Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Development of an Intelligent Real-Time Multi-Camera Facial Recognition and Tracking System

Mogeeb M. Saeed

mogeeb.saeed@alsaeeduni.edu.ye

Al-Saeed University

Osama M. Sallam

Taiz University

Mohanad W. Alghazali

Taiz University

Salman A. Ahmed

Taiz University

Ayman K. Al-Kuhlani

Taiz University

Research Article

Keywords: Face Recognition, Real-Time Tracking, Deep Learning, Multi-Camera Surveillance, Intelligent Security Systems

Posted Date: November 23rd, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11365724/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations:
No competing interests reported.

Development of an Intelligent Real-Time Multi-Camera Facial Recognition and Tracking System

Mogeeb A. Saeed[1]; Osama M. Sallam[2]; Mohanad W. Alghazali[3]; Salman A. Ahmed[4]; Ayman K. Al-Kuhlani[5]

[1] Department of cybersecurity, Faculty of Engineering & IT, Alsaeed University, Yemen

[2,3,4,5] Department of Information Technology, Faculty of Engineering & IT, Taiz University, Yemen

*Corresponding author E-mail: mogeeb.saeed@alsaeeduni.edu.ye

ABSTRACT

In light of the increasing security demands, traditional surveillance systems suffer from limitations such as reliance on manual monitoring, high error rates, and a lack of integrated forensic capabilities. This research paper presents the design and implementation of an intelligent real-time face recognition and tracking system, designed to automate the processes of detecting, tracking, and identifying individuals across multiple video streams. The system uses YOLOv8 for fast and accurate face detection, SORT for a reliable and efficient real-time tracking mechanism, Dlib-based deep learning models, and Principal Component Analysis (PCA) for effective feature extraction. The Support Vector Machine (SVM) classifier is used offline to verify the quality of face embeddings, achieving test accuracy exceeding 97% across multiple standard and custom datasets, including "Labeled Faces in the Wild" (LFW), Pins, and a manually prepared dataset. Experimental tests prove perfect consistency between detection speed, recognition rate, and tracking stability since the system supports multiple cameras' data at a mean rate of 8.1 frames per second on an NVIDIA CUDA-compatible GPU. The full-featured desktop application includes automated notification, forensic evidence management, and database management, with a full-featured and complete solution to modern surveillance applications.

Keywords: Face Recognition, Real-Time Tracking, Deep Learning, Multi-Camera Surveillance, Intelligent Security Systems.

Introduction

In the present age of digitalization, intelligent surveillance systems are now a part of public security infrastructure and business applications and are imbuing human surveillance with analytical capabilities that substitute its erstwhile capabilities. The significant and sustained investment in this field is underscored by market analysis, which projects the global facial recognition market will expand to USD 13.4 billion by 2028, growing at a compound annual growth rate of 16.3% [1], The significance and growing application of multi-camera systems for surveillance for enhanced security and identification control are amply documented [2]. Along with this tendency, Kabanda G. [3] also provide a comprehensive description of multi-camera monitoring networks in surveillance systems, highlighting the significance of such networks in the contemporary world of security systems. All existing systems, both manual and semi-automatic, necessarily have a radical drawback seriously impairing their effectiveness. Depending on the human staff to monitor multiple streams of cameras simultaneously

necessarily leads to fatigue and fluctuating attention, increasing the danger of catastrophic security errors.

On the other hand, most existing automated systems are not able to handle elaborate real-world circumstances. They struggle with recognition accuracy where there are changing light conditions, moving camera positions, and dense crowding, leading to high rates of misidentification or missed detection. Moreover, the majority of these systems lack embedded forensic tools, thus are passive and not proactive. This deficiency is evident when conducting an investigation of an incident since systems do not offer a means to review past detection records or correlate them with visual evidence, making future investigations difficult [4].

To fill these inadequacies, this work introduces an intelligent, multi-camera system for real-time face recognition and tracking. The long-term goal is to develop an integrated solution that not only identifies targeted individuals correctly and at pace, but also provides a robust mechanism for continuous tracking over a network of cameras and analysis capabilities to facilitate investigation. The methodology suggested is on the basis of state-of-the-art AI model integration using YOLOv8 for initial face detection and then SORT for motion tracking and deep learning for feature extraction and recognition [5],[6].

