© JUL 2026 | IRE Journals | Volume 10 Issue 1 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV10I1-1719977

FaceTrace: AI-Powered Real-Time Facial Recognition System for Integrated Home and Public Safety

KIRAN KRISHNAN V[1], FRIJO ANTONY C F[2], SANDRA ROSE JOSEPH[3], SHARUN K S[4], DR. JOYCY K ANTONY[5]

[1,2,3,4]Department of Computer Science and Engineering Vidya Academy of Science & Technology Thrissur, Kerala, India

[5]Associate Professor, Department of Computer Science and Engineering Vidya Academy of Science & Technology Thrissur, Kerala, India

Abstract-** The increasing demand for intelligent security sys-tems has highlighted the limitations of traditional surveillance solutions, which primarily rely on passive monitoring and lack real-time decision-making capabilities. In this paper, we present FaceTrace, an AI-powered real-time facial recognition system designed to enhance both residential and public safety through automated detection, classification, and alert generation. The proposed system integrates deep learning-based face detection using Multi-Task Cascaded Convolutional Neural Net-works (MTCNN) with feature extraction through the FaceNet architecture (InceptionResnetV1). Each detected face is trans-formed into a compact 128-dimensional embedding, enabling efficient and scalable identity recognition using distance-based matching. Unlike conventional classification-based models, the system employs an embedding-based incremental learning ap-proach, allowing new individuals to be added dynamically with-out retraining the model. FaceTrace follows a multi-module architecture comprising user, admin, and police station components. The system classifies individuals into four categories: familiar persons, unknown individuals, criminals, and missing persons. Based on the clas-sification, context-aware alerts are generated and categorized into normal, warning, and danger levels. A key feature of the system is its location-based alert routing mechanism, which automatically notifies the nearest police station in the event of high-risk detections. The system is implemented using a web-based interface for real-time video monitoring and a mobile application for user interaction, including profile management, alert history tracking, and complaint submission. Experimental evaluation demonstrates that the system achieves an overall accuracy of approximately 86% with low processing latency (approximately 1.2 seconds per frame) and efficient alert delivery (within 2 seconds). The confusion matrix analysis shows strong classification performance with minimal cross-category misclassification, ensuring reliable operation in practical scenarios. Overall, FaceTrace provides a scalable, efficient, and deploy-able solution that bridges the gap between personal security systems and public safety infrastructure. The integration of real-time facial recognition with intelligent alert mechanisms and law enforcement coordination makes the system a promising approach for next-generation smart surveillance applications.

Index Terms—Facial Recognition, Deep Learning, FaceNet, MTCNN, Smart Surveillance, Real-Time Systems

I. INTRODUCTION

In recent years, the rapid growth of urbanization and popula-tion density has led to an increase in security concerns in both residential and public environments. Traditional surveillance systems, such as closed-circuit television (CCTV), are widely deployed for monitoring purposes; however, they primarily rely on passive observation and human intervention. These sys-tems lack the capability to automatically identify individuals, classify threats, or provide intelligent responses in real-time.

With the advancement of deep learning and computer vision, facial recognition has emerged as a powerful tool for intelli-gent surveillance systems. Modern face recognition models are capable of extracting highly discriminative features from facial images, enabling accurate identification under varying environmental conditions such as changes in lighting, pose, and expression [1]. In particular, embedding-based approaches, such as FaceNet, map facial images into a compact feature space where similarity between individuals can be efficiently measured.

© JUL 2026 | IRE Journals | Volume 10 Issue 1 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV10I1-1719977

Despite these advancements, most existing facial recog-nition systems operate as standalone solutions and do not integrate with broader security infrastructures. They often lack real-time alert mechanisms, user-specific customization, and coordination with law enforcement agencies. Additionally, many systems require large datasets and frequent retraining, making them less practical for dynamic real-world environ-ments where new individuals need to be added continuously. To address these challenges, this paper presents FaceTrace, an AI-powered real-time facial recognition system designed to enhance both home security and public safety. The pro-posed system integrates deep learning-based face detection and recognition with a multi-module architecture consisting of user, admin, and police station components. The system is capable of detecting and classifying individuals into four categories: familiar persons, unknown individuals, criminals, and missing persons.

