# FACE RECOGNITION IN CROWD

Alok Yadav[1], $PriyanshuMaurya^2$
Department of Computer Science and Engineering
Indian Institute of Information Technology Vadodara - ICD
Email: 202311007@diu.iiitvadodara.ac.in[1], 202311067@$diu.iiitvadodara.ac.in^2$

***Abstract*—Automated Face Recognition is one of the widely used systems in a security and surveillance that can identify an individual in a crowded place. Monitoring in a crowded environment manually is a very tedious and time taking process and human intervention makes it more prone to error. Therefore for these types of cases the model recognizes peoples faces and sends information(i.e camera number, time, confidence of detection) to the security authority. The paper uses a deep neural network method to recognize faces in a crowded environment where input is live video stream captured from CCTV camera, once the video is captured, the video is splitted into frames, and each frame is fed into model. The lightweight and fast fine tuned YOLOv8 model is used for face detection which accurately detects faces present in frames and a green bounding box is generated if a person's face matches with the face in a database. The similarity metrics are used to calculate similarity value and compared to a threshold value, and if it passes, then it is considered as matched otherwise labeled as unknown.**

***Index Terms*—Facenet512, YOLOv8, Similarity Matrices, Facial Recognition, Deep Learning**

## I. Introduction

In today's word, it is very difficult to surveillance using the older methods, because of the huge crowds present in every corner of the world. To address these types of issues we need a system the accurately detect and recognize individuals in real time. We have developed a real time surveillance monitoring system which will work well even in dense crowd with maintaining the system's security. [1]

The face recognition process starts with the data collection phase, where a dataset contains facial images of various individuals, both normal and occluded faces. To ensure robustness in our model our dataset consists of more than 150 images of each person to include variety angles and different occlusion pattern. This large variety of images helps the model learn individual facial characteristics more accurately and make this model more efficient to conventional face recognition methods. [2]

Our model uses a deep learning technique specifically YOLOv8, trained on WIDERFACE dataset which includes 398,733 faces from 32,203 images. This dataset is widely used for face detection tasks and enables the model to accurately locate faces even under varying lighting environments such as occlusions, different poses. [3]

The proposed model uses FaceNet, which is a deep learning based model that matches faces by creating embeddings of a given face.After the faces are located in the detection stage, they are passed in FaceNet512 to extract these embeddings. Based on the confidence score of the predictions, the system selects the result with high confidence score to increase the overall accuracy of the system. [4]

The model captures live footage from a CCTV camera and extracts the frames at regular intervals of one frame per second. From the captured frames, the YOLO model detects the faces in each frame, and face matching is done by comparing face embeddings created by FaceNet model of detected faces. The system generates an alert message if the processed face matches existing faces in the database.

## II. Related Work

In our study, we trained YOLOv8 on the WIDERFACE dataset to improve its adaptability to real-world surveillance conditions. During the training phase, we noticed that the application of random brightness and rotation augmentations significantly improved detection in dim lighting. Unlike MTCNN, which failed under such conditions, our YOLOv8 model maintain stable performance without the need for heavy GPU resources like RetinaFace.

In the domain of face detection, two other best models are **MTCNN** [5] and **RetinaFace** [6]. MTCNN is the most widely used because it demands less computational resources and produces satisfactory results but not the best compared to RetinaFace. It performs poorly in low-light conditions, but struggle to detect side-profile faces and works slower compared to YOLO. However, RetinaFace achieves higher accuracy compared to MTCNN and is little more than YOLO but much more computationally heavy than other models. All these challenges motivated us to train our self model, which is more capable of detect faces under low-light conditions, side-profile faces and takes less computation power. [2]

## III. Methodology

This paper focuses on presenting an approach for detecting faces from facial images. The methodology is divided into two primary stages:

1) **Face Detection:** Once the cctv is connected to the system the model captures frame form live stream and these frames are used to identify the positions of facial features. [2]
2) **Face Recognition:** Once the frame is captures the model begins to recognize the detected faces by comparing them with an existing database that contains facial images of multiple individuals. [2]

### A. Proposed System Architecture

The overall architecture of our face recognition system, as illustrated in Fig. 1, uses a two-stage pipeline that combines YOLOv8 for face detection and FaceNet512 for face recognition. This integrated approach enables real-time face identification in densely crowded environments.

