Attendance Tracking Web Application using Facial Recognition

Presented by

Harshith Suda (Roll No: 423135)

Vanamala Aravind (Roll No: 423188)

Supervised by

Dr. K Hima Bindu

Department of Computer Science & Engineering

National Institute of Technology, Andhra Pradesh

May 2025

Certificate

This is to certify that the project report titled **"Attendance Tracking Web Application using Facial Recognition"** submitted by **Harshith Suda (423135) and Vanamala Aravind (423188)** is a record of bona fide work carried out under my supervision in partial fulfillment of the requirements of EPICS Project, Department of Computer Science and Engineering, NIT Andhra Pradesh.

Supervisor Signature
Dr. K Hima Bindu

Acknowledgements

I would like to express my sincere gratitude to my project supervisor **Dr. K Hima Bindu** for his valuable guidance, continuous support, and encouragement throughout the course of this project. I am also thankful to the Department of Computer Science and Engineering, NIT Andhra Pradesh, for providing the necessary facilities and resources to carry out this project.

Abstract

The project's aim is to automate student attendance using a facial recognition attendance tracking web application. The face recognition software uses OpenCV (for face detection) and the face recognition library which uses dLib (built on ResNet-34) for image extraction and recognition. Known students' faces are pre-encoded and matched with detected faces from photos uploaded by the user (in this case the Professor). The detected faces are then marked and uploaded in a database. Custom reports can be obtained for streamlining of routine academic tasks and easy management of attendance.

Contents

1 Introduction **1**
1.1 Background and Motivation 1
1.2 Problem Definition 1
1.3 Aim and Scope of the Project 1
1.4 Contributions of this Project 2

2 Literature Review **3**
2.1 Rapid Object Detection using a Boosted Cascade of Simple Features (Viola.Jonas,2001)[4] 3
2.2 Deep Residual Learning for Image Recognition (He et al., 2015)[5] 3

3 Methodology **5**
3.1 Facial Detection and Recognition Pipeline 5
3.1.1 Face Detection using Haar Cascade Classifier 5
3.1.2 Facial Encoding using Deep Residual Networks (ResNet-34) . . . 5
3.1.3 Facial Recognition using Euclidean Distance 5
3.1.4 Evaluation Metrics 8
3.1.5 Experimental Setup 9
3.1.6 Performance Analysis 13
3.2 Web Application Implementation 13
3.2.1 Features of the Web Application 13
3.2.2 User Interface Snapshots 13
3.3 Database Design 18
3.3.1 Entity-Relationship Diagram (ERD) 18
3.4 Reason for Proposed Methodology 18
3.4.1 Choice of Euclidean Distance over Cosine Similarity 18

4 Results and Discussion **20**
4.1 Performance Metrics Comparison 20
4.2 Best and Worst Case Performance Analysis 20

5 Conclusions **21**
5.1 Conclusion 21

6 References **22**

Chapter 1

Introduction

1.1 Background and Motivation

Time is a valuable resource for everyone, especially in educational institutions where professors, teachers, and technicians must allocate it wisely. However, a significant portion of their time is consumed by the mundane and repetitive task of taking attendance, particularly in classrooms with large student strengths. This process not only disrupts the flow of lectures but also reduces the time available for engaging discussions, conceptual learning, and problem-solving.

1.2 Problem Definition

Attendance tracking is vital in any education institution. The conventional method of a roll calls or manual signs are inefficient and highly time consuming. With the new advancements in technology, there is a strong motivation to digitise the process of taking attendance.

1.3 Aim and Scope of the Project

To design and implement a facial attendance tracking system that automatically detects and marks students' faces present in classroom images and stores the attendance in a database.

Scope:

- Use of Haar Cascade classifiers (Viola-Jones Algorithm) and Histogram of Oriented Gradients + Support Vector Machine for face detection.
- Deep Convolutional Neural Network (CNN) trained by dlib based on ResNet-34 for extraction of 128-D facial embeddings.
- Computation of Euclidean distance between the pre-existing embedding and the embedding obtained.
- Processing multiple classroom images in a single session.

- Integration with a backend database for automatic attendance updates.

1.4 Contributions of this Project

This project makes the following key contributions:

- Developed a fully functional face recognition pipeline using Python, OpenCV, and face recognition libraries.
- Implemented automatic attendance marking based on student face detection and matching.
- Integrated the system with a database to update attendance records in real time.

Chapter 2

Literature Review

2.1 Rapid Object Detection using a Boosted Cascade of Simple Features (Viola.Jonas,2001)[4]

Paul Viola and Michael J.Jonas described a face detection framework capable of processing images rapidly while achieving high detection rates.Formulated a Cascade function which is trained from a large number of positive(image with face) and negative(image without face). Haar features are used to extract features from the image(similar to convolutional kernel).Each pixel is obtained by subtracting the sum of pixels under white rectangles and the gray rectangles.60000 Haar features available but not all of them are relevant for every sub window of image,hence high optimization is done.