The major contributions of the work are:

1. Building a hybrid system architecture that combines the speed of face detection of YOLOv8 and tracking performance of SORT to facilitate uninterrupted and robust real-time surveillance.
2. Creating a comprehensive solution with real-time detection, instant alarm, and forensic analysis modules in the aim of providing end-to-end solution for the entire security process.
3. Creating and implementing a complete desktop user interface through which easy system management, monitoring, and analysis of data can be made available in such a manner that state-of-the-art technology is made accessible to security officers.
4. Test the performance of the system on standard (e.g., LFW) and self-collected data, and demonstrate its usability in practice.

The paper is structured as follows: **Section 2** gives an overview of literature and work in the field. **Section 3** gives a clear description of the methodology proposed. **Section 4** is devoted to experimental discussion and results presentation. **Last Section 5** contains a conclusion.

RELATED WORKS

There are some works in face detection, tracking, and recognition. The majority of the works try to achieve these goals from different aspects, either by building a multi-camera system, single-camera system, or focusing on recognition only. But all the works try to build an effective face recognition system. We have classified these works into two large categories as follows:

Face Detection and Recognition Research

Jayaram et al. [7] presented an effective framework to detect and recognize multiple persons in videos using deep learning using YOLOv8. The primary objective was to improve surveillance person recognition performance by combining YOLOv8 with a modified CSPDarknet53 architecture. The study utilized a new dataset called S-DEC and the VTB dataset for evaluation. The result exhibited high performance, but the experiment lacked a clear tracking process to preserve the identity of individuals from frame to frame. Aarthi Rajagopalan et al [8] This study

contrasted two deep learning approaches to building a real-time face recognition system. One used a GoogLeNet model with alterations and the other used a FaceNet model. A home-bred, manually created dataset was utilized. The results showed that the FaceNet approach was stronger, more accurate, and data-efficient and showed the strength of metric learning over hyperparameter tuning for plain classification for face recognition tasks. Suguna et al. [9] Here, face recognition was realized in real-time using a combination of Multi-Task Cascaded Convolutional Networks (MTCNN) for face detection, FaceNet for feature extraction (embeddings), and SVM for classification. The system was pre-trained with the LFW dataset and tested using a self-generated dataset. The proposed system achieved a high accuracy of 99.85% when the faces were in front view or rotated slightly, but could not maintain the recognition when the face was turned fully to the side.

Face Detection, Recognition, and Tracking Experiments

Majeed et al. [10] proposed a framework for face recognition and tracking in real-time based on the integration of a light-weight convolutional neural network (CNN) and hybrid feature extraction methods (LDA and HOG). The proposed framework was evaluated using the MUCT database. The proposed framework achieved optimal performance metrics, surpassing similar models. However, the study identified the need for more extensive and diverse databases to avoid overfitting, and the current evaluation was limited to observing a single individual. Azatbekuly et al. [5] developed a smart video surveillance system for offices using the YOLOv8 algorithm. The system utilized YOLOv8-nano for object detection and utilized a Dlib-based library for face feature detection. The constructed modules worked effectively in real-world office use, but it has not been implemented on resource-constrained or low-end devices. Jiang et al. [11] proposed a novel method called "face tracking feature extraction" to improve real-time multi-face recognition systems. The system used MTCNN for face detection and a location-based tracking algorithm for avoiding the re-extraction of features for each frame. The result showed significant improvement in the frame rate (FPS) without decreasing accuracy. However, the system is still not robust enough in situations with overlapping faces or moving faces at a higher speed. Li, Y. et al. [4] proposed a reinforcement learning-based method combining face detection and dynamic tracking. MTCNN was used by the system to detect the face and a Kalman filter-based algorithm with SORT framework for tracking. The performance outcomes revealed extremely high improvements in the detection success rate and dynamic capture. However, the system's effectiveness is limited by the relatively lower detection capacity of MTCNN in complex real-world conditions. Durga et al. [12] In this paper, a framework for face recognition and tracking of multiple faces was provided. The framework integrated various approaches such as Haar Cascades for detection, LBP for feature extraction, FaceNet for face recognition, and SORT and DeepSORT algorithms for tracking. The model performed well under poor conditions. The weakness of the study is that it has been tested on only a very small, hand-crafted dataset, i.e., the result is not generalizable.