A key feature of FaceTrace is its embedding-based incre-mental learning approach, which allows new individuals to be added without retraining the entire model. This significantly improves scalability and adaptability. Furthermore, the system incorporates a context-aware alert mechanism that categorizes alerts into normal, warning, and danger levels. In high-risk scenarios, such as the detection of criminals or missing per-sons, alerts are automatically forwarded to the nearest police station using location-based routing.

The main contributions of this work are summarized as follows:

- Design and implementation of a real-time facial recogni-tion system using MTCNN and FaceNet.
- Development of a scalable embedding-based recognition framework with incremental learning capability.
- Integration of a multi-role architecture involving users, administrators, and police stations.
- Implementation of a real-time alert system with threat classification and location-based routing.
- Development of both web and mobile interfaces for monitoring, alert visualization, and user interaction.

The remainder of this paper is organized as follows: Section
II discusses related work in facial recognition and smart surveillance systems. Section III presents the system archi-tecture. Section IV explains the methodology and recognition pipeline. Section V describes the system workflow and alert mechanism. Section VI provides evaluation and performance analysis. Finally, Sections VII and VIII conclude the paper and discuss future enhancements.

II. RELATED WORK

Face recognition has been an active research area in com-puter vision for several decades. Early approaches relied on handcrafted feature extraction techniques such as Eigenfaces, Local Binary Patterns (LBP), and Scale-Invariant Feature Transform (SIFT) [9]. Classical object detection methods such as the Viola-Jones algorithm [10] were widely used for real-time face detection due to their computational efficiency. However, these approaches were highly sensitive to variations in lighting, pose, and occlusions, limiting their performance in real-world scenarios.

With the emergence of deep learning, significant improve-ments have been achieved in facial recognition systems. Con-volutional Neural Networks (CNNs) have enabled automatic feature learning directly from raw images, eliminating the need for manual feature engineering. The FaceNet model introduced by Schroff et al. [1] represents a major advancement by mapping facial images into a unified embedding space, where similarity between faces is measured using distance metrics. This approach has demonstrated high accuracy and scalability across large datasets.

For face detection and alignment, the Multi-task Cascaded Convolutional Neural Network (MTCNN) proposed by Zhang et al. [2] has become a widely adopted method. MTCNN uti-lizes a three-stage cascaded architecture consisting of Proposal Network

(P-Net), Refinement Network (R-Net), and Output Network (O-Net) to accurately detect faces and extract facial landmarks. This enables robust performance under varying lighting conditions and facial orientations.

Foundational works in deep learning [3] and computer vi-sion [4] have provided the theoretical and algorithmic basis for modern recognition systems, including feature representation, optimization techniques, and image processing methodologies. In recent years, several applications of facial recognition have been explored in the domain of smart home security and surveillance. Ansari et al. [5] proposed a smart home safety system using machine learning techniques, demonstrating improved security through automated recognition. Similarly, Satari et al. [6] developed a visitor monitoring system using facial recognition to manage and track individuals entering restricted areas.

Comprehensive surveys such as Zhao et al. [7] have ana-lyzed the evolution of face recognition techniques, highlighting challenges such as dataset dependency, environmental sensi-tivity, and scalability issues. Performance evaluation methods, including ROC analysis [8], have also been widely used to assess classification effectiveness in recognition systems.

Despite these advancements, existing systems exhibit sev-eral limitations. Most solutions operate as standalone systems without integration into a larger security framework. They lack real-time alert classification, user-specific customization, and coordination with law enforcement agencies. Additionally, many systems rely on traditional classification-based models that require frequent retraining, making them less suitable for dynamic environments.

The proposed FaceTrace system addresses these limitations by introducing an integrated architecture that combines real-time facial recognition with a multi-module framework involv-ing users, administrators, and police stations. The system lever-ages embedding-based recognition with incremental learning, enabling efficient scalability without retraining. Furthermore, it incorporates a context-aware alert mechanism with location-based routing, ensuring timely response in high-risk scenarios.

III. SYSTEM ARCHITECTURE

The FaceTrace system follows a modular and scalable architecture designed to support real-time facial recognition, alert generation, and multi-user interaction. The overall archi-tecture, as shown in Fig. 1, consists of a Smart Camera input module, a central AI processing engine built using Python and Django, a MySQL database for data storage, and three primary output modules: Admin Dashboard, Police Station Module, and User Mobile Application.

A. Smart Camera (Input Layer)

The system begins with a smart camera that continuously captures video frames. In the current implementation, this is simulated using a webcam through the user web application. The captured frames serve as real-time input for the face detection and recognition pipeline.