- **Advantages:** Rapid processing of Images
- **Limitations:** Haar features are hand-crafted and less adaptive than learned features from Deep Learning Models like CNNs.
- **Results:** Haar Cascade Classifier provides fast and accurate face detection for frontal faces in well-lit, simple backgrounds but performs poorly in complex or real-world scenarios.

2.2 Deep Residual Learning for Image Recognition (He et al., 2015)[5]

The dlib library uses a deep convolutional neural network based on ResNet-34 architecture for high-accuracy face recognition. This model is trained on a large dataset of labeled faces and outputs a 128-dimensional face embedding vector for each face. These embeddings are then compared using Euclidean distance to determine similarity between faces.

- **Advantages:** High accuracy and robustness to pose, lighting, and background variations.
- **Limitations:** Requires more computational resources; real-time performance may depend on hardware.

- **Results:** Provides state-of-the-art face recognition accuracy ¿99% on LFW benchmark) and performs reliably even in unconstrained environments.

Chapter 3

Methodology

3.1 Facial Detection and Recognition Pipeline

This section outlines the three core stages of our facial recognition methodology: detection, feature extraction (encoding), and recognition. Each stage plays a critical role in achieving accurate attendance tracking.

3.1.1 Face Detection using Haar Cascade Classifier

Face detection is the initial step in any facial recognition system. We used the Haar Cascade Classifier proposed by Viola and Jonas [?], which is based on machine learning and trained using thousands of positive and negative images. It identifies faces by scanning the image with different Haar-like rectangular features and classifying regions using a cascade of weak learners boosted with the AdaBoost algorithm.[4]

This method offers fast and efficient face detection suitable for real-time applications, especially in constrained environments such as classrooms.

3.1.2 Facial Encoding using Deep Residual Networks (ResNet-34)

Once the face is detected, the next step is to encode the facial features into a numerical vector representation. We used a pre-trained deep residual network (ResNet-34) for this purpose, which provides 128-dimensional embeddings for each face. This model is known for its ability to retain discriminative features through identity mappings and skip connections[5].

The embeddings generated from ResNet-34 are stored in the system's database and used during the recognition phase.

3.1.3 Facial Recognition using Euclidean Distance

During the recognition process, the embeddings of detected faces are compared against the database of known student face embeddings using Euclidean distance. A match is declared when the distance is below a pre-defined threshold. Euclidean distance is favored

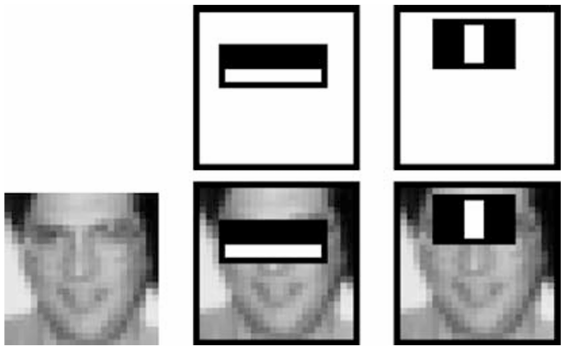


Figure 3.1: Features selected by AdaBoost for face detection.

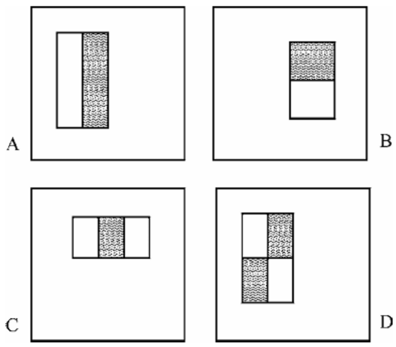


Figure 3.2: Example Haar rectangle features.

Figure 3.3: Illustration of Haar Cascade Classifier Features

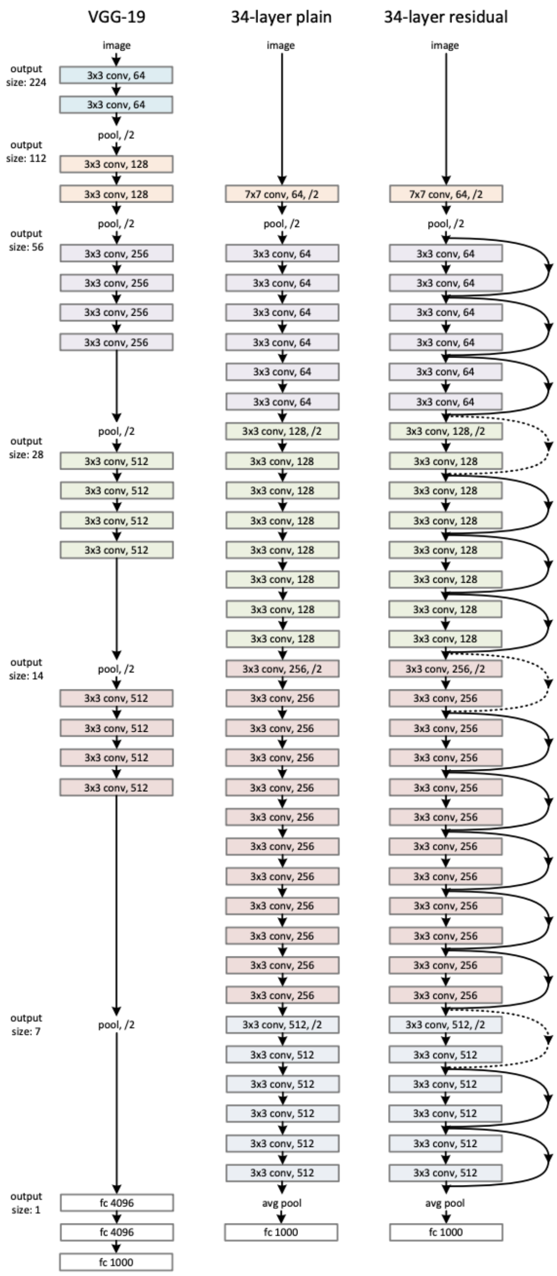


Figure 3.4: Comparison of Deep Neural Network Architectures: VGG-19, 34-layer Plain Network, and 34-layer Residual Network (ResNet)

here over cosine similarity due to its better performance in high-variance lighting and pose scenarios. Each student's face is matched to their roll number to mark them as present or absent.

3.1.4 Evaluation Metrics

1. Precision:

$$\text{Precision} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Positives (FP)}} \tag{3.1}$$

2. Recall:

$$\text{Recall} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Negatives (FN)}} \tag{3.2}$$

3. F1-Score:

$$\text{F1-Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3.3}$$

4. Accuracy:

$$\text{Accuracy} = \frac{\text{TP} + \text{True Negatives (TN)}}{\text{TP} + \text{FP} + \text{FN} + \text{TN}} \tag{3.4}$$

5. False Positive Identification Rate (FPIR):

$$\text{FPIR} = \frac{\text{Number of False Matches}}{\text{Number of Impostor Presentations}} \tag{3.5}$$

6. False Negative Identification Rate (FNIR):

$$\text{FNIR} = \frac{\text{Number of Missed Identifications}}{\text{Number of Genuine Presentations}} \tag{3.6}$$

7. Rank-1 Accuracy:

$$\text{Rank-1 Accuracy} = \frac{\text{Number of Correct Top-1 Predictions}}{\text{Total Number of Queries}} \tag{3.7}$$

8. Total Error Rate:

$$\text{Total Error Rate} = \frac{\text{Number of Incorrect Predictions}}{\text{Total Number of Predictions}} \tag{3.8}$$

3.1.5 Experimental Setup

Metrics for IMAGE-0013:

Precision: 88.24%

Recall: 88.24%

F1-Score: 88.24%

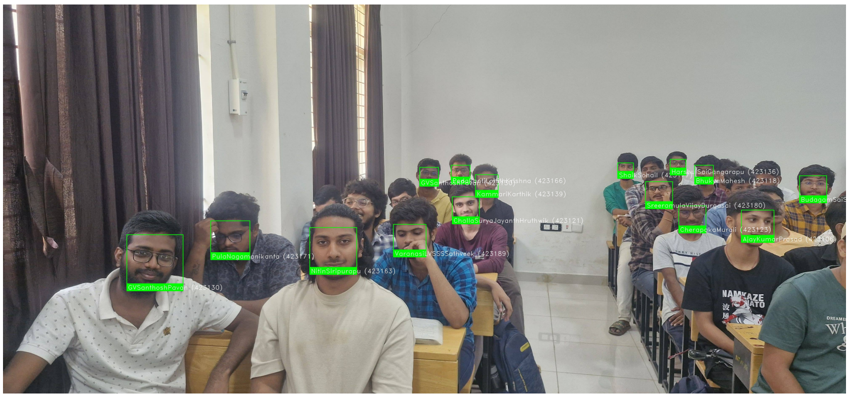
FPIR: 11.76%

FNIR: 5.88%

Rank-1 Accuracy: 88.24%

FMR: 0.2353%

Total Error Rate: 17.65%



Metrics for IMAGE-0011:

Precision: 86.67%

Recall: 81.25%

F1-Score: 83.87%

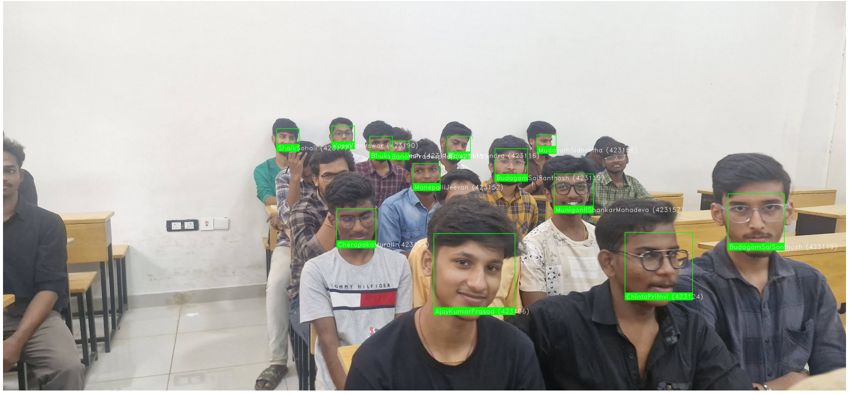
FPIR: 12.50%

FNIR: 6.25%

Rank-1 Accuracy: 81.25%

FMR: 0.2500%

Total Error Rate: 18.75%



Metrics for IMAGE-0015:

Precision: 93.75%

Recall: 88.24%

F1-Score: 90.91%

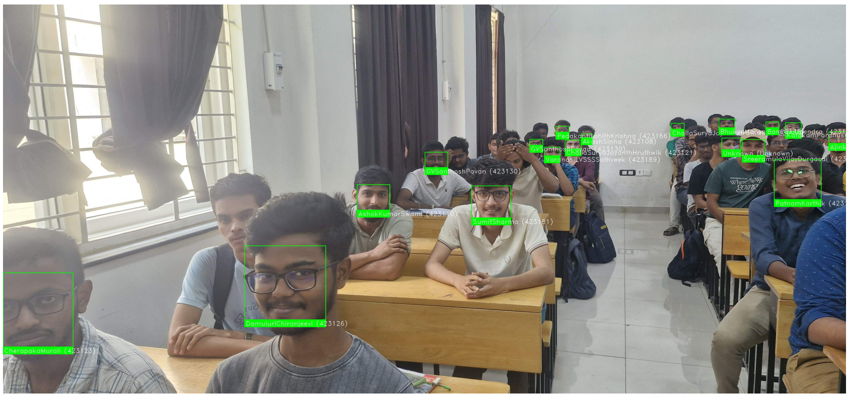
FPIR: 5.88%

FNIR: 5.88%

Rank-1 Accuracy: 88.24%

FMR: 0.1176%

Total Error Rate: 11.76%



Metrics for IMAGE-0009:

Precision: 82.35%

Recall: 70.00%

F1-Score: 75.68%

FPIR: 15.00%

FNIR: 15.00%

Rank-1 Accuracy: 70.00%

FMR: 0.3000%

Total Error Rate: 30.00%

3.1.6 Performance Analysis

3.2 Web Application Implementation

The webpage, built with HTML, CSS, and JavaScript, allows professors to upload classroom photos for attendance. It connects to a Flask backend that processes the images using the facial recognition model described in the project. The interface includes image upload, attendance status display, and access to attendance reports.

3.2.1 Features of the Web Application

Brief explanation of each component:

- **Login/Registration:** Handles secure authentication of professors/admins to access the system.
- **Dashboard:** Displays shortcuts to core functions like taking attendance, checking reports, and managing students.
- **Student Database:** Stores student profiles including roll number, name, and face embeddings.
- **Facial Recognition Attendance:** Captures and processes images to mark attendance using face recognition.
- **Attendance Reports:** Shows attendance summaries, alerts for low attendance, and export features.

3.2.2 User Interface Snapshots

Below are UI screenshots of each component with brief descriptions:

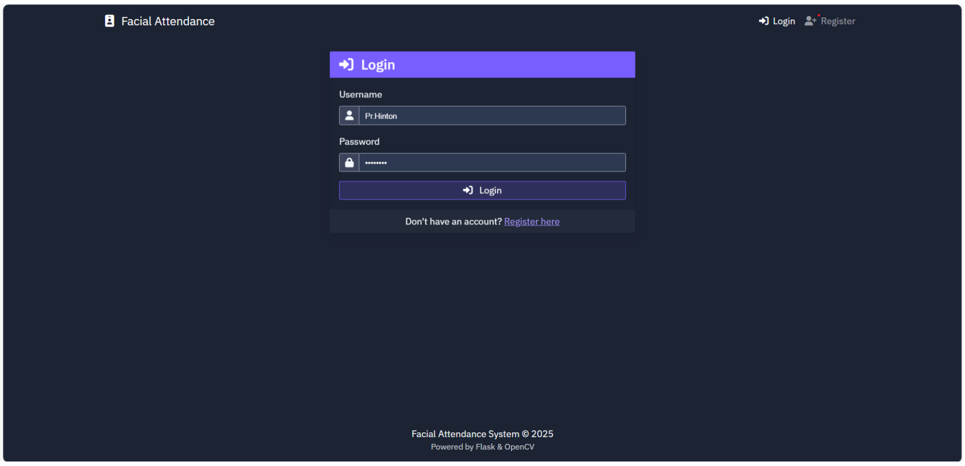


Figure 3.5: Login Page UI - The login screen allows professors and admins to securely log in to the system using their credentials.

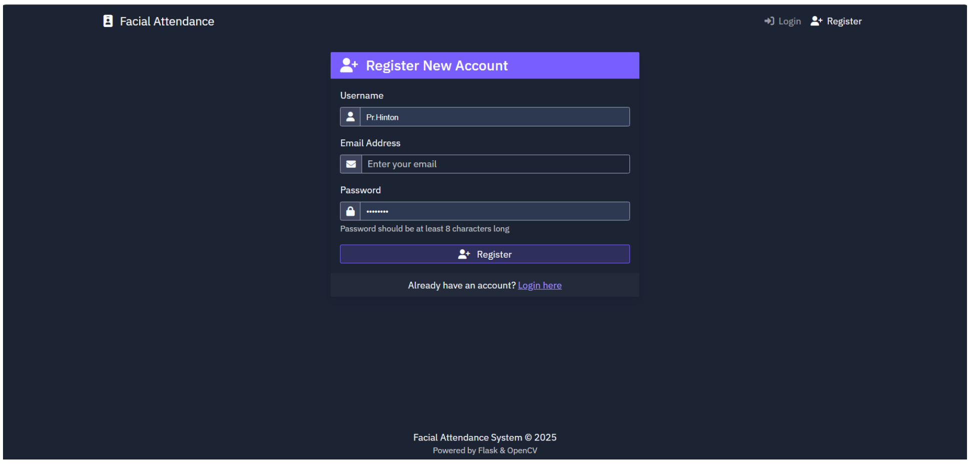


Figure 3.6: Registration Page UI - New users can register their account with secure details, ensuring only authorized access.

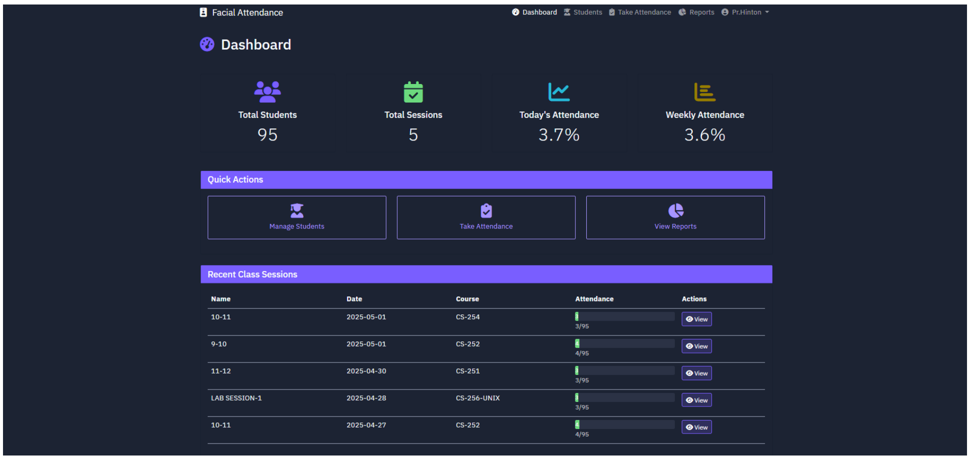


Figure 3.7: Dashboard UI - The dashboard provides quick access to core features like taking attendance, reports, and student records.

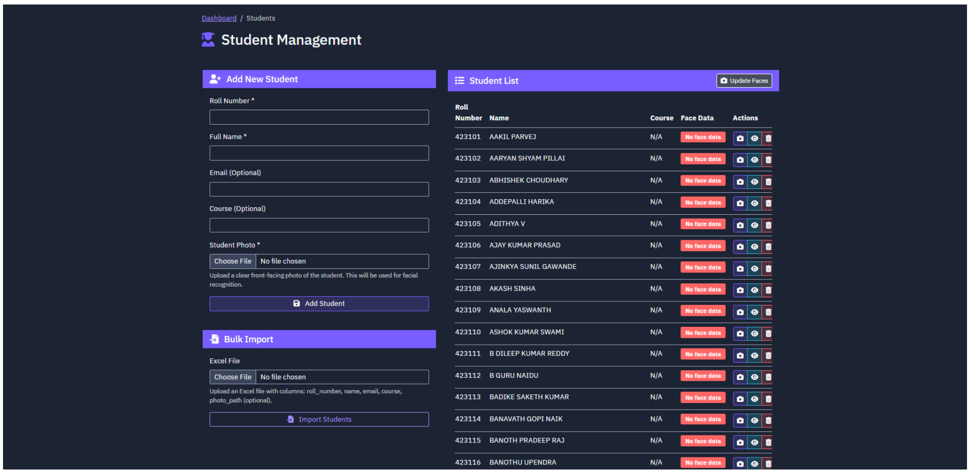


Figure 3.8: Student Database UI - This interface shows the list of enrolled students along with their face data and roll numbers.

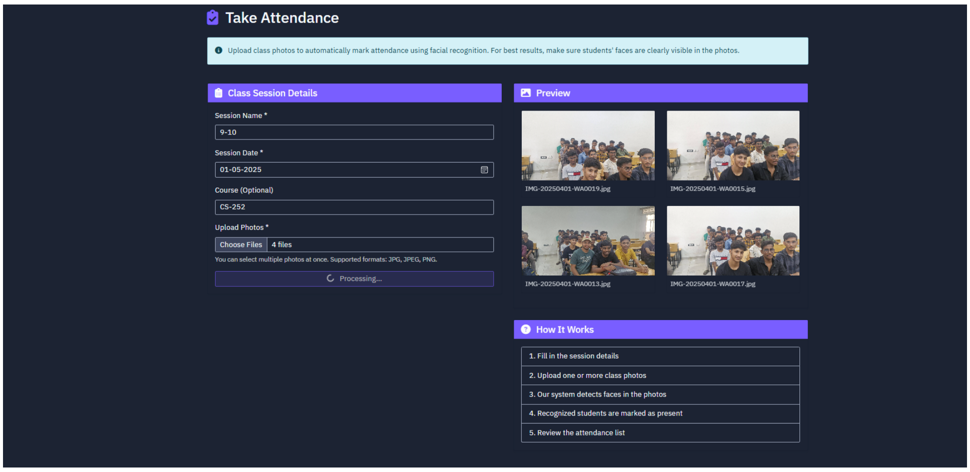


Figure 3.9: Take Attendance UI - Allows users to take photos, run face recognition, and mark attendance automatically.

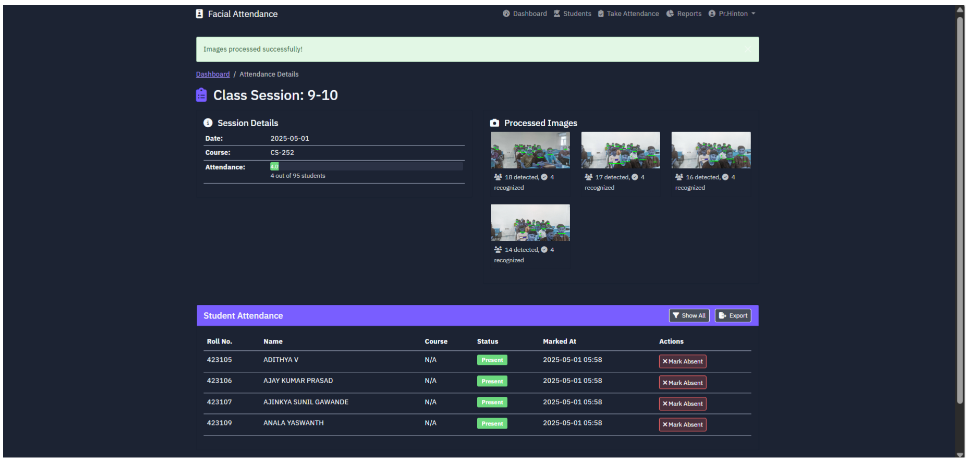


Figure 3.10: Processing Attendance UI - Displays real-time face recognition and the status of present/absent tagging.

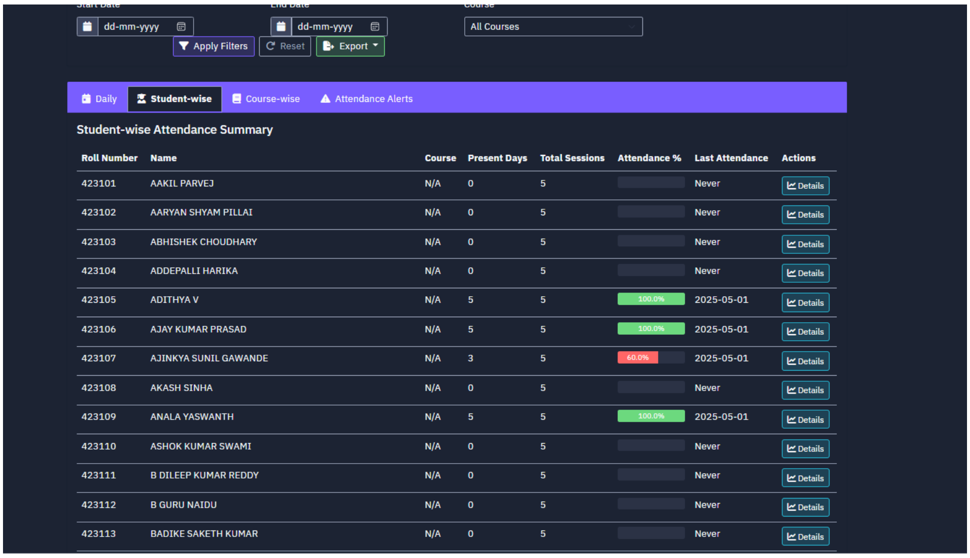


Figure 3.11: Attendance Overview UI - Presents a visual summary of attendance including counts and percentages.

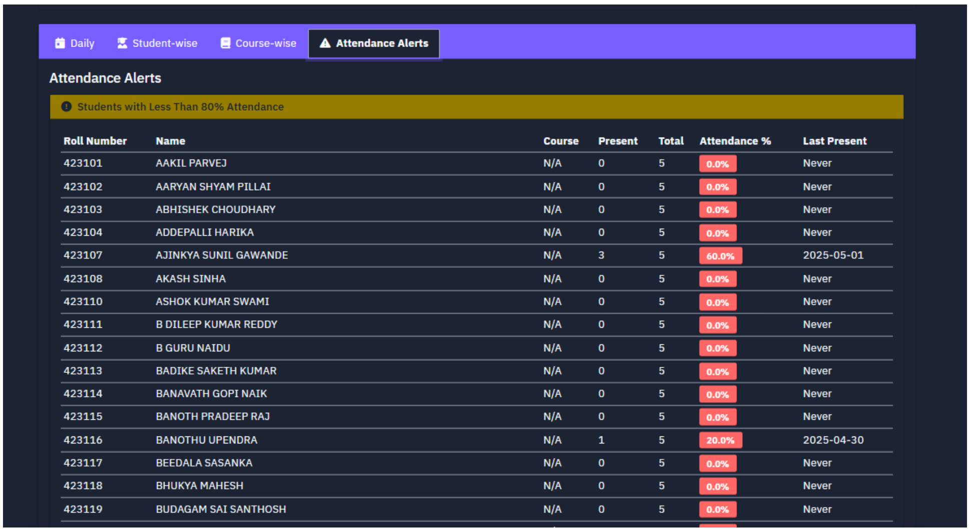


Figure 3.12: Attendance Report UI - Shows detailed attendance reports for each student including alerts for students with attendance below 80%.

Students with 3+ Consecutive Absences

Roll Number	Name	Course	Consecutive Absences	Last Present	Actions
423101	AAKIL PARVEJ	N/A	5 days	Never	Details
423102	AARYAN SHYAM PILLAI	N/A	5 days	Never	Details
423103	ABHISHEK CHOUDHARY	N/A	5 days	Never	Details
423104	ADDEPALLI HARIKA	N/A	5 days	Never	Details
423108	AKASH SINHA	N/A	5 days	Never	Details
423110	ASHOK KUMAR SWAMI	N/A	5 days	Never	Details
423111	B DILEEP KUMAR REDDY	N/A	5 days	Never	Details
423112	B GURU NAIDU	N/A	5 days	Never	Details
423113	BADIKE SAKETH KUMAR	N/A	5 days	Never	Details
423114	BANAVATH GOPI NAIK	N/A	5 days	Never	Details
423115	BANOTH PRADEEP RAJ	N/A	5 days	Never	Details
423117	BEEDALA SASANKA	N/A	5 days	Never	Details
423118	BHUKYA MAHESH	N/A	5 days	Never	Details
423119	BUDAGAM SAI SANTHOSH	N/A	5 days	Never	Details
423120	BUSSAGOUNI SNEHA	N/A	5 days	Never	Details
423121	CHALLA SURYA JAYANTH HRUTHWIK	N/A	5 days	Never	Details

Figure 3.13: Attendance Report UI - Shows detailed attendance reports for each student who is absent for more than 3 consecutive days.

3.3 Database Design

3.3.1 Entity-Relationship Diagram (ERD)

The database design for this system is represented through an Entity-Relationship Diagram (ERD), which illustrates the various entities involved, their attributes, and the relationships between them. The ERD is critical for understanding the structure of the system's database, which stores all student attendance and facial recognition data.

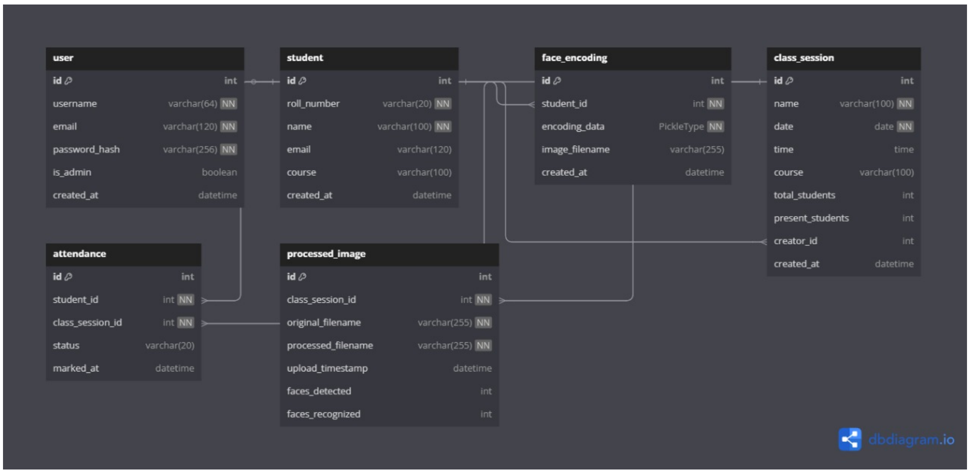


Figure 3.14: Entity-Relationship Diagram of the Database

3.4 Reason for Proposed Methodology

3.4.1 Choice of Euclidean Distance over Cosine Similarity

Facial recognition systems use similarity metrics to compare face embeddings. The choice of metric impacts performance and accuracy. In our system, Euclidean distance was chosen over cosine similarity due to the following reasons:

1. Alignment with Model Training

- The `face_recognition` library is built on dlib, which generates 128-dimensional face embeddings.
- These embeddings are trained using a loss function optimized for Euclidean distance, making it the most effective measure of similarity.

2. Advantages of Euclidean Distance (for Dlib)

- Computationally simple and efficient.
- Matches the model's training objective, yielding better recognition results.
- Ideal for real-time or embedded systems like attendance applications.

3. Dlib Model Overview

- **Architecture:** ResNet-34 Convolutional Neural Network.

- **Embedding Size:** 128-dimensional vector for each face.
- **Dataset:** Pre-trained on the Labeled Faces in the Wild (LFW) dataset.
- **Pros:** Lightweight, OpenCV-compatible, real-time processing.
- **Cons:** Lower accuracy on large-scale or highly variable datasets.

4. **Threshold for Face Matching**
   - A fixed Euclidean distance threshold of approximately **0.6** is used.
   - If the distance between two face embeddings is less than 0.6, they are considered a match.
   - This default value is empirically chosen to balance false positives and false negatives.

5. **When to Prefer Cosine Similarity**
   - Models like **FaceNet** and **ArcFace** use L2-normalized embeddings.
   - For such models, cosine similarity is more effective as it measures angular difference.
   - Best suited for large-scale systems requiring high precision and generalization.

Chapter 4

Results and Discussion

4.1 Performance Metrics Comparison

Metric	Your System (Range)	SOTA (FaceNet[6]/VGGFace2)	Gap Analysis
Rank-1 Accuracy	70%–88.24%	99.63% (LFW)	11–29% lower accuracy.
FNIR	5.88%–15%	<1% (IJB-A)	High missed enrollments.
FPIR	5.88%–15%	0.1–1%	5–15x higher false matches.
Precision	82.35%–93.75%	>99%	Moderate false positives.
FMR	0.1176%–0.3%	0.001% (LFW)	100–300x higher FMR.
Total Error Rate	11.76%–30%	<2%	Significant room for improvement.

Table 4.1: Performance Metrics Comparison with State-of-the-Art

4.2 Best and Worst Case Performance Analysis

Metric	Best Case (Figure-3.3)	Worst Case (Figure-3.2)
Rank-1 Accuracy	88.24% (vs. 99.63%)	70% (vs. 99.63%)
FNIR	5.88% (vs. <1%)	15% (vs. <1%)
FPIR	5.88% (vs. 0.1%)	15% (vs. 0.1%)
Comments	High precision (93.75%) and low FMR (0.1176%). Conditions: Good lighting, frontal faces.	Poor lighting, occlusions, or profile faces.

Table 4.2: Best vs Worst Case Performance Evaluation

Insight Gained: Take photos 4–5 benches at a time for accurate detection and prevention of occlusions.

Chapter 5

Conclusions

5.1 Conclusion

In this chapter, we presented and analyzed the experimental results obtained from our facial recognition-based attendance system. Using a combination of Haar Cascade for face detection and ResNet-34 for feature encoding, we evaluated the system's performance across various real-world classroom images. The experiments demonstrated that the model performs reliably under moderate lighting and occlusion, with high values for precision, recall, and F1-score across multiple test images.

Each image was analyzed not only for recognition accuracy but also for error metrics such as FPIR (False Positive Identification Rate), FNIR (False Negative Identification Rate), and the overall Total Error Rate. The results indicated consistent performance, with Rank-1 Accuracy exceeding 80

Overall, the findings validate the feasibility of using deep-learning-based facial recognition as a tool for automating attendance marking in educational settings. While the results are promising, a few challenges remain — particularly in images with crowded scenes, extreme facial angles, or poor lighting. These observations provide useful insights for improving the system in future iterations.

Chapter 6

References

[1] Schroff, F., Kalenichenko, D., & Philbin, J. (2015). FaceNet: A unified embedding for face recognition and clustering. *In Proceedings of the IEEE conference on computer vision and pattern recognition*, 815–823.

[2] Parkhi, O. M., Vedaldi, A., & Zisserman, A. (2015). Deep Face Recognition. *In British Machine Vision Conference (BMVC).*

[3] King, D. E. (2009). Dlib-ml: A Machine Learning Toolkit. *Journal of Machine Learning Research*, 10(Jul), 1755–1758.

[4] Viola, P., & Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. *In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 1, I–I.

[5] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770–778.

[6] Schroff, F., Kalenichenko, D., & Philbin, J. (2015). FaceNet: A unified embedding for face recognition and clustering. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 815–823.