Research proposed method

This research aims to develop an intelligent system for real-time face recognition and tracking using multiple cameras. This section explains the methodology followed in our research step by step, starting from data selection and processing, through preprocessing and feature engineering, to implementing and validating the basic models for detection, tracking, and recognition.

Figure 1. Overall Architecture of the Proposed Face Recognition System

Selecting data

To ensure the robustness and accuracy of the system, we used a variety of data sets. For the face detection task, a dataset of over 16,700 OIDv4 images was collected from Google, annotated in the YOLO format. To develop the recognition module and measure its performance, we used a several of data sets, including the academic benchmark "Labeled Faces in the Wild" (LFW), the "Face Recognition from Pins" dataset, the "Five Celebrity Faces" dataset, and a small "Handheld Faces" dataset. We designed it specifically for rapid prototyping and system tuning.

Pre-processing data and Augmentation

We created a unified preprocessing pipeline to prepare the data. We resized all images to uniform dimensions: 640 × 640 pixels for the YOLOv8 detection model and 160 × 160 pixels for the feature extraction model. To improve the generalization of the YOLOv8 detector, we applied several data augmentation techniques during training, including random horizontal flips, brightness and contrast adjustments, and slight scale and rotation changes. To train and evaluate the model, we split the datasets into 80/20 training and test sets.

Face Detection and Alignment

The first operational step in the pipeline is to locate faces in the image. We used the YOLOv8n model for this task due to its optimal balance of speed and accuracy for real-time applications. To enhance the quality of the detected face for recognition, we applied a margin expansion technique, where the detected checkbox is enlarged by 10% to ensure the entire face area is captured.

Figure 2. Face Detection Process

After detection, a geometric alignment step is performed to address head pose variations. The MediaPipe Face Mesh model is used to detect 468 facial landmarks. From these landmarks, the eye center coordinates are used to calculate the head rotation angle, then applied a convolutional transform to the image to normalize the eyes. This normalization is essential for the consistency of the feature extraction process.

Figure 3. All Alignment Process

Feature Extraction and Verification

The aligned 160x160 pixel face image is fed into a pre-trained deep neural network from the Dlib library. This network outputs a 128-dimensional vector, or embedding, that serves as a unique digital face print. To create a more robust and coherent feature vector, we apply Principal Component Analysis (PCA) to these 128-dimensional embeddings. This process reduces the dimensions to 50 components, maintain maximum variance and enhancing the discriminative ability of the landmarks.To quantitatively validate the quality of these PCA-enhanced embeddings, we conducted offline evaluation using SVM. We trained the SVM to classify embeddings from our labeled datasets. The high accuracy achieved in these tests confirmed that our feature engineering process produces highly reliable and separable embeddings, which is suitable for an effective recognition system.

Figure 4. Tracking and Recognition Methodology

The final system integrates pre-validated components into a streamlined, efficient, and real-time pipeline. The tracking component is based on the Simple Online Real-Time Tracking (SORT) algorithm. SORT receives bounding box detections from the YOLOv8 face detector and assigns a unique and consistent trajectory identifier (Track ID) to each person. To maintain temporal stability and preserve identities across frames, SORT uses a Kalman filter that models object motion and predicts future locations even during brief occlusions or detection failures. Additionally, the Hungarian algorithm is used for efficient data correlation, matching new detections with existing trajectories based on spatial proximity. For the recognition phase, the system uses face embeddings generated by a pre-trained deep learning model from the Dlib library. These embeddings are initially 128-dimensional vectors representing distinctive facial features; however, to achieve computational efficiency and improve matching robustness, PCA is applied to reduce their dimensionality, typically to 50 principal components. The system performs recognition by calculating the Euclidean distance between the currently tracked facial embedding and the embeddings stored in the database for known individuals. The Euclidean distance calculated as follows:

