FaceTrace AI: A Deep Learning-Based Video Surveillance System for Automated Suspect Identification

Hamid Iqbal *, Haroon Arshad *, Muhammad Amaaz *, Asjad Amin *
*Department of Information and Communication Engineering, Faculty of Engineering and Technology,
The Islamia University of Bahawalpur, Pakistan

***Abstract*—As surveillance cameras are everywhere, there's a strong need for systems that can automatically spot and track suspects without constant human monitoring. Looking at CCTV manually is slow and prone to error. In this paper, we introduce FaceTrace AI that works in two modes. SCRFD is used for face detection, and ArcFace ResNet-50 is adopted to generate 512-dimensional embeddings. Known suspects are recognized by a linear SVM trained on L2-normalized embeddings, and unknown suspects are detected using cosine similarity with user-provided images.**

Model is tested based on 619 images from three subjects with 1 FPS thresholds of 0.68 and 0.55. Powered by the Dockerized FastAPI backend, FaceTrace AI records important attributes and generates a PDF with timestamping, confidence scores, and annotated frames—achieving precision at commercial scale.

***Index Terms*—Face Recognition, Video Surveillance, Deep Learning, ArcFace, Support Vector Machine, SCRFD, Automated Suspect Detection, Computer Vision, Cloud Deployment**

I. Introduction

Large-scale deployment of surveillance cameras results in huge amount of video data, manual CCTV monitoring is quite time-consuming and error-prone that require heavy human resources and postpone the real-time responses and post-event suspects identification [1] [2] [3].

Face recognition has greatly evolved with the advent of deep learning beyond Eigenfaces and LBP across pose and lightening [4] [5]. CNN-based models such as DeepFace and FaceNet can learn discriminative embeddings, but ArcFace enhance discrimination, and detectors such as MTCNN and SCRFD enable accurate real-time detection [6] [7] [8].

But the existing monitoring systems are inadequate. Commercial solutions cover only database matching, whereas academic similarity-based softwares do not provide persistent monitoring capabilities [9]. This makes agencies to depend on different tools, which complicates the workflow. High computational cost limits deployment for institutions lacking GPU-based computing facilities [10] [11] [12].

In this paper we present FaceTrace AI, an integrated surveillance system that provides database identification as well as search for unknown suspects with similarity comparisons. With SCRFD detection, ArcFace embeddings, and SVM classification the system achieves accurate CPU-only processing that can be used to analyze video automatically generating cloud-accessible forensic reports containing timestamps, confidence scores and marked frames [13].

II. System Architecture

FaceTrace AI features a three-stage design, which is training phase, deployment stage, and runtime inference stage-specifically designed for accuracy and efficiency.

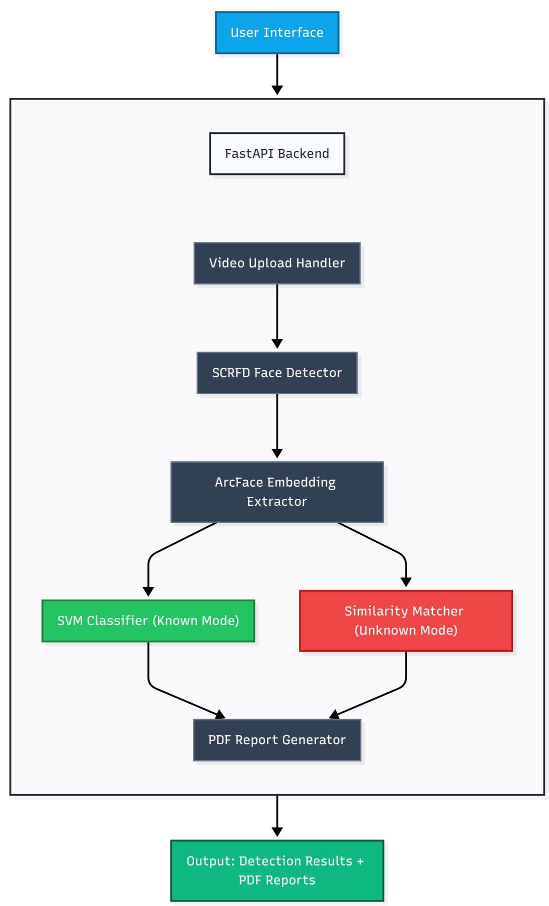

Fig. 1. Overview of the FastAPI-driven processing pipeline integrating face detection, ArcFace embeddings, known/unknown suspect identification, and PDF report generation.

The framework contains five main modules: the web frontend, the FastAPI backend, face detection, embedding and

DOI: https://doi.org/10.67535/tsp.000002.027
Published under the ISAISS 2026 Proceedings

classification engine, and a reports generator. Its modular design allows decentralized development and seamless end-to-end operation.

The training is performed off-line and operates on a face database that includes the whole image acquisition data set to generate models for identification. Face images are detected and aligned by SCRFD, and ArcFace ResNet-50 is employed to encode them with 512-dimensional unit embeddings. These vectors are used to train a linear SVM classifier, which captures identity boundaries; the trained SVM model, label encoder and training metadata (including recommended confidence thresholds) are saved for deployment. The deployment phase combine all built process into a Docker with the FastAPI server, SCRFD and ArcFace model from InsightFace and custom SVM classifier. We use containerization for reproducible execution and the cloud deployment so that GPU not required, but practical process time is still viable. The frontend hosted separately is communicating with the backend using REST endpoints to upload video, processing and get back reports.

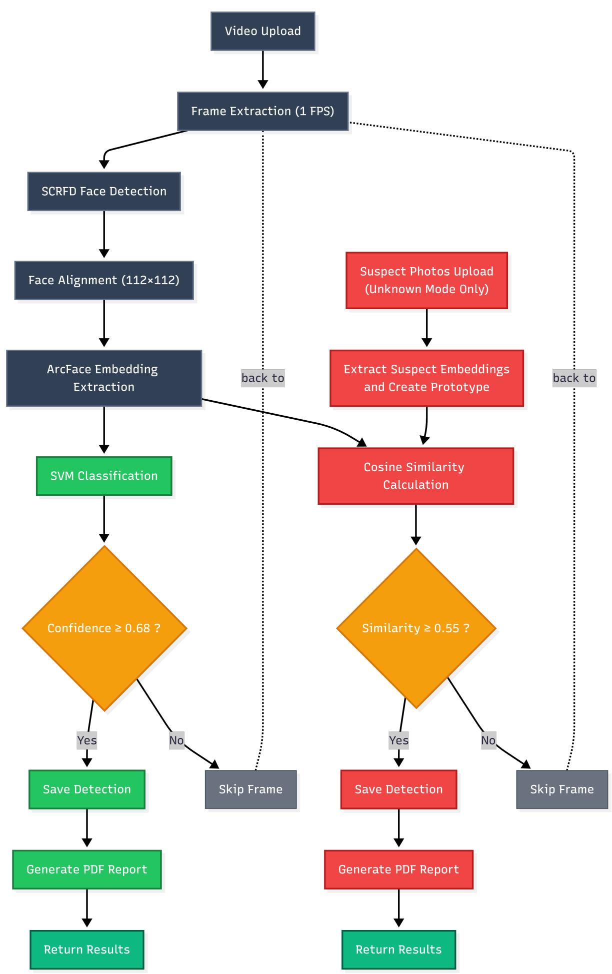

Fig. 2. Dual-mode workflow of FaceTrace AI using SVM for known-mode identification and cosine similarity for unknown-mode matching.

There are two ways of matching in runtime inference, which are based on the same detection and embedding pipeline. In known mode, uploaded surveillance footage is taken every second to trade off processing speed with coverage. Every frame is sent to SCRFD for face detection, and bounding boxes and landmarks are generated for alignment. The 112×112 cropped faces, normalization, and embeddings are produced with ArcFace. The SVM classifier would apply probability scores, and detections with confidence above 0.68 are predicted and below are considered as unknown.

For unknown mode, we use the same detection pipeline but compare with a suspect prototype computed by averaging multiple user-provided reference embeddings. Cosine similarity is used to calculate matches, and a 0.55 cutoff was selected as optimal between sensitivity and specificity for investigations.

The report generation subsystem is ReportLab to generate PDF files with timestamps, confidence score, bounding boxes, identity labels and embedded detection frames (up to six, depending on the file size). Reports are placed in a managed directory with automatic disposal which retains the most recent 100 files to ensured disk usage is not abused on restricted cloud storage.

Overall, practical visualization is heavily pursued in the proposed solution as follows: stateless API design for scalable operation, efficient processing supported by periodic frame sampling, bounded resources with control of their use, and user-friendly forensic reports that are friendly to security environments.

A. *Maintaining the Integrity of the Specifications*

The IEEEtran class file is used to format your paper and style the text. All margins, column widths, line spaces, and text fonts are prescribed; please do not alter them. You may note peculiarities. For example, the head margin measures proportionately more than is customary. This measurement and others are deliberate, using specifications that anticipate your paper as one part of the entire proceedings, and not as an independent document. Please do not revise any of the current designations.

III. DATASET COLLECTION AND PREPROCESSING

We created a custom dataset of 3 people under various surveillance conditions in terms of lighting, pose, expression and distance from camera. Photographs were taken with handheld smartphone cameras during several indoor and outdoor sessions over a period of weeks, adding temporal and environmental variation.

The original dataset of 1,005 images (Hamid: 335, H.Arshad: 340 and M. Amaaz :330) was manually quality filtered to a collection of high-quality images (619 samples with overall retention rate 61.6 %; Hamid Iqbal:66.9%, Haroon Arshad:56.2%, Amaaz :61.8%). We manually discarded multi-face, severe motion blur, pose angle more than 45 degrees for at least one axis of relative rotation, occluded or low-light-images that could lead to noise in training.

The pre-processing pipeline uses the SCRFD face detection to detect faces. The detected faces are aligned with respect to five facial landmarks (eyes, nose, mouth corners) by affine transformation, generating 112×112 pixels ready for ArcFace embedding extraction. This normalization standardizes the shape of all the samples, independently of their acquisition conditions.

A balanced data set of 619 images was used from a cross-validation study (191-224 samples per subject). We use a stratified splitting of 70/20/10: 433 images for training, 124 for validation and 62 for testing. Stratification guarantees a proportional distribution of subjects for each split.

IV. METHODOLOGY

FaceTrace AI adopts a pipeline that consists of several components with face localization, feature extraction, and identity classification or similarity matching to process the raw surveillance video frames into detected identities. We process videos with duration of 1 second of length (1FPS) in order to find a tradeoff between computational efficiency and temporal coverage, sampling 3.3% of frames at standard 30 FPS while preserving enough detection chances.

A. Face Detection Using SCRFD

SCRFD (Sample and Computation Redistribution for Face Detection) is the first stage, which detects faces including their geometric information to align. We set SCRFD to 640×640 pixel input resolution and 0.50 confidence threshold for inference. The detector outputs bounding boxes with coordinates (x, y, width, height), confidence scores and five facial landmarks (eye centers, nose tip and mouth corners). Frames in which no faces are detected are discarded, and frames with multiple faces have each face processed through the pipeline.

B. Face Alignment and Feature Extraction

The five SCRFD points provide a geometrically normalized face with an affine transformation and are used to make faces aligned to canonical upright position. Aligned regions are then zoomed to 112×112 pixels using bilinear interpolation after transformed from BGR (OpenCV default color space) to RGB, following ArcFace detector model specifications.

The embedding vectors are then compressed into 512 dimensions from the ArcFace model with ResNet-50 backbone. ArcFace uses additive angular margin loss and the representations are adjusted to have a cosine similarity that preserves facial similarity on the unit hypersphere. We normalize the embeddings with L2 norm to project them on to unit length, so that vectors are comparable for classification and similarity matching.

C. Known Mode: SVM Classification

For known suspect detection, we have our model a linear Support Vector Machine trained on the normalized embeddings of the training set. The linear kernel is appropriate for high-dimensional spaces with deep learning features that are approximately linear separable. We use the SVM with probability calibration and C=1.0, imposing balanced margin-error trade-off.

At inference time, for each embedding, we form a probability distribution over the three trained identities. We take the largest probability as our confidence. Faces with confidence score above 0.68 are then accepted as positive identification, and labeled by bounding box in green indicating person's name as well as the reported confidence score. Less confident computations are rejected as unknowns. This threshold was determined using a validation experiment to reduce false positives, while retaining detection sensitivity.

D. Unknown Mode: Similarity Matching

In unknown mode, similarity-based matching without the SVM classier is used. Users submit 1-10 reference images of a suspect. The system then takes the embeddings of all reference images and computes an element-wise mean, forming a prototype embedding for the average face. A prototype is L2 normalized and used as a matching reference. While processing videos, embeddings of each detected face are compared to the prototype suspect via cosine similarity (dot product between normalized vectors). This score ranges between 1 and +1 where the value of 1 indicates maximum similarity. A threshold of 0.55 for suspect matching is the value we use. Faces over this threshold are considered a match and are given a red bounding box marking them as "SUSPECT" with their similarity score, to differentiate how they look from known mode detections. The detection results consist of time stamps, the bounding boxes, the identity tags and the confidence/similarity score.

V. RESULTS AND EVALUATION

We compare FaceTrace AI in terms of both the classification accuracy, class-wise performance and computation efficiency. All experiments are based on the 70/20/10 stratified split leaving 62 images for test.

TABLE I
OVERALL MODEL PERFORMANCE

Metric	Value
Training Accuracy	100.00%
Validation Accuracy	100.00%
Test Accuracy	100.00%
Test Precision	1.0000
Test Recall	1.0000
Test F1-Score	1.0000
5-Fold CV	100.00% ± 0.00%
Confidence (Mean)	0.987

A. Classification Performance

The linear SVM classifier trained on ArcFace embeddings obtain 100% accuracy on the test set which has 62 images of three subjects (Hamid Iqbal:22, Haroon Arshad:20, Muhammad Amaaz:20).

The confusion matrix of Figure 3 shows perfect classification (0 misclassification on all the identity pairs), thus showing that the 512-dimensional ArcFace feature space has full separability between three subjects despite using a moderate dataset size.

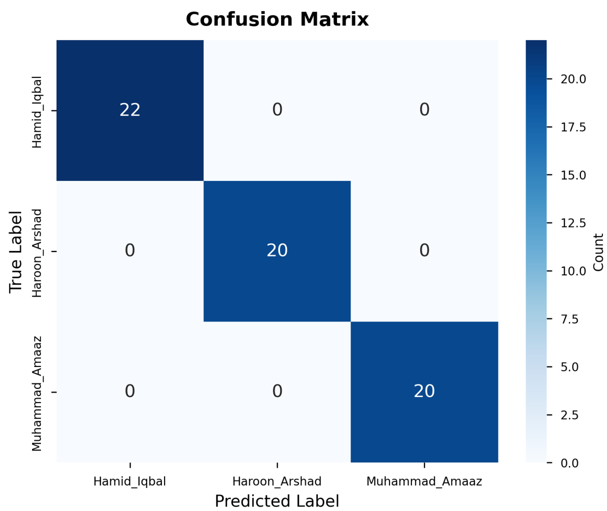

Fig. 3. Confusion Matrix

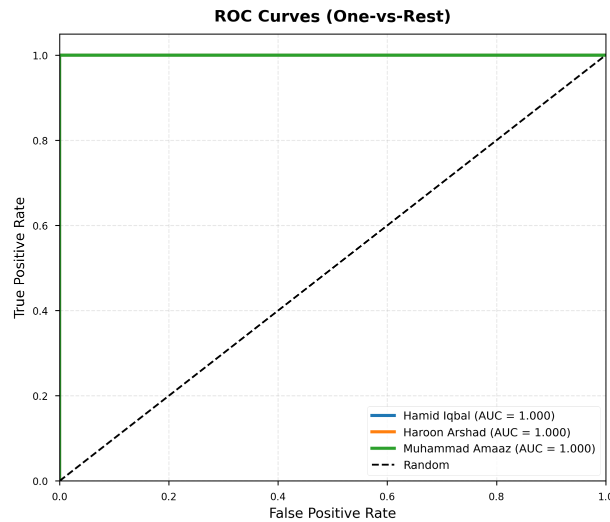

Fig. 5. ROC Curve Analysis

B. Per-Class Performance Metrics

From per-class inspection, we can see that the performance is well balanced for all subjects with a value of 1.000 for precision, recall and F1-score for each individual as demonstrated by Figure 4. The homogeneous metrics suggest there is no classifier bias towards specific identities and that the training performed with only 433 samples has learned discriminative boundary. Perfect recall shows that the confidence threshold of 0.68 accepts all true positives and results in zero false positives in the test set.

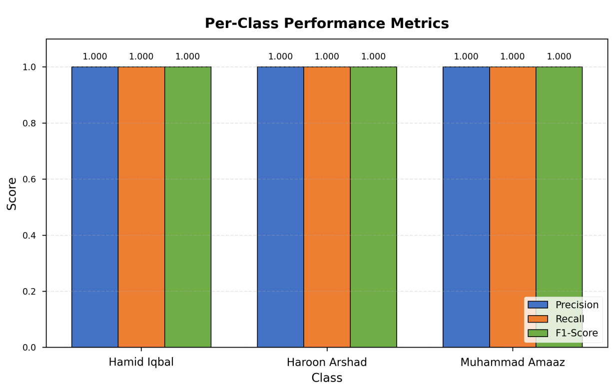

Fig. 4. Per-Class Performance Metrics

C. ROC Curve Analysis

Figure 5 shows the ROC curves for one-vs-rest classification, with AUC=1.000 achieved in all classes. The curves ascend to the top left-hand corner with a downstream inflection involving 100% true positive rate at zero false positive rate, reflecting perfect separation in the decision function space. This result provides the evidence to support the combination of ArcFace embeddings and linear SVM classification in small scale face recognition.

VI. Discussion

The results indicate that FaceTrace AI attains a 100% classification accuracy and remains practical to deploy for surveillance use. And, dual-mode design combines the system-level data discovery and similarity-based search into one, thus avoiding discrete systems so as to lower operation complexity. Using pre-trained embeddings in ArcFace principle allows for strong performance even with relatively low number of samples (433 training samples distributed over three subjects) which is a fraction of the thousands of images per- person needed to train from scratch. This in turn proves that even a CPU-only deployment to the cloud can be accessible without costly GPU hardware, making it available for small enterprises and universities.

VII. Limitations

Several constraints warrant acknowledgment. This three-individual dataset is a proof-of-concept, however large-scale application would need to include tens or hundreds of subjects in order to demonstrate scalability. Using solely CPU, although available for anyone to test out, narrow down the system so it can only process through recorded clips in a batch-type fashion but not live streaming. If processing is GPU-accelerated, per-frame computation time could be reduced by an order of magnitude (potentially 5-10× faster), facilitating live video monitoring with instantaneous alerting. Sampling at 1 frame per second (FPS) presents a theoretical risk of missing brief appearances, but most surveillance applications are based on several-second long subjects. Curated datasets in a relatively controlled environment may not be able to reflect difficult CCTV imaging conditions such as severe motion blur, extreme low-light conditions, very poor resolution and substantial facial partial occlusion.

VIII. Future Work

The dataset will be revised to prioritize improvements such as the inclusion of additional participants with different demographics, in order to test scalability and remove bias. If we added GPU acceleration, then using the software with any live video stream would be possible and real time. Multi-camera tracking could track persons across facility networks, whilst compatibility with the existing security management system would allow for efficient operational flows. Other attributes like age estimation, gender classification or emotion recognition could offer richer context to identity information. Completion of the webcam detection capability that is currently in development would provide investigators with interactive tools to quickly review and validate footage.

IX. Conclusion

The paper introduced FaceTrace AI, a novel hybrid video surveillance system with two modes of operation: detect and identify, featuring SCRFD face detection, ArcFace ResNet-50 embeddings and linear SVM based classification for automated suspect identification. The system supports both the matching of individuals against a database of pre-registered people (confidence threshold = 0.68) and the identity-based detection unknown suspects from reference images (similarity threshold = 0.55). Practical accessibility (without dedicated hardware) of cloud deployment via dockerised FastAPI. Testing on 619 face images from three subjects returned the perfect precision, recall and F1-score, indicating that for small-scale face recognition tasks our approach is both effective and robust. The system produces forensic reports including timestamps, confidence scores and annotated frames for use in an investigative report. Major contributions includes a unified architecture of both modes without the need for two sets of surveillance softwares, good performance on small datasets such as those used in organization security and deployment using CPU-based cloud with easy access and automatic evidence generation. FaceTrace AI eases burden of manual surveillance and speeds suspect identification in the context of security operations and law enforcement investigations.

References

[1] V. Opanasenko, S. K. Fazilov, S. Radjabov, and S. S. Kakharov, "Multilevel face recognition system," *Cybernetics and Systems Analysis*, vol. 60, no. 1, pp. 146–151, 2024.

[2] E. L. Piza, "Cctv video surveillance and crime control," *The Oxford handbook of evidence-based crime and justice policy*, p. 265, 2024.

[3] C. Ryando, R. Sigit, B. S. B. Dewantara *et al.*, "Comparison of machine learning algorithms for face classification using facenet embeddings," *The Indonesian Journal of Computer Science*, vol. 13, no. 4, 2024.

[4] M. Kim, A. K. Jain, and X. Liu, "Adaface: Quality adaptive margin for face recognition," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 18 750–18 759.

[5] S. Serengil and A. Özpınar, "A benchmark of facial recognition pipelines and co-usability performances of modules," *Bilişim Teknolojileri Dergisi*, vol. 17, no. 2, pp. 95–107, 2024.

[6] 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, pp. 535–539.

[7] J. K. Iype and S. Sebastian, "Comparative analysis of state-of-the-art face recognition models: Facenet, arcface, and openface using image classification metrics," in *International Conference on Computational Sciences and Sustainable Technologies*. Springer, 2023, pp. 222–234.

[8] B. S. Bhargava, K. U. T. Reddy, V. Srihari, and K. V. Sharma, "Integration of facenet and deepface algorithms for optimized attendance management in educational systems," *Macaw International Journal of Advanced Research in Computer Science and Engineering*, vol. 9, no. 1, pp. 1–11, 2023.

[9] J. Chi, C. K. On, H. Zhang, and S. S. Chai, "A review of deep convolutional neural networks in mobile face recognition." *International Journal of Interactive Mobile Technologies*, vol. 17, no. 23, 2023.

[10] X. Wang, "Design and implementation of real-time face recognition system using deep learning," in *2024 IEEE 6th International Conference on Civil Aviation Safety and Information Technology (ICCASIT)*. IEEE, 2024, pp. 65–67.

[11] Z.-Y. Deng, H.-H. Chiang, L.-W. Kang, and H.-C. Li, "A lightweight deep learning model for real-time face recognition," *IET Image Processing*, vol. 17, no. 13, pp. 3869–3883, 2023.

[12] B. Nath and P. Mahanta, "An analysis of various face recognition models available in opencv and deepface," *Available at SSRN 5531966*.

[13] P. Thanishka and P. Avinash, "Backend optimization and face recognition in real-time surveillance using deepface," in *International Conference on Computational Intelligence in Data Science*. Springer, 2025, pp. 389–403.