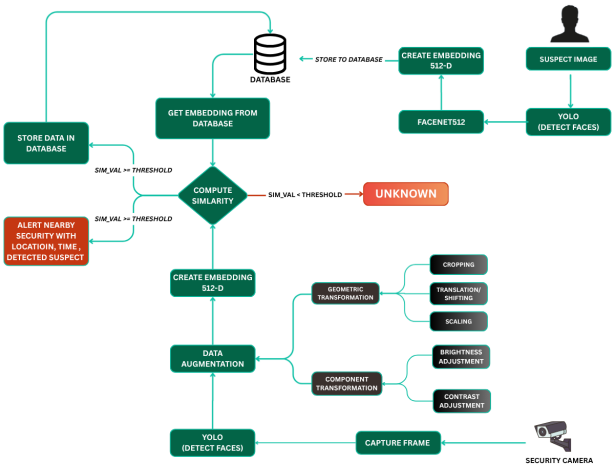


Fig. 1. Model Architecture

As shown in the diagram, the system processes input video streams through the following stages:

1) **Frame Extraction:** Video input is taken at 1 frame per second
2) **Face Detection:** YOLOv8 identifies and localizes faces in each frame
3) **Face Processing:** Detected faces are cropped, aligned, and resized
4) **Embedding Generation:** FaceNet512 converts faces to 512-dimensional embeddings
5) **Similarity Matching:** Cosine similarity used to compares embeddings
6) **Decision & Alert:** Matches trigger alerts to security authorities

### B. Deep Learning methods for Face Detection

*1) MTCNN:* Multi-Task Cascaded Convolutional Neural Network is a popularly used deep learning-based face detection model which is highly accurate and efficient. The model uses three stage pipeline containing of a P-NET, R-NET, and O-NET where each layer gradually refines the results. MTCNN also predicts the five facial keypoints such as eyes, nose and mouth, which boosts alignment accuracy and recognition performance. The main drawback of MTCNN is not to handle variations in pose, lighting, and partial occlusion which does not makes it suitable for real time systems. [7]

*2) RetinaFace:* RetinaFace is a single-stage face detector that performs joint face detection and facial landmark localization, based on a RetinaNet architecture, and uses a feature pyramid network(FPN) to detect faces of various scales effectively. RetinaFace introduces pixel-wise face localization supervision to improve bounding box precision and employs multi-task learning to predict five facial landmarks (eyes, nose, and mouth corners). [6]

*3) YOLOv8:* We have used YOLOv8, an highly accurate object detection model, trained on a WiderFace dataset for accurately detecting multiple faces per frame. The detected faces are cropped and then passed for facial recognition step. To avoid redundant cropping of the same face across video frames, we employed a nearest neighbor approach for efficient object tracking, ensuring seamless continuity. [3], [8]

### C. Facial Recognition

Face recognition works by comparing newly detected faces with the faces existing faces in a database. To make this process more accurate and reliable for real time system a stronger model is needed which compare faces with minimal error and at a high accuracy. FaceNet is one of the best deep learning model that improves recognition by converting each face into a compact numerical vector called an embedding, instead of using older methods like Eigenfaces or Fisherfaces. These high dimensional embeddings are then compared using the methods like cosine similarity and Euclidean distance, which measures how close two faces are based on their numerical representations. [2], [4] The shown Fig. 2 shows the model architecture of FaceNet. [4]

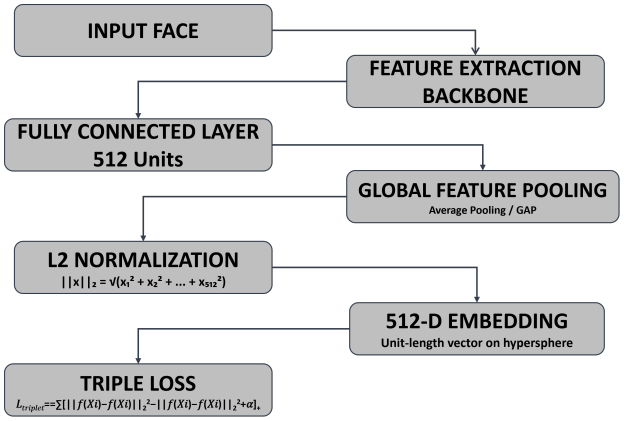


Fig. 2. FaceNet workflow diagram

*1) Triplet-Loss Function:* Facenet uses triplet loss function to separate the positive pair from the negative pair, the positive and the negative pair are refers to the faces of same and the different persons respectively. [2]

$$L_{\text{triplet}} = \sum_{i=1}^{N} \left[ \|f(X_i^A) - f(X_i^P)\|_2^2 - \|f(X_i^A) - f(X_i^N)\|_2^2 + \alpha \right]_+ \tag{1}$$

[2]

*2) Embedding Similarity Measurement:* We have used cosine similarity [9]instead of the Euclidean distance to measure the similarity between embeddings:

$$\text{cosine\_similarity}(A, B) = \frac{f(A) \cdot f(B)}{\|f(A)\| \, \|f(B)\|} \tag{2}$$

If CosineSimilarity $>=$ threshold, then person $A$ and person $B$ are more likely to be the same person. [2]

### D. Data collection and processing

The training dataset used in this project is WIDERFACE dataset which is a collection of over 32,000 images which contains approx 400,000 faces.The images in dataset are taken from different places and in different condition, pose, and with different scaling.Widerface dataset is organised on 61 differnt event classes.In each class, we randomly take 40%10%50% for training, validation, and testing set for the model. [10]

The testing dataset includes 14 classes, in which each containing around 150 images of various cricketers and Bollywood Celebrities taken from internet and organized in subdirectories by person.The images captured under different conditions and angels, frontal, left, right, and distant views. Prior to extraction, the images are resized and preprocessed to ensure uniformity across different values.Then the frame is passed to the enhancement model which fixes the brightness of frame to improve extraction of faces. 3.

