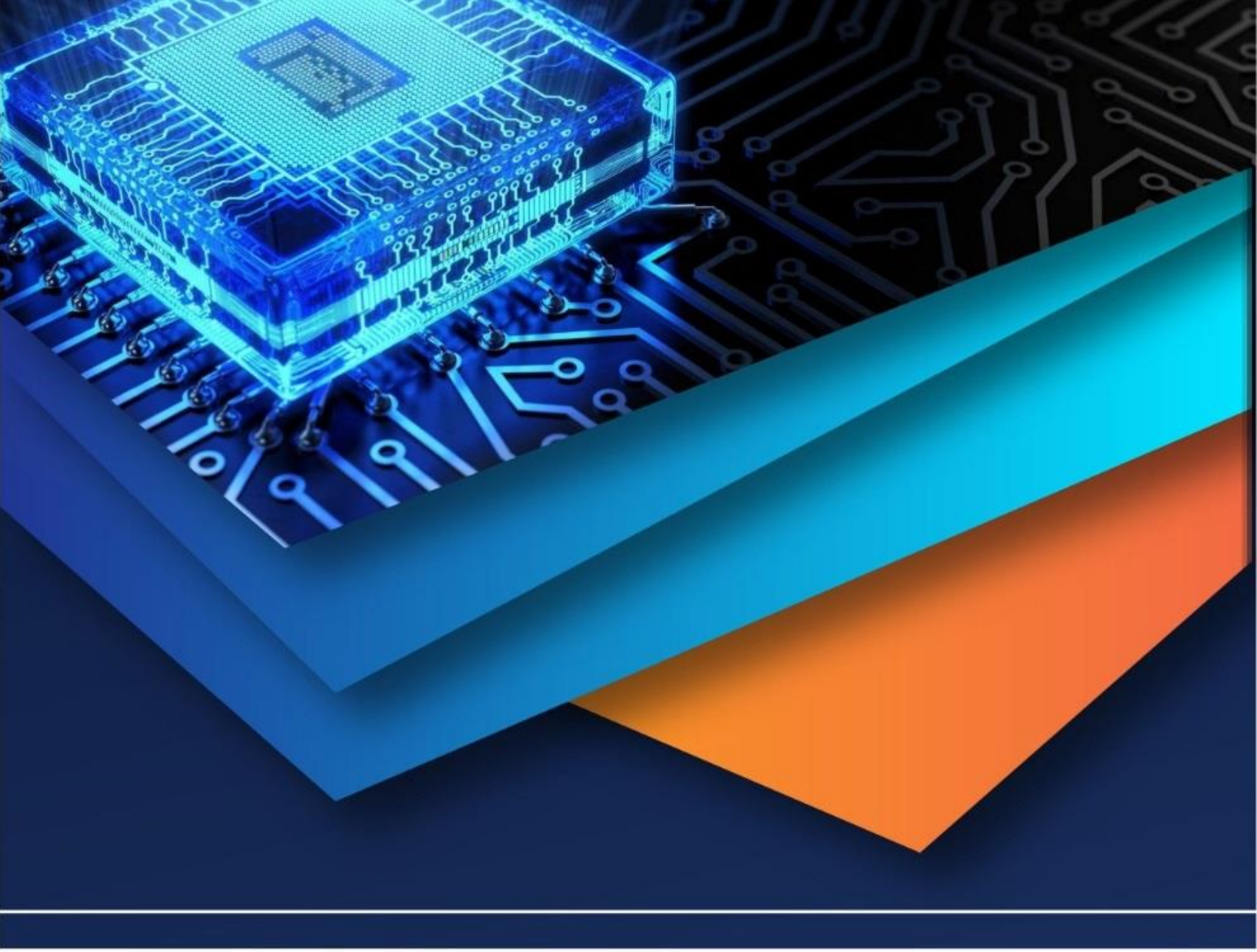
This report looks at eight different papers on facial recognition for attendance tracking. Each one shows unique ways to build such systems. The work draws from computer vision and machine learning tools. It includes YOLO for picking out faces in the frame. CNNs handle the breakdown of facial details. LBPH and PCA come in for matching identities. A few systems pull from live video streams as things happen. Others capture still images at set intervals through a session. The software choices spread out widely. Python paired with OpenCV shows up often. Cloud services like the Azure Face API fill other roles.