$$d(A,B) = \sqrt{\sum_{i=1}^{N}(Ai - Bi)^2} \qquad (1)$$

A smaller Euclidean distance indicates a greater similarity between the two embeddings, meaning that the faces most likely belong to the same person. The system sets a predetermined threshold (such as 0.4), determined empirically through validation experiments, to distinguish matches from mismatches. When the distance between the tracked embedding and any stored embedding falls below this threshold, identification is confirmed, and the system assigns the corresponding tag to the trajectory identifier. This distance-based matching methodology significantly reduces computational costs compared to frame-by-frame classification using SVM. It enables fast and reliable real-time identification, which is critical for multi-camera surveillance scenarios that process multiple video streams simultaneously. By combining the accuracy and speed of YOLOv8 for detection, SORT for accurate tracking, and enhanced embeddings using PCA with Euclidean distance-based identification, the system achieves efficient and consistent identification of individuals in dynamic and complex environments.

System Implementation and Architecture

System Architecture:

Figure 5. System Architecture

The system was implemented as a standalone desktop application with rich and secure surveillance smartness. This system architecture was chosen to deliver robust performance and autonomous data handling, which makes it a perfect candidate for deployment in secure environments. The system maintains support for adding a number of specified IP cameras, which are processed using the Real-Time Streaming Protocol (RTSP), and webcams located locally, and the video streams are processed using the OpenCV library. Our system is fault-tolerant; all the critical data, including user profiles, criminal records, facial embeddings, and detection logs, are stored and kept in a MySQL database. The system also provides a feature-rich logging capability

that records all significant events and user activities for auditing and debugging purposes. Key architectural features include:

1. A secure authentication module for users to control access.
2. Config panels for user, camera source, and criminal profile handling.
3. An automated email notification system to deliver event-driven notifications to pertinent staff members.
4. An evidence capture module which captures automatically video segments of validated recognitions.

User Interface and Functional Workflow:

To demonstrate the functioning of the system, we describe the core components of the graphical user interface (GUI) and their operational order. The Main Dashboard serves as the central location for real-time monitoring. It provides a grid layout of multiple cameras with real-time display of video streams. When a known individual is identified, the person's face is highlighted with a labeled bounding box, and an entry is added promptly to a real-time event log on the same screen. The same interface also displays the most crucial system statistics, such as processing rate (FPS) and the number of objects in tracking, providing instantaneous feedback about the system performance. For forensic processing, Criminals Management & Investigation interface was designed. This is a very useful tool by which administrators can manage the database of people recognized. The tool supports searching by name or by uploading an image for similarity matching. A valuable feature is the "Detection History" view, which provides an absolute chronological list of all sightings of a selected person logged, so it is simple to analyze following the event. The system includes an Offline Video Analysis module. This interface allows the selection of a pre-recorded video file to play via the same detection and recognition pipeline as live streams. It is for forensic analysis of past events, generating a report of all the people detected in the video. Finally, for administrative reasons, there are special interfaces to User Management and System Settings. These allow administrators to control user levels of access and add or modify camera stream sources, thus making the system flexible and easy to administer in a live situation.

Main System Interfaces:

Figure 6. Main Dashboard

For greater clarity, the system's user interfaces are briefly summarized as follows:

1. **Main Dashboard (Surveillance Interface):**

This central interface displays live video feeds from all configured cameras within a unified network. It dynamically highlights recognized faces and records detection events in real time with detailed information such as identity, camera source, and timestamp. The dashboard also provides basic controls for starting and stopping surveillance, enabling comprehensive situational awareness. as shown in figure.6.

2. **Adding a Criminal Interface:**

Authorized users can add new criminal profiles by entering personal and criminal history details. Facial images can be uploaded or captured directly from connected cameras. Upon saving, the

system automatically processes the image to create and store a facial embedding, integrating the individual into the live recognition database. as shown in figure.7.

Figure 7. Add a criminal (suspect)

3. Criminal Management and Investigation Interface:

Figure 8.Criminal Management

This interface supports advanced search functions by name or facial similarity, along with record management. It features a chronological detection log, listing all recorded appearances of each individual across cameras. This functionality helps track movements and conduct detailed investigations. Crime Detail Interface: Automatically enabled by high-confidence recognition, it displays a direct comparison between stored database images and live camera footage, along with box selections and metadata such as detection time, camera source, and criminal history summary, facilitating quick and accurate verification. as shown in figure 8.

Results and discussion

Results

The performance of the proposed system was evaluated to determine its effectiveness in face detection, recognition, and real-time tracking. This section presents the quantitative results from our experiments. We conducted the evaluation in three major sections to verify the efficiency of each stage:

first, evaluating the performance of the YOLOv8s detection model;

second, verifying the quality of the feature extraction pipeline using an SVM classifier; and

third, the performance testing of the integrated system in a real-time operational scenario.

Face Detection Results

To select the optimal model for real-time face detection, we compared the performance of YOLOv8s against other models. The experiments showed that YOLOv8s is significantly faster than traditional methods like MTCNN while maintaining high accuracy. The performance of the YOLOv8s model was calculated using metrics such as precision, recall, and mean Average Precision (mAP). We compare two instances of the YOLO family: YOLOv8s, and YOLOv11s. The result for the comparison between the two is summarized in Table 1.

The two models were trained with exactly the same arguments and against exactly the same division of training data, all models train it with 40 epochs. Then we chose the best one to implementing our proposed method.

Table 1. Comparison Between YOLO Models

Model	Metric	Validation Value	Test Value
YOLOv8s	Precision (B)	0.902	0.899
	Recall (B)	0.842	0.840
	mAP50 (B)	0.901	0.899
	mAP50-95 (B)	0.622	0.620

	Fitness	0.645	0.643
	Inference Speed (ms)	2.5	2.5
YOLOv11s	Precision (B)	0.918	0.893
	Recall (B)	0.874	0.865
	mAP50 (B)	0.935	0.893
	mAP50-95 (B)	0.650	0.648
	Fitness	0.673	0.670
	Inference Speed (ms)	15.00	16.80

YOLOv8s is very well-balanced in terms of speed and accuracy and turns out to be the most flexible model for real-time applications. YOLOv11s Although marginally better in accuracy, consumes almost double the inference time as YOLOv8s. Again, this model performs comparably well on both the visible and unseen data alike, hence proving its robustness and generalizing ability.

Figure 9. All Result in Validation Of YOLOv8s Model

The results in figure.9 show that the YOLOv8s model achieved a high mAP50 of 0.899 on the test set, confirming its accuracy. The model correctly identified 8,950 faces (True Positives) and demonstrated a strong balance between precision and recall, achieving a peak F1-Score of 0.86. The model's fast inference speed of 2.5 ms makes it highly suitable for real-time applications.

Fig. 10: Confusion Matrix in Validation Task Of YOLOv8s Model

Face Feature Extraction Results

We applied an SVM model to test the feature extraction process on several standard and custom-built datasets. The high accuracy achieved in these offline tests validates the ability of our feature engineering pipeline (Dlib + PCA) to produce highly discriminative embeddings.

Table 2: Result of Testing SVM on LFW Dataset with 5-Folds

Fold	**Train Accuracy**	**Test Accuracy**
1	98.397%	97.107%
2	98.242%	98.140%
3	98.345%	97.107%
4	98.501%	98.137%
5	98.760%	97.308%
Average	-	**97.56%**

Table 3: The Result Of Test SVM Model On 5-Celebraties Dataset With 5-Folds

Fold	**Train Accuracy**	**Test Accuracy**
1	100.0%	100.0%
2	100.0%	100.0%
3	100.0%	100.0%

4	100.0%	100.0%
5	100.0%	100.0%
Average	-	**100.0%**

The results, particularly the average test accuracy of 97.56% on LFW, 98.00% on the Pins dataset, and 99.55% on our custom manual dataset, confirm that the extracted features are robust and highly separable. This high performance can be attributed to the effectiveness of the feature extraction and enhancement techniques used.

Table 4: Result of Testing SVM on Pin Dataset with 5-Folds

Fold	**Train Accuracy**	**Test Accuracy**
1	98.412%	97.850%
2	98.530%	98.230%
3	98.290%	97.950%
4	98.610%	98.120%
5	98.745%	97.950%
Average	-	**98.00%**

Table 5: Result of Testing SVM on Manual Dataset with 5-Folds

Fold	**Train Accuracy**	**Test Accuracy**
1	99.80%	98.20%
2	99.60%	98.50%
3	99.90%	98.00%
4	99.20%	97.50%
5	99.30%	97.80%
Average	-	**98.00%**

System Performance Evaluation Results