Fig. 3. Validation Dataset Images

*1) Face Preprocessing:* After the face is detected, the bounding box crops out the face from the image [11]. The cropped face is then resized to 640×640 pixels so that every input to the FaceNet512 model has the same size. This size is selected because it keeps the important facial details while still being fast to process.. [2]

*2) Feature generation process:* when a face detected in frame, the FaceNet512 model creates 512-dimension embeddings, that shows numerical representation of the facial features.Each dimension capture a specific characteristic learned by the model, such as eyes, nose, and jaw structure. The detailed representation helps the model to distinguish main differences between faces, maximizing recognition accuracy. [2].

The embedding vector simplifies the comparison process during recognition. If model is comparing each pixel from both images, it gives inefficient and prone to error approach, embeddings enables the use of efficient distance metrics (like Cosine Similarity, euclidean distance) to determine if two person matches. [2]

Storage and Retrieval: The generated embeddings are stored in a SQL database (SQLite for development, PostgreSQL for production) in JSON format for efficient storage and retrieval. This acts as a model to match the embeddings of face detected in frame and stored embeddings in database

Embeddings are saved in JSON format to minimize computation time, allowing rapid face retrieval and matching without the need for repeated embedding calculations.

*3) Model training:* The YOLO model is trained on the WIDER FACE dataset which includes 398,733 faces from 32,203 images. The dataset is divided into a training split of 80% and a testing split of 20%. The shuffle technique is used to randomize the selection of images dataset, ensuring that the order of samples does not impact how data is partitioned into training and testing subsets. [3]

TABLE I
YOLOv8 TRAINING PARAMETERS

| Dataset | WIDER FACE |
|---|---|
| Epochs | 50 |
| Batch size | 16 |
| Optimizer | Adam |
| Learning Rate | 0.001 |
| Hardware | T4 GPU |
| Training Time | 3 hours |

## IV. Results and Discussion

Our experimental results confirm that combining YOLOv8 for face detection and FaceNet for face recognition yields better performance. YOLOv8 exhibited excellent accuracy in detecting and separating faces even under challenging conditions involving changes in illumination, pose, and facial expression.Fig. 4.

| Person | Total | Detected | Total Rec. | Correct Rec. | Wrong Rec. | Not Rec. | Acc. (%) |
|---|---|---|---|---|---|---|---|
| sachin | 150 | 150 | 109 | 103 | 6 | 41 | 68.67 |
| chris_gayle | 119 | 113 | 108 | 103 | 5 | 5 | 91.15 |
| ms_dhoni | 148 | 142 | 113 | 104 | 9 | 29 | 73.24 |
| amitabh_bachchan | 150 | 150 | 138 | 131 | 7 | 12 | 87.33 |
| as_rai | 150 | 150 | 145 | 139 | 6 | 5 | 92.67 |
| RG_Sharma | 150 | 142 | 115 | 102 | 13 | 27 | 71.83 |
| srk | 150 | 150 | 131 | 128 | 3 | 19 | 85.33 |
| r_ashwin | 150 | 150 | 133 | 129 | 4 | 17 | 86.00 |
| kapil_dev | 150 | 150 | 124 | 113 | 11 | 26 | 75.33 |
| h_kaur | 150 | 150 | 112 | 106 | 6 | 38 | 70.67 |
| amir | 153 | 153 | 132 | 122 | 10 | 21 | 79.74 |
| AK | 150 | 150 | 135 | 131 | 4 | 15 | 87.33 |
| virat_kohli | 150 | 150 | 131 | 129 | 2 | 19 | 86.00 |
| salmankhan | 150 | 150 | 139 | 123 | 16 | 11 | 82.00 |

TABLE II
TESTING RESULT TABLE

$$T = \text{Total images} \tag{3}$$

$$D = \text{Faces detected} \tag{4}$$

$$C = \text{Correctly recognized faces} \tag{5}$$

$$W = \text{Wrongly recognized faces} \tag{6}$$

$$TotalRecognized = (C + W) \tag{7}$$

$$NotRecognized = D - (C + W) \tag{8}$$

$$Rec.Accuracy = \frac{C}{D} \times 100 \tag{9}$$

The YOLO model has achieved excellent accuracy of 98.6% on test dataset and the combined YOLO and FaceNet model has achieved high accuracy of 81.12% on the dataset contains images of cricketers and Bollywood celebrities captured from different angles and variant lighting conditions.