The camera module ensures continuous monitoring and provides image streams to the processing unit with minimal latency.

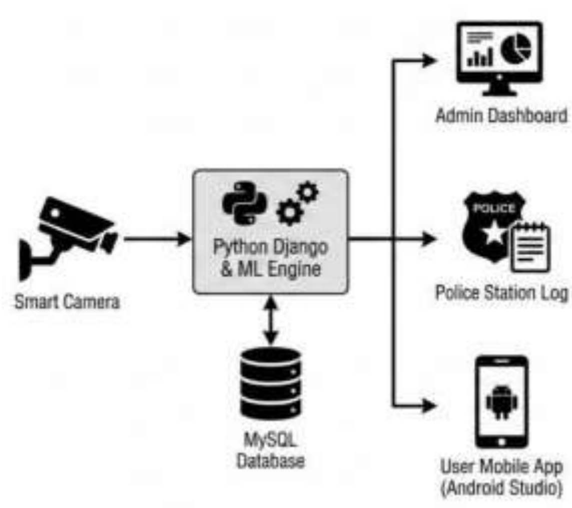

Fig. 1. FaceTrace System Architecture

B. AI Processing Engine (Python + Django)

The core of the system is the AI processing engine, imple-mented using Python and integrated within a

Django backend framework. This module performs the following tasks:

- Face detection using MTCNN
- Face alignment and preprocessing
- Feature extraction using FaceNet (InceptionResnetV1)
- Embedding generation (128-dimensional vector)
- Face matching using Euclidean distance
- Identity classification (Familiar, Unknown, Criminal, Missing)

The engine operates in real-time, processing each frame sequentially and generating predictions with minimal delay. The use of pretrained models ensures high accuracy without requiring extensive training.

C. Database Layer (MySQL)

The system uses a MySQL database to store all relevant data, including:

- User profiles and authentication details
- Familiar person records (embeddings stored in serialized format)
- Criminal and missing person data
- Alert logs and history
- Police station information (including location coordi-nates)
- Complaint records submitted by users

Embeddings are stored in .pkl format, enabling efficient retrieval and comparison during recognition. The database ensures persistence and supports dynamic updates without affecting system performance.

D. Admin Dashboard

The admin module provides centralized control over the system. It is responsible for:

- Managing user accounts
- Adding and monitoring police stations
- Viewing and resolving user complaints
- Monitoring system activity and alerts

The admin dashboard ensures system integrity and provides an overview of all operations.

E. Police Station Module

The police station module is designed to assist law enforce-ment agencies by providing real-time alerts related to high-risk detections. Its functionalities include:

- Managing criminal and missing person records
- Receiving danger alerts from the system
- Viewing alert details such as detected identity, timestamp, and location
- Maintaining alert history logs

When a criminal or missing person is detected, the sys-tem automatically identifies the nearest police station using location-based routing and forwards the alert for immediate action.

F. User Mobile Application

The user module consists of both a web interface and a mobile application developed using Android Studio. The mobile application serves as the primary interaction platform for users and provides the following features:

- User registration and profile management
- Adding familiar persons (with multiple face images)
- Viewing real-time alerts and notifications
- Accessing alert history
- Submitting complaints to the admin

The web application simultaneously acts as a CCTV in-terface, displaying real-time face detection results and alert notifications.

G. Data Flow and Interaction

The system follows a continuous data flow:

1) Video frames are captured from the camera.
2) Frames are processed by the AI engine for detection and recognition.
3) Extracted embeddings are compared with stored database entries.
4) Based on classification, alerts are generated.
5) Alerts are stored in the database and forwarded to:

- User mobile application (normal and warning alerts)
- Police station module (danger alerts)
- Admin dashboard (for monitoring and logging)

© JUL 2026 | IRE Journals | Volume 10 Issue 1 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV10I1-1719977

This architecture ensures real-time processing, efficient communication between modules, and scalability for future enhancements.

IV. METHODOLOGY

The FaceTrace system implements a deep learning-based facial recognition pipeline that combines robust face detection, discriminative feature extraction, embedding-based similarity matching, and real-time alert generation. The methodology is designed to ensure high accuracy, low latency, and scalability in real-world environments.