The common thread in every paper centers on easing the load of hands-on tasks. Reliability gets a big push forward. Adaptability to various settings stands out as key. Certain projects add mobile options for checking in. Real-time notifications keep everyone updated. Ties to current school platforms make integration smoother.

Diving into these diverse methods reveals the rising role of facial recognition. It automates daily chores like attendance checks. The result goes beyond simple speed. Security strengthens across the board. The whole setup prepares better for what comes next.

REFERENCES

[1] S. Sawhney, K. Kacker, S. Jain, S. N. Singh, and R. Garg, "Real-Time Smart Attendance System using Face Recognition Techniques," 9th Int. Conf. on Cloud Computing, Data Science & Engineering (Confluence), pp. 522–525, 2019.
[2] V. Suresh, S. C. Dumpa, C. D. Vankayala, H. Aduri, and J. Rapa, "Facial Recognition Attendance System Using Python and OpenCV," Quest Journals: Journal of Software Engineering and Simulation, vol. 5, no. 2, pp. 18–29, 2019.
[3] R. Hartanto and M. N. Adji, "Face Recognition for Attendance System Detection," 2018 10th Int. Conf. on Information Technology and Electrical Engineering (ICITEE), pp. 376– 381, IEEE, 2019.
[4] S. Chowdhury, S. Nath, A. Dey, and A. Das, "Development of an Automatic Class Attendance System using CNN-based Face Recognition," 2020 Emerging Technology in Computing, Communication and Electronics (ETCCE), pp. 1–6, IEEE, 2020.
[5] N. Gupta, P. Sharma, V. Deep, and V. K. Shukla, "Automated Attendance System Using OpenCV," 2020 8th Int. Conf. on Reliability, Infocom Technologies and Optimization (ICRITO), Noida, India, pp. 1226–1230, 2020. DOI: 10.1109/ICRITO48877.2020.9197760.
[6] C. C. K. Chan and C.-C. Chen, "Continuous real-time automated attendance system using robust C2DCNN," in Proc. 3rd IEEE Int. Conf. on Knowledge Innovation and Invention (ICKII), 2020, pp. 96–99, doi: 10.1109/ICKII50300.2020.9318855.
[7] P. Pattnaik and K. K. Mohanty, "AI-based techniques for real-time face recognition-based attendance system: A comparative study," in Proc. 4th Int. Conf. on Electronics, Communication and Aerospace Technology (ICECA), 2020, pp. 1034–1039, doi: 10.1109/ICECA49313.2020.9297451.
[8] T. S. Ngo 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, p. 307, 2020.
[9] S. Khan, A. Akram, and N. Usman, "Real Time Automatic Attendance System for Face Recognition Using Face API and OpenCV," Wireless Personal Communications, vol. 113, pp. 469–480, 2020.
[10] H. Yang and X. Han, "Face Recognition Attendance System Based on Real-time Video Processing," IEEE Access, 2020.C. Gomes et al., "Class Attendance Management System using Facial Recognition," ITM Web Conf., vol. 32, p. 02001, 2020. DOI: 10.1051/itmconf/20203202001.
[11] D. Sunaryono et al., "An Android-Based Course Attendance System Using Face Recognition," J. King Saud Univ. - Comput. Inf. Sci., vol. 33, no. 3, pp. 304–312, 2021. DOI:10.1016/j.jksuci.2021.06.001.
[12] A. Singh, S. Bhatt, and A. Gupta, "Automated Attendance System with Face Recognition," International Journal of Engineering Applied Sciences and Technology, vol. 5, no. 12, pp. 233–241, Apr. 2021.
[13] A. Rao, "AttenFace: A Real Time Attendance System Using Face Recognition," International Institute of Information Technology, Hyderabad, arXiv:2211.07582v1 \[cs.CV], 2022.
[14] E. Kagona, "Facial Recognition Attendance Scheme on CCTV Cameras Using Open Computer Vision and Deep Learning: A Case Study of International University of East Africa (IUEA)," Adv. J. of Science, Technology and Engineering, vol. 2, no. 1, pp. 1–27, 2022. DOI: 10.52589/AJSTE-HYVTCZ9E.
[15] M. V. B. Chandrasekhar, C. Vamsi, B. Y. Rao, R. Meghana, and Y. Premkumar, "Automated Attendance System with AI and Real-Time Camera Access using Facial Recognition," Communications on Applied Nonlinear Analysis, vol. 32, no. 9s, 2025.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY