IIITD Gate Entry Automation

Manan Aggarwal
IIIT Delhi
manan22273@iiitd.ac.in

Shobhit Raj
IIIT Delhi
shobhit22482@iiitd.ac.in

Souparno Ghose
IIIT Delhi
souparno22506@iiitd.ac.in

Abstract

This project presents a computer vision system designed to automate student registration at IIITD's gate after 10 PM under low-light conditions. We address the challenge by decomposing the task into face detection from CCTV video frames and subsequent face recognition. A manually curated dataset of 150 images from 50 batchmates, with three views per person and detailed bounding box annotations, captures the real-world variations at our college. The system processes frames extracted from nighttime video through a sequential pipeline that first uses DAI for face detection, an MTCNN + AdaFace backbone for direct recognition, and ByteTrack with majority-vote merging to smooth temporal IDs. Targeted optimizations like frame skipping interpolation, and a faster NMS, make it capable of real-time inference, while also achieving recognition accuracy of 87.5%.

1. Problem Statement

At **IIIT Delhi**, students entering or exiting the campus after 10 PM must manually log their entry in a physical register. With high student traffic, this process often results in long wait times of up to 20 minutes, causing inconvenience to both students and security personnel. Automating this process with a **computer vision-based face recognition system** can significantly improve efficiency and security.

To address this, we propose a fully automated face recognition system capable of identifying students even under **low-light conditions** and **varying poses**. Our system takes a **live video feed** from CCTV/security cameras at the IIITD gate as input and outputs an automatic entry in an online database, capturing details such as the student's name, roll number, and time of entry or exit. The primary users will be security guards and administrative staff, who can monitor **live feeds** and **real-time entry logs** via a **web-based interface**.

2. User Interface & progress

2.1. User Interface

Our web interface is developed using React & TypeScript for the frontend, Flask for the backend and a SQLite database. The interface consists of two main panels. The first is the Dashboard, which displays student entry/exit logs including their name, roll number, time, and photo. The second is the Live Surveillance Panel, which shows the real-time CCTV video feed from the IIITD gate. We are leveraging Bolt AI LLM [1] as an AI-based coding assistant to upspeed development and refine our codebase. This interface enables security and administrative personnel to monitor campus access efficiently through a clear, intuitive, and responsive web-based dashboard.

2.2. Progress and Methodology

The flowchart representing our current methodology and future plans is given in Fig 1

Our current implementation includes:

1. **Low-Light Face Detection:** Frames are first passed through the DAI detector to localize face bounding boxes under challenging illumination.
2. **Joint Feature Extraction & Recognition:** Each detected face is fed directly into an MTCNN backbone for landmark-aligned feature extraction, followed by the AdaFace model's quality-adaptive margin classifier for immediate identity prediction.
3. **Track-Based Aggregation:** Detections are linked across consecutive frames using ByteTrack to form face tracks. A majority-vote over all AdaFace predictions within each track yields a robust final identity assignment for that track.

3. Related Work

Several recent studies have tackled the challenge of low-light face detection and recognition by addressing domain gaps, illumination variations, and pose discrepancies.

HLA-Face: Joint High-Low Adaptation for Low Light Face Detection [12] introduces a novel framework that bridges the gap between normal and low-light images. Its key contribution is a bidirectional low-level adaptation scheme that brightens low-light images while simultaneously distorting normal-light images to generate intermediate states. Complementing this, the paper employs a multi-task high-level adaptation strategy through self-supervised learning to align semantic features across domains.

Boosting Object Detection with Zero-Shot Day-Night Domain Adaptation [4] proposes a zero-shot domain adaptation framework specifically designed to handle the scarcity of low-light annotated data. The method trains object detectors solely on well-lit images and then generalizes them to low-light scenarios by learning illumination-invariant features. Central to this approach is a Retinex-based reflectance representation learning module combined with an interchange-redecomposition-coherence procedure that reinforces the stability of the learned representations. This strategy effectively circumvents the challenges of collecting and annotating low-light data, and its insights are directly applicable to designing robust detection components for low-light environments.

Low-FaceNet: Face Recognition-Driven Low-Light Image Enhancement [5] presents a unified deep learning framework that jointly tackles low-light image enhancement and face recognition. The approach integrates an image enhancement network with four distinct modules—contrastive learning, feature extraction, semantic segmentation, and face recognition—to counteract the effects of underexposure and color bias. By leveraging unpaired normal-light and low-light images as positive and negative samples, Low-FaceNet effectively preserves critical facial features while enhancing image quality.

Pose Attention-Guided Profile-to-Frontal Face Recognition [10] addresses the issue of extreme pose variation, a common problem in unconstrained face recognition scenarios. The paper introduces a unified coupled network that incorporates a novel Pose Attention Block (PAB). This module leverages pose information as auxiliary guidance by employing both channel and spatial attention mechanisms to extract pose-invariant, discriminative features. The network is trained using a class-specific contrastive loss to map profile and frontal faces into a common embedding space. The innovative use of pose attention not only mitigates the accuracy drop caused by pose discrepancies but also informs strategies for handling diverse facial orientations.

Controllable and Guided Face Synthesis for Unconstrained Face Recognition [9] introduces a generative framework that synthesizes high-quality, identity-preserving face images under unconstrained conditions. By integrating guidance signals—including pose, expression, and illumination—into a generative adversarial model, the method is capable of reducing the domain gap between unconstrained and controlled environments. This controllable synthesis not only aids in augmenting training datasets with diverse samples but also improves recognition accuracy when input images are of low quality.

AdaFace: Quality Adaptive Margin for Face Recognition [8] introduces a novel approach to enhancing face recognition performance, particularly in low-quality image scenarios. The authors propose an adaptive margin function that adjusts the emphasis on training samples based on their image quality, approximated through feature norms. This method allows the model to focus more on informative samples while de-emphasizing those that are less reliable due to poor quality. By integrating image quality into the loss function, AdaFace effectively balances the learning process, leading to improved recognition accuracy across various challenging datasets.

For our baseline, we primarily build upon the domain adaptation and guided synthesis strategies from [12] and [4] for the face detection module, while utilizing the enhancement and recognition techniques from [5]. Additionally, the pose attention mechanism from [10] and the data augmentation insights from [9] are incorporated to address pose variations and improve overall system robustness. These combined methodologies provide a comprehensive foundation to tackle the challenges of low-light face detection and recognition for our task.

4. Datasets & Evaluation Metrics

4.1. Dataset

For our project, we manually curated a dedicated dataset by collecting three face photos (frontal, left profile, and right profile, an example is shown in Fig. 2) from **50** IIITD batchmates, resulting in a total of **150** images. Our test set was constructed by recording 5 videos at the college gate during nighttime under low-light conditions. Frames were then extracted from these videos to evaluate our system in a realistic setting. Each image of the test set was manually annotated with precise bounding boxes using an annotation tool named ImgLab [2], thereby creating reliable ground truth data.

Exploratory data analysis (EDA) results on the dataset is in table 1 with accompanying graphs in Fig. 7

Our dataset comprises 50 persons and 150 images, with an average resolution of approximately **2232x3043** pixels, a brightness range from **72.94** to **193.93**, and a blur score range from **7.92** to **1244.84**. These statistics highlight a broad variability in image quality and illumination, reflecting the real-world conditions encountered at our college gate.

4.2. Evaluation Metrics

For evaluation, we employ standard metrics tailored to each module in our pipeline:

- **Face Detection:** primarily evaluated using the **Intersection over Union (IoU)**, to assess the overlap between predicted and ground-truth bounding boxes. In addition, precision, recall, and F1-score are reported for detection accuracy.
- **Face Recognition:** **Accuracy** is used as the principal metric, while F1-score along with true positive, false positive, and false negative counts are also documented to evaluate identification performance and robustness.

5. Analysis of Results

In our evaluation, we experimented with two detection models DAI [4] and HLA [12] using both single-scale and multi-scale settings under varying lighting conditions (Normal, Light, and Dark). As shown in Tables 3, 4, and 5, the DAI model generally outperformed the HLA model in terms of precision, recall, and F1 score across all conditions. Resource usage was also tracked while running these models which is further summarized in Table 2. These results suggest that the DAI model is more robust to variations in illumination, making it better suited for our application.

We tracked the resource usage of the LowFaceNet enhancer [5], we have listed the results in Table 6. For the recognition stage, we compared five models: LowFaceNet [5], Pose Attention Guided model [10], QMagFace [11], ArcFace [3] and AdaFace [8]. The results of these experiments are highlighted in Table 7.

We carried out a series of targeted experiments and system level optimizations to both improve recognition accuracy and dramatically accelerate our pipeline. The key findings are summarized below:

1. **Recognition Accuracy AdaFace** [8] **vs. LowFaceNet** [5]: By replacing our previous enhancement-then-recognition approach with an end-to-end AdaFace backbone, we saw face recognition accuracy jump from 12% (with LowFaceNet) to 59% on our low-light IIITD test set. We also benchmarked other state-of-the-art losses, the results can be seen in Table 7. QMagFace [11] and ArcFace [3] which each outperformed LowFaceNet, but AdaFace's quality-adaptive margin proved the most effective for degraded inputs.

2. **Whole-Pipeline Latency Reduction**: Originally, processing a 7s, 30 fps video took over 2 minutes. By skipping every k=6 frames during DAI detection while skipping m=3 frames for the entire pipeline, and linearly interpolating bounding-boxes between successive detections, we cut total runtime to 11 s, The results of these experiments are summarized in Table 8. Further optimizations within the DSFD backbone, caching the per-image "prior" tensor (which is constant for our fixed frame size), swapping our Python NMS for TorchVision's optimized C implementation, and applying the confidence threshold (0.6) before NMS to reduce candidate boxes from 9000 to 20–30 brought end-to-end time down to 8.37s. This ideas was inspired by [7]

3. **Model Quantization**: We quantized the DAI detector (DSFD) from FP32 to FP16, slashing its GPU memory footprint from 190 MB to 95 MB without measurable accuracy loss. This halved both model size and memory bandwidth requirements, further improving throughput for our GPU resources.

4. **Alternative Deblurring Trials**: To explore explicit image restoration, we evaluated a GAN-based deblurring module KeepGAN [6]. Although it produced visually sharper faces, the generative distortions degraded recognition performance by introducing subtle feature artifacts, as can also be seen in Figures 12, 13, 14.

5. **Track-Level Merging Technique**: To address ID fragmentation when skipping frames, we introduced a majority-vote merging strategy: any two track fragments that receive the same AdaFace identity in several frames are unified into a single track. This reduced spurious ID switches and improved the consistency of our final entry logs. These results are also shown in Table 8

Together, these enhancements, model replacement, frame-skipping interpolation, DSFD micro-optimizations, quantization, and careful pruning of unnecessary preprocessing have transformed our prototype into a much more accurate and responsive system, ready for real-time deployment and further multi-camera integration.

Error Analysis

To better understand the limitations of our pipeline, we systematically examined failure modes in each of its three core modules.

Detection Errors (DAI) Under very low illumination, the DAI detector occasionally produces spurious bounding boxes, so-called "hallucinations" around dark patches or background clutter (Figure 8). While these false positives do not significantly degrade overall performance (they are pruned by our track-level voting scheme), they can momentarily increase computational load and risk minor misassignments in high-traffic frames.

Recognition Errors (AdaFace) The AdaFace recognizer shows high accuracy on near-frontal, well-focused faces but degrades on extreme side profiles and heavily blurred crops (Figure 9). We experimented with GAN-based deblurring [6], yet although visual sharpness improved, feature distortions led to no net gain—in some cases.

Tracking Errors (ByteTrack + Interpolation) To accelerate detection, we skip k frames between DAI inferences and linearly interpolate bounding boxes. However, if a subject moves more than the tracker's association threshold during an interval, ByteTrack spawns a new track ID (Figure 10). We merge identities across track fragments by majority voting, but when the recognizer itself mislabels one fragment (for instance, during a brief occlusion), the merged identity can become incorrect (Figure 11).

6. Compute Requirements

We are able to manage the compute requirements for our project using our local setup. The development and testing are supported by the following hardware configuration:

- **GPU:** Nvidia GeForce GTX 1650 Super 4 GB GPU
- **CPU:** AMD Ryzen 5 3500 (6 Cores / 6 Threads) @ 3.6 GHz
- **Memory:** 16 GB RAM

These resources were sufficient for our project's development and experimentation needs.

7. Individual Tasks

Each team member contributed collaboratively while also taking ownership of specific components of the project. The individual responsibilities are outlined in Table 9.

8. Next Steps

Future work should focus on extending beyond pedestrian entry to also support vehicle entry, capturing drivers as they enter into the campus and also track delivery drivers as they arrive and depart. Integration of handling occlusions for instances of busy gate scenes and exploring occlusion aware techniques to recover lost identities. Refining detection and recognition modules to boost accuracy while preserving sub second per-frame latency. At last, the web interface can be enhanced to present consolidated pedestrian and vehicle logs in real time, providing security staff and administrators with a seamless monitoring tool.

References

[1] Bolt ai. https://bolt.new/. [Accessed 27-03-2025]. 1

[2] GitHub - NaturalIntelligence/imglab: To speedup and simplify image labeling/ annotation process with multiple supported formats. — github.com. https://github.com/NaturalIntelligence/imglab. 2

[3] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4685–4694, 2019. 3

[4] Zhipeng Du, Miaojing Shit, and Jiankang Deng. Boosting object detection with zero-shot day-night domain adaptation. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12666–12676, 2024. 2, 3

[5] Yihua Fan, Yongzhen Wang, Dong Liang, Yiping Chen, Haoran Xie, Fu Lee Wang, Jonathan Li, and Mingqiang Wei. Low-facenet: Face recognition-driven low-light image enhancement. *IEEE Transactions on Instrumentation and Measurement*, 73:1–13, 2024. 2, 3

[6] Ruicheng Feng, Chongyi Li, and Chen Change Loy. Kalman-inspired feature propagation for video face super-resolution. In *European Conference on Computer Vision (ECCV)*, 2024. 3, 4

[7] Hukkelas. Hukkelas/dsfd-pytorch-inference: A high-performance pytorch implementation of face detection models, including retinaface and dsfd. 3

[8] Minchul Kim, Anil K. Jain, and Xiaoming Liu. Adaface: Quality adaptive margin for face recognition. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18729–18738, 2022. 2, 3

[9] Feng Liu, Minchul Kim, Anil Jain, and Xiaoming Liu. Controllable and guided face synthesis for unconstrained face recognition. In *ECCV*, 2022. 2

[10] Moktari Mostofa, Mohammad Saeed Ebrahimi Saadabadi, Sahar Rahimi Malakshan, and Nasser M. Nasrabadi. Pose attention-guided profile-to-frontal face recognition. In *2022 IEEE International Joint Conference on Biometrics (IJCB)*, pages 1–10, 2022. 2, 3

[11] Philipp Terhörst, Malte Ihlefeld, Marco Huber, Naser Damer, Florian Kirchbuchner, Kiran Raja, and Arjan Kuijper. Qmagface: Simple and accurate quality-aware face recognition. In *2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 3473–3483, 2023. 3

[12] Wenjing Wang, Wenhan Yang, and Jiaying Liu. Hla-face: Joint high-low adaptation for low light face detection. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16190–16199, 2021. 2, 3

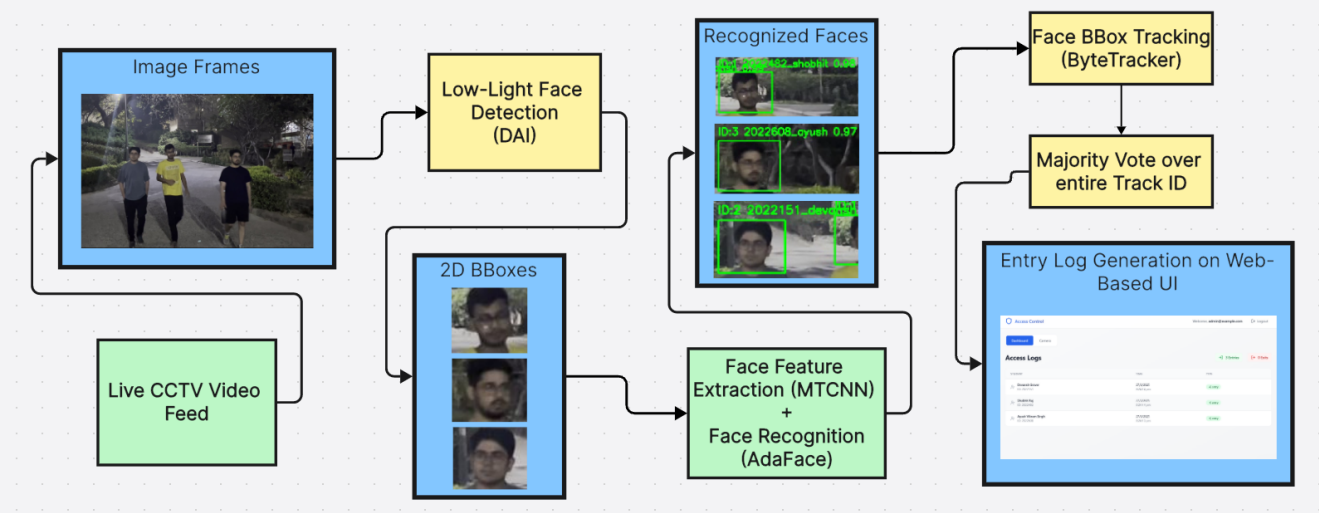


Figure 1. Complete Methodology Pipeline

Figure 2. Different Poses Captured in the dataset

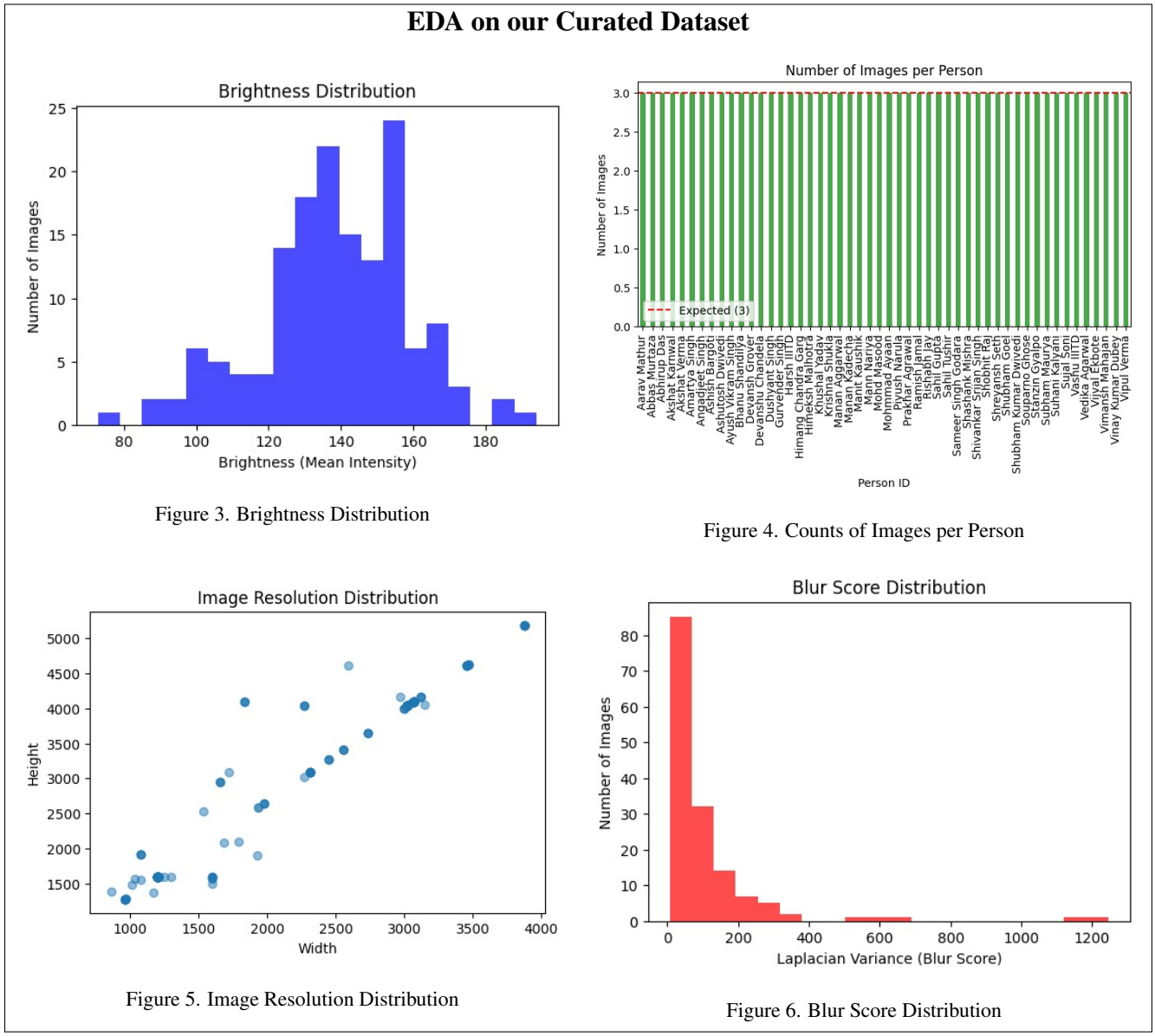


Figure 3. Brightness Distribution

Figure 4. Counts of Images per Person

Figure 5. Image Resolution Distribution

Figure 6. Blur Score Distribution

Figure 7. Exploratory Data Analysis (EDA) on our Curated Dataset

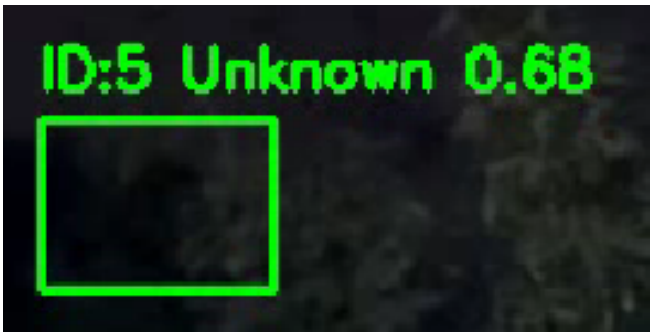


Figure 8. Error in DAI (Random object is Marked)

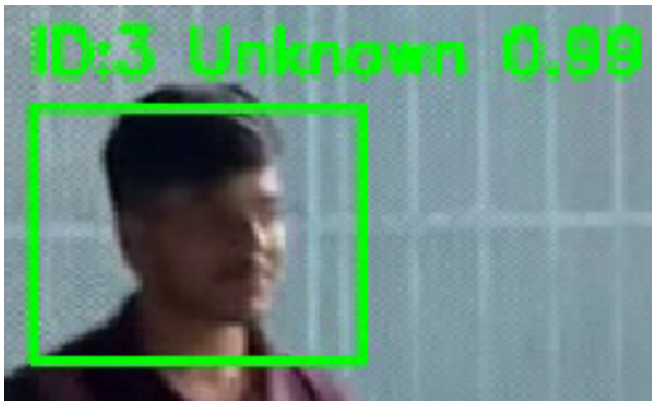


Figure 9. Error in AdaFace (Known image is labeled Unknown)

Figure 10. Error In Tracking when Skipping Detection Iterations

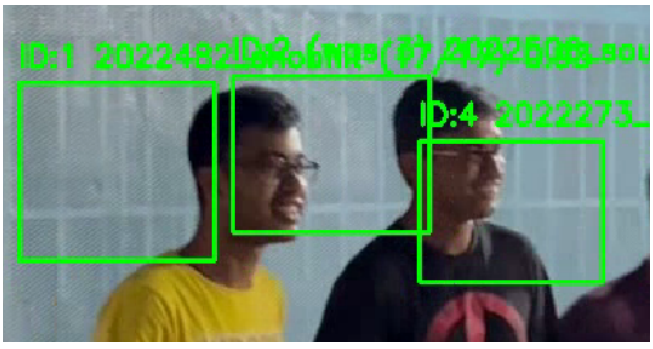


Figure 11. Error In Tracking when Merging Trackers

(a) Dataset Image

(b) Image from Frame

(c) KEEP Generated

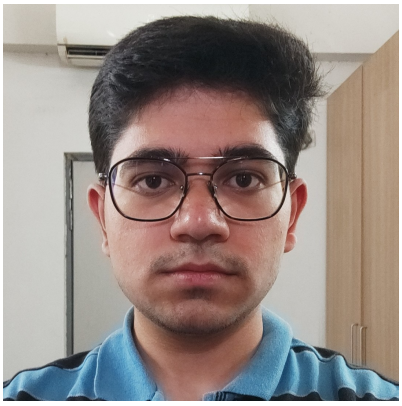
Figure 12. KEEP Generated Images

(a) Dataset Image

(b) Image from Frame

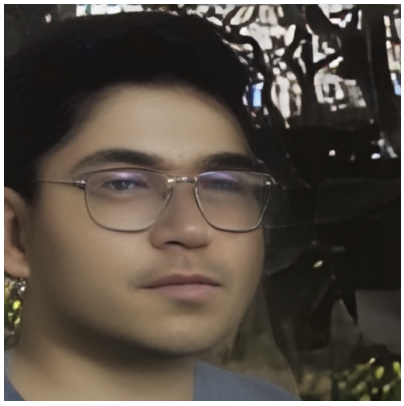
(c) KEEP Generated

Figure 13. KEEP Generated Images



(a) Dataset Image

(b) Image from Frame



(c) KEEP Generated

Figure 14. KEEP Generated Images

Table 1. EDA Summary of the Curated Dataset

Metric	**Mean**	**Variance**	**Min**	**Max**	**Std. Dev.**
Width (pixels)	2232.69	820810.40	860.00	3880.00	905.99
Height (pixels)	3043.53	1525638.37	1280.00	5184.00	1235.17
Brightness (intensity)	138.27	426.53	72.94	193.93	20.65
Blur Score	108.38	26740.06	7.92	1244.84	163.52

Table 2. Compute Resource Usage for DAI and HLA Models

Model	Scale Mode	x-size	GPU Memory (MB)	Time	RAM (GB)
DAI	With Multi-Scale	640	1760	9m0s	1.35
	Without Multi-Scale	1440	1538	1m25s	1.24
HLA	With Multi-Scale	640	2526	4m16s	1.1625
	Without Multi-Scale	1440	2696	1m8s	1.0625

Table 3. Detection Metrics under Normal Conditions

Model	Scale	TP	FP	FN	Precision	Recall	F1 Score
DAI	Single Scale	280	17	63	0.9428	0.8163	0.8750
HLA	Single Scale	241	54	102	0.8169	0.7026	0.7555
DAI	Multi Scale	282	25	61	0.9186	0.8222	0.8677
HLA	Multi Scale	199	53	144	0.7897	0.5802	0.6689

Table 4. Detection Metrics under Light Conditions

Model	Scale	TP	FP	FN	Precision	Recall	F1 Score
DAI	Single Scale Light	115	1	4	0.9914	0.9664	0.9787
HLA	Single Scale Light	101	13	18	0.8860	0.8487	0.8670
DAI	Multi Scale Light	115	6	4	0.9504	0.9664	0.9583
HLA	Multi Scale Light	104	11	15	0.9043	0.8739	0.8889

Table 5. Detection Metrics under Dark Conditions

Model	Scale	TP	FP	FN	Precision	Recall	F1 Score
DAI	Single Scale Dark	165	16	38	0.9116	0.8128	0.8594
HLA	Single Scale Dark	140	41	63	0.7735	0.6897	0.7292
DAI	Multi Scale Dark	167	19	36	0.8978	0.8227	0.8586
HLA	Multi Scale Dark	95	42	108	0.6934	0.4680	0.5588

Table 6. Compute Resource Usage for LowFaceNet Enhancer Module

Module	**GPU Memory (MB)**	**RAM (GB)**	**Time**
LowFaceNet Enhancer	352	0.91	6.5 s

Table 7. Compute and Performance Metrics for Classification Modules

Model	GPU (MB)	RAM (GB)	Accuracy	Precision	Recall	F1 Score	TP	FP	Time (s)
LowFaceNet	322	1.023	0.1286	0.1292	0.9643	0.2278	27	182	5.52
Pose Attention	1296	1.19	0.0329	0.0352	0.3333	0.0637	8	219	5.10
ArcFace	476	1.085	0.1893	0.1893	1.0000	0.3183	46	197	6.46
QMagFace	492	1.135	0.5185	0.5185	1.0000	0.6829	126	117	9.53
AdaFace	1652	1.265	0.5926	0.6102	0.9931	0.7559	144	92	10.84

Table 8. Performance Comparison of Pipeline Configurations

Configuration	**GPU (MB)**	**RAM (GB)**	**Accuracy**	**Precision**	**Recall**	**F1 Score**	**Time (s)**
Complete Pipeline	2054	1.767	0.8750	0.9333	0.9333	0.9333	365.76
Detection Frame Interpolation	2054	1.674	0.6316	0.7500	0.8000	0.7742	43.27
Tracking ID merge	2054	1.658	0.6111	0.7857	0.7333	0.7586	43.33

Team Member	**Responsibilities**
Manan Aggarwal	Detection module (DAI, HLA), AdaFace, Pipeline Compilation, Tracking ID Merge
Shobhit Raj	LowFaceNet, Analysis of results, ArcFace, Detection Frame Interpolation
Souparno Ghose	Pose Attention, QMagFace, UI development, Quantization, Tracking
Additional Contribution: Dataset and Video collection	

Table 9. Individual contributions to the project