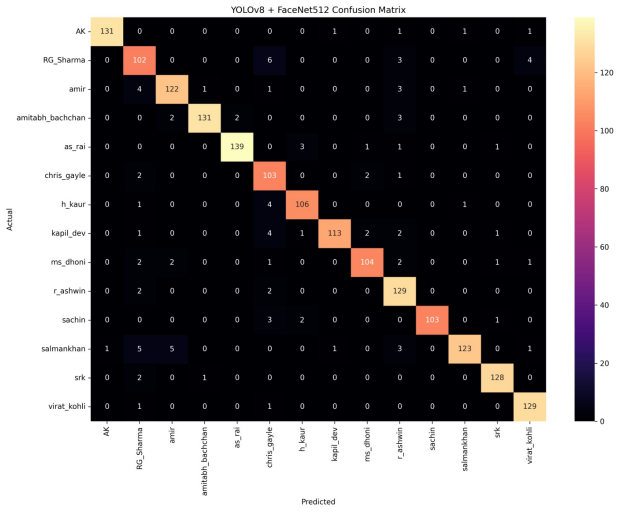


Fig. 4. Confusion Matrix

TABLE III
PERFORMANCE METRICS

| Metric | Value |
|---|---|
| Precision | 0.88 |
| Recall | 0.75 |
| F1-score | 0.81 |

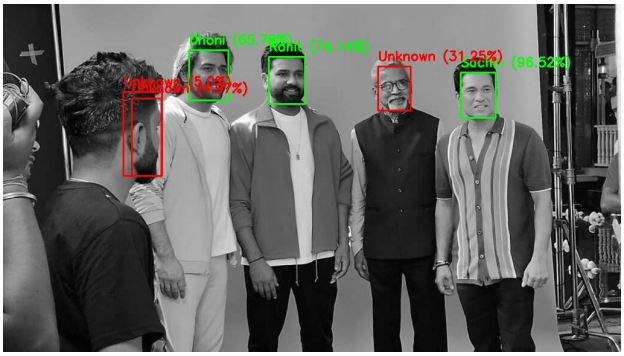


Fig. 5. Face recognition from an image

## V. CONCLUSION

In this work, an efficient face detection and recognition model has been implemented by with YOLOv8 for face detection and FaceNet512 for face recognition, provides high accuracy in face detection and recognition under varying lighting conditions and poses. It provides capability to give real time inferences which makes it well suited for surveillance and security applications.

This paper also compare different models for face recognition on the basis of their accuracy, robustness, computation time and power.

Overall, the proposed approach offers a strong balance between detection accuracy and computational efficiency which makes it perfect for real time face detection systems such as CCTV cameras.

## REFERENCES


[1] S. Tang, Z. Zhang, L. Lin, and K. He, "Pyramidbox: A context-assisted single shot face detector," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 812–828.

[2] S. Mummaneni, V. C. S. R. Mudunuri, S. V. V. Bommaganti, B. V. Kalle, N. Jacob, and E. S. R. Katari, "Face recognition in dense crowd using deep learning approaches with ip camera," *Informatyka, Automatyka, Pomiary w Gospodarce i Ochronie Środowiska*, vol. 15, no. 2, pp. 44–50, 2025.

[3] C. Zhang, X. Xu, Y. Fu, and S. Zhang, "Face detection using yolo-based deep learning model trained on wider face dataset," *IEEE Access*, vol. 8, pp. 158 807–158 817, 2020.

[4] A. Firmansyah, T. F. Kusumasari, and E. N. Alam, "Comparison of face recognition accuracy of ArcFace, FaceNet and FaceNet512 models on DeepFace framework," in *2023 International Conference on Computer Science, Information Technology and Engineering (ICCoSITE)*. IEEE, 2023.

[5] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," *IEEE Signal Processing Letters*, vol. 23, no. 10, pp. 1499–1503, 2016.

[6] J. Deng, J. Guo, E. Ververas, I. Kotsia, and S. Zafeiriou, "Retinaface: Single-shot multi-level face localisation in the wild," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 5202–5211.

[7] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," *IEEE Signal Processing Letters*, 2016.

[8] G. Jocher and et al., "Yolov8: Cutting-edge real-time object detection," Ultralytics, 2023, available: https://github.com/ultralytics/ultralytics.

[9] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "Deepface: Closing the gap to human-level performance in face verification," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1701–1708, 2014.

[10] S. Yang, P. Luo, C. C. Loy, and X. Tang, "Wider face: A face detection benchmark," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
[11] M. Wang and W. Deng, "Deep face recognition: A survey," *Neurocomputing*, vol. 429, pp. 215–244, 2021.