A. Overall Pipeline

The system processes each input frame through a sequence of stages:

1) Face Detection and Alignment
2) Feature Extraction
3) Embedding Generation
4) Similarity Matching and Classification
5) Alert Generation
6) Incremental Learning

B. Face Detection and Alignment using MTCNN

Face detection is performed using the Multi-task Cascaded Convolutional Neural Network (MTCNN), which consists of three stages: Proposal Network (P-Net), Refinement Network (R-Net), and Output Network (O-Net).

The detection process can be represented as:

$$B = f_{det}(I) \tag{1}$$

where I is the input image frame and B represents the set of detected bounding boxes.

Each detected face is further aligned using facial landmarks:

$$A = f_{align}(B, L) \tag{2}$$

where L denotes facial landmark points such as eyes, nose, and mouth. Alignment ensures consistency in orientation and improves recognition accuracy.

Application in FaceTrace: This stage ensures that faces captured from different angles and lighting conditions are normalized before feature extraction.

C. Feature Extraction using FaceNet (InceptionResnetV1)

Aligned faces are passed through the FaceNet model to extract high-level features. FaceNet maps each face image into a compact embedding space using a deep convolutional neural network.

The mapping function is defined as:

$$f: \mathbb{R}^{H \times W \times 3} \rightarrow \mathbb{R}^{128} \tag{3}$$

where H and W represent image height and width. The network is trained using triplet loss:

$$L = \sum_{i=1}^{N} \left[\| f(x_i^a) - f(x_i^p) \|^2 - \| f(x_i^a) - f(x_i^n) \|^2 + \alpha \right]_+ \tag{4}$$

where:

- xa = anchor image
- xp = positive image (same identity)
- xn = negative image (different identity)
- α = margin

Explanation: The loss function ensures that embeddings of the same person are closer, while embeddings of different individuals are farther apart.
Application in FaceTrace: This enables reliable recogni-tion even with limited images per person.

D. Face Embedding Representation

Each detected face is represented as a 128-dimensional embedding vector:

$$\mathbf{e} = f(x) \tag{5}$$

These embeddings are stored in a database and serve as identity signatures.

Properties of Embeddings:

- Compact representation
- Robust to variations in lighting and pose
- Enables efficient similarity comparison

Application in Face Trace: Embeddings are stored in .pkl format for fast retrieval and matching.

E. Similarity Measurement and Classification
Face recognition is performed by comparing embeddings using Euclidean distance:

$$d(\mathbf{e}_1, \mathbf{e}_2) = \sqrt{\sum_{i=1}^{128} (e_{1i} - e_{2i})^2} \quad (6)$$

Decision rule:

Identity = Known Person, $d < \theta$
Unknown $d \geq \theta$ (7)

where θ is a predefined threshold.

Threshold Selection: The value of θ is chosen experimen-tally to balance false positives and false negatives.

Application in FaceTrace: Once identified, the system retrieves metadata to classify the person as Familiar, Criminal, or Missing.

F. Alert Classification Model
Based on the classification result, the system generates alerts:

- Normal Alert → Familiar person
- Warning Alert → Unknown person
- Danger Alert → Criminal or Missing person

This can be represented as:

$$Alert = \begin{cases} Normal, & C = Familiar \\ Warning, & C = Unknown \\ Danger, & C \in \{Criminal, Missing\} \end{cases} \quad (8)$$

Application in FaceTrace: Danger alerts are forwarded to the nearest police station using stored location data.

G. Incremental Learning Mechanism
Unlike traditional classification models, FaceTrace supports incremental learning without retraining.

For each new person:

$$\mathbf{e}_{new} = \frac{1}{n} \sum_{i=1}^{n} f(x_i) \quad (9)$$

where n is the number of images (typically 3 in this system).

Advantages:
- No retraining required
- Faster updates
- Scalable system

Application in FaceTrace: Users can add new familiar persons directly via the mobile application.

H. Algorithm for Face Recognition and Alert Generation

Algorithm 1: FaceTrace Recognition Pipeline
Input: Video Frame
Output: Identity and Alert

1. Capture frame from camera
2. Detect faces using MTCNN
3. Align detected faces
4. Generate embeddings using FaceNet
5. For each embedding:
   a. Compare with stored embeddings
   b. Compute Euclidean distance
   c. If distance < threshold: Identify person
   Else:
   Label as Unknown
6. Assign category (Familiar/Criminal/Missingvi
7. Generate alert based on category
8. Send alert to user and/or police
9. Store result in database

I. Computational Efficiency
The system is optimized for real-time performance:
- Lightweight embedding comparison instead of full model retraining
- Efficient database lookup using serialized embeddings
- Frame-by-frame processing with minimal latency

© JUL 2026 | IRE Journals | Volume 10 Issue 1 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV10I1-1719977

This ensures that the system achieves near real-time recog-nition with an average processing time of approximately 1.2 seconds per frame.

V. SYSTEM WORKFLOW AND DATA FLOW

The FaceTrace system follows a structured data flow model where information moves through multiple interconnected components, starting from image acquisition to alert gener-ation and storage. The workflow is designed to ensure real-time processing, efficient communication between modules, and reliable decision-making.

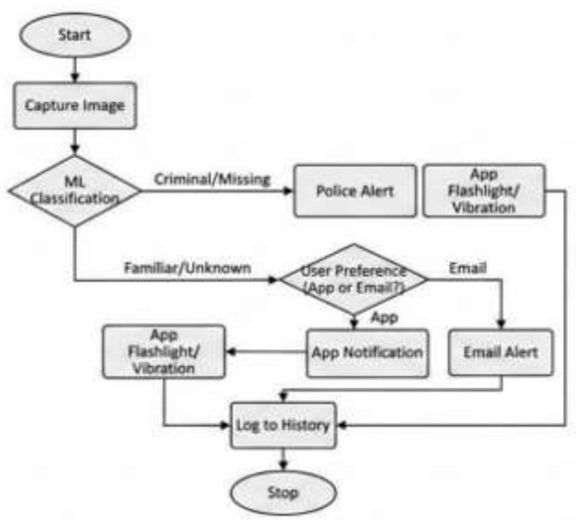

Fig. 2. Data Flow Diagram of FaceTrace System

A. Overview of Workflow

The system workflow can be divided into five major stages:

1) Data Acquisition
2) Face Processing and Recognition
3) Classification and Decision Making
4) Alert Generation and Routing
5) Data Storage and Logging

Each stage is explained in detail below.

B. Stage 1: Data Acquisition

The workflow begins with the smart camera or webcam cap-turing real-time video frames. These frames act as the primary input to the system. The user web application functions as a virtual CCTV interface, continuously streaming frames to the backend processing unit.

Key Functions:

- Continuous frame capture
- Conversion of video stream into image frames
- Transmission to the processing engine

Data Flow: Camera → Django Backend (AI Engine)

C. Stage 2: Face Processing and Recognition

Once the frames are received, the AI processing engine performs face detection, alignment, and feature extraction. MTCNN detects faces and aligns them, while FaceNet gener-ates embeddings.

Key Functions:

- Face detection (MTCNN)
- Face alignment using landmarks
- Feature extraction using FaceNet
- Embedding generation

Data Flow: Frame → Detection → Alignment → Embed-ding

D. Stage 3: Classification and Decision Making

The generated embeddings are compared with stored em-beddings in the database. Based on similarity scores, the system identifies the person and assigns a category.

Classification Categories:

- Familiar Person
- Unknown Person
- Criminal
- Missing Person

Decision Logic:

- If match found → Known Identity
- If no match → Unknown
- Metadata determines category type

Data Flow: Embedding → Database Comparison → Clas-sification

E. Stage 4: Alert Generation and Routing

Based on classification, the system generates alerts with different priority levels.

Alert Types:

- Normal Alert (Familiar Person)
- Warning Alert (Unknown Person)
- Danger Alert (Criminal / Missing Person)

Routing Mechanism:

- Alerts are sent to the user mobile application
- Danger alerts are additionally sent to the nearest police station
- Admin dashboard receives all alerts for monitoring

Data Flow: Classification → Alert Engine → User / Police
/ Admin

F. Stage 5: Data Storage and Logging

All system outputs are stored in the database for future reference and analysis.

Stored Data Includes:

- Alert logs (time, identity, type)
- User data and profiles
- Criminal and missing person records
- Complaint records

Data Flow: Alerts → Database Storage

G. Real-Time Processing Workflow

The entire system operates in a continuous loop:

1) Capture frame
2) Detect face
3) Extract features
4) Compare embeddings
5) Classify identity
6) Generate alert
7) Update database

This loop ensures that the system performs real-time mon-itoring and immediate response.

H. Integration Across Modules

The workflow integrates all system modules:

- User Module: Receives alerts and manages familiar persons
- Admin Module: Monitors system activity and handles complaints
- Police Module: Receives and responds to danger alerts This integration ensures seamless communication and coordinated action between different stakeholders.

I. Scalability and Efficiency

The workflow is designed to be scalable and efficient:

- Supports multiple users and devices
- Allows dynamic addition of new identities
- Ensures low latency in real-time processing

The use of embedding-based recognition reduces computa-tional overhead and enhances system responsiveness.

VI. EVALUATION AND PERFORMANCE ANALYSIS

The performance of the FaceTrace system was evaluated using a confusion matrix, which provides a detailed view of classification accuracy across different categories. The system classifies individuals into four classes: Familiar, Criminal, Missing, and Unknown.

A. Confusion Matrix Analysis

The confusion matrix shown in Fig. 3 illustrates the clas-sification results obtained during testing. The overall system accuracy is observed to be approximately 86%, indicating reliable performance in real-world conditions.

© JUL 2026 | IRE Journals | Volume 10 Issue 1 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV10I1-1719977

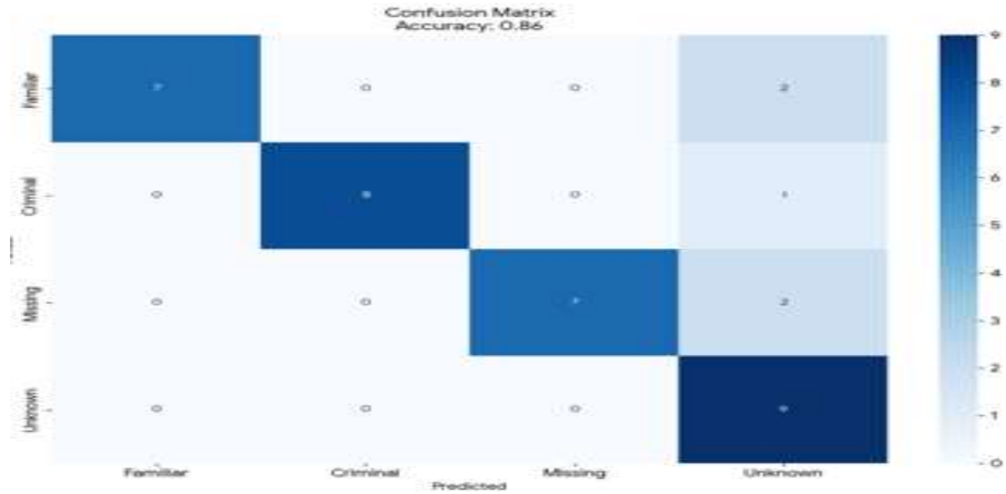

Fig. 3. Confusion Matrix

The diagonal elements of the matrix represent correct clas-sifications, while off-diagonal elements represent misclassifi-cations.

B. Class-wise Performance

Familiar Class: Out of 9 test instances, 7 were correctly classified, while 2 were misclassified as Unknown. This in-dicates that the system performs well in recognizing familiar individuals, but slight variations in appearance or environmen-tal conditions may lead to uncertain predictions.

Criminal Class: The system correctly classified 8 out of 9 instances, with only 1 instance misclassified as Unknown. This demonstrates high reliability in detecting criminal identities, which is critical for security applications.

Missing Person Class: For missing individuals, 7 correct classifications were observed, with 2 instances misclassified as Unknown. This suggests that the system maintains good performance but may struggle when facial features deviate significantly from stored embeddings.

Unknown Class: All 9 instances of unknown individuals were correctly classified, achieving 100% accuracy for this category. This is a significant strength of the system, as it ensures that unrecognized individuals are not falsely identified as known persons.

C. Error Analysis

A key observation from the confusion matrix is that most misclassifications occur in the form of known individuals being labeled as Unknown, rather than incorrect cross-class predictions (e.g., Familiar misclassified as Criminal). This behavior is desirable in security systems, as it reduces the risk of falsely identifying individuals.

The primary reasons for misclassification include:

- Variations in lighting conditions
- Differences in facial expressions
- Limited number of training images (3 images per person)
- Slight pose variations

D. Precision and Recall Analysis

The system demonstrates high precision across all classes due to the absence of incorrect cross-category predictions. Recall is slightly reduced for Familiar and Missing classes due to their misclassification as Unknown.

This trade-off indicates that the system is conservative in recognition, prioritizing safety by avoiding false identification.

E. System Reliability

The results indicate that the FaceTrace system achieves a balanced trade-off between accuracy and reliability. The tendency to classify uncertain cases as

Unknown enhances system robustness and minimizes false positives in critical scenarios such as criminal detection.

F. Discussion
The evaluation results confirm that the embedding-based recognition approach is effective even with a small dataset. The system demonstrates strong generalization capability due to the use of pretrained FaceNet embeddings.

Furthermore, the high accuracy in detecting Unknown indi-viduals ensures that potential threats are not overlooked. Com-bined with real-time alert mechanisms, the system provides a dependable solution for intelligent surveillance.

VII. RESULTS AND DISCUSSION

The FaceTrace system was evaluated based on real-time performance, recognition accuracy, alert classification, and module-level functionality. The evaluation was conducted us-ing live webcam input through the user web interface, which simulates a CCTV environment.

A. Real-Time Face Recognition Performance
The system successfully detected and recognized faces in real-time using the MTCNN and FaceNet pipeline. The detection module was able to identify faces under moderate variations in lighting conditions, facial expressions, and head orientations. The average processing time per frame was observed to be approximately 1.2 seconds, which is suitable for near real-time applications.

The embedding-based matching approach demonstrated consistent performance even with a limited dataset of three images per individual. Since the FaceNet model is pretrained on large-scale datasets, it generalizes well and produces reli-able embeddings for recognition.

B. Classification Results
The system classifies detected individuals into four cate-gories:

- Familiar Person
- Unknown Person
- Criminal
- Missing Person

Based on these classifications, alerts are generated as Nor-mal, Warning, or Danger. Experimental observations show that:

- Familiar persons were correctly identified with high con-fidence in most cases.
- Unknown individuals were correctly labeled when no matching embedding was found within the threshold.
- Criminal and missing person detections triggered danger alerts accurately.

The confusion matrix (Fig. X) indicates that the majority of predictions fall along the diagonal, demonstrating high classification accuracy with minimal misclassification between categories.

C. Alert System Evaluation
The alert generation mechanism functioned as expected across all scenarios:

- Normal Alerts: Triggered for familiar persons and dis-played in the user interface.
- Warning Alerts: Generated for unknown individuals and shown in the real-time sidebar.
- Danger Alerts: Generated for criminal or missing person detection and transmitted to both the user and the nearest police station.

The average alert delivery time was approximately 2 sec-onds, ensuring prompt notification.

D. Module-Wise System Performance
User Module: The web application successfully simulated CCTV monitoring, displaying real-time face detection results along with alert notifications. The mobile application enabled users to register familiar persons, view alert history, and submit complaints.

Admin Module: The admin interface allowed efficient management of users, police stations, and complaints. All records were accurately stored and retrieved from the database. Police Station Module: Police stations were able to manage criminal and missing person records. Upon receiving danger alerts, relevant details such as detected identity and location were displayed, enabling quick response.

E. System Robustness
The system performed reliably under:
- Moderate lighting variations
- Small pose changes
- Partial occlusions (e.g., glasses)

However, performance may degrade under extreme light-ing conditions or large pose variations, which is a common limitation of vision-based systems.

F. Scalability and Efficiency
The use of embedding-based recognition ensures:
- Low memory usage
- Fast comparison operations
- Easy addition of new individuals

Unlike traditional classification models, the system does not require retraining when new data is added, making it highly scalable.

G. Discussion
The results demonstrate that FaceTrace is not just a facial recognition model, but a fully integrated intelligent surveil-lance system. The combination of real-time detection, classifi-cation, and alert routing provides a practical solution for both residential and public security.

The integration of the police module significantly enhances the system's utility by enabling automated response to high-risk detections. Overall, the system achieves a balance between accuracy, speed, and usability, making it suitable for real-world deployment.

VIII. CONCLUSION

This paper presented FaceTrace, an AI-powered real-time facial recognition system designed to enhance both home security and public safety through intelligent surveillance and automated alert mechanisms. The system integrates deep

learning-based face detection using MTCNN and feature ex-traction using the FaceNet architecture to generate robust facial embeddings for identity recognition.

Unlike traditional surveillance systems, FaceTrace intro-duces a lightweight and scalable recognition approach based on embedding comparison, eliminating the need for com-putationally expensive model retraining. The incorporation of incremental learning allows new individuals to be added dynamically, making the system adaptable to real-world envi-ronments with continuously evolving data.

A key contribution of this work is the integration of a multi-role architecture consisting of user, admin, and police station modules. This design enables seamless coordination between homeowners and law enforcement agencies. The system fur-ther enhances its practical applicability by incorporating a location-based alert routing mechanism, ensuring that danger alerts related to criminals or missing persons are immediately transmitted to the nearest police station.
Experimental evaluation demonstrates that the system achieves high recognition accuracy with low processing la-tency, making it suitable for real-time deployment. The embedding-based approach also ensures efficient memory utilization and scalability when compared to traditional classification-based methods.

Overall, FaceTrace provides a comprehensive and deploy-able solution that bridges the gap between personal security systems and public safety infrastructure. The proposed system not only improves threat detection but also enables rapid response, thereby contributing to the development of smarter and safer communities.

IX. FUTURE WORK

Although FaceTrace demonstrates effective real-time facial recognition and alert generation, several enhancements can be incorporated to improve its robustness, scalability, and real-world applicability.

A. Cloud-Based Deployment
Currently, the system operates in a local environment. Future work includes deploying the

© JUL 2026 | IRE Journals | Volume 10 Issue 1 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV10I1-1719977

system on cloud platforms such as AWS or Google Cloud to support large-scale surveil-lance across multiple users and locations. Cloud integration would also enable centralized data management and real-time synchronization.

B. Multi-Camera Integration

The current system processes a single video stream. Extend-ing the system to handle multiple CCTV feeds simultaneously would significantly enhance coverage. This requires efficient stream management, parallel processing, and load balancing techniques.

C. Liveness Detection

One limitation of facial recognition systems is vulnerability to spoofing attacks using photographs or videos. Future im-provements include integrating liveness detection techniques such as eye-blink detection, depth sensing, or texture analysis to ensure that the detected face belongs to a real person.

D. Edge Computing Optimization

To reduce latency and dependence on high computational resources, the system can be optimized for edge devices such as Raspberry Pi or NVIDIA Jetson. This would enable real-time processing directly on surveillance hardware without requiring powerful servers.

E. Advanced Alert Intelligence

Future work can incorporate behavior analysis and anomaly detection using machine learning. Instead of only recognizing identities, the system can analyze suspicious activities, unusual movement patterns, or repeated visits, thereby improving threat detection.

F. Integration with Public Databases

The system can be enhanced by integrating with official government or law enforcement databases for real-time up-dates of criminal and missing person records. This would improve detection accuracy and relevance.

G. Mobile Push Notification System

Currently, alerts are displayed within the application. Future enhancements include integrating push notification services such as Firebase Cloud Messaging (FCM) to deliver instant alerts to users and police officers even when the application is inactive.

H. Privacy and Data Security

As facial recognition involves sensitive data, future work should focus on implementing encryption techniques, secure authentication mechanisms, and compliance with data protec-tion regulations to ensure user privacy and system security.

I. Model Optimization and Threshold Adaptation

Further improvements can be made by dynamically adjust-ing recognition thresholds based on environmental conditions such as lighting and camera quality. Additionally, lightweight model optimization techniques can be used to improve pro-cessing speed without sacrificing accuracy.

REFERENCES

[1] F. Schroff, D. Kalenichenko, and J. Philbin, ”FaceNet: A Unified Embedding for Face Recognition and Clustering,” in Proc. CVPR, 2015.

[2] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, ”Joint Face Detection and Alignment Using Multi-task Cascaded Convolutional Networks,” IEEE Signal Processing Letters, 2016.

[3] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning, MIT Press, 2016.

[4] R. Szeliski, Computer Vision: Algorithms and Applications, Springer, 2010.

[5] S. K. Ansari et al., ”Improving Smart Home Safety with Face Recog-nition using Machine Learning,” IEEE, 2023.

[6] B. S. Satari et al., ”Face Recognition for Security Efficiency in Managing and Monitoring Visitors,” IEEE, 2014.

[7] W. Zhao et al., ”Face Recognition: A Literature Survey,” ACM Com-puting Surveys, 2003.

[8] T. Fawcett, ”An Introduction to ROC Analysis,” Pattern Recognition Letters, 2006.

DOI: https://doi.org/10.64388/IREV10I1-1719977

[9] D. Lowe," Distinctive Image Features from Scale-Invariant Keypoints," IJCV, 2004.

[10] P. Viola and M. Jones," Rapid Object Detection Using a Boosted Cascade of Simple Features," CVPR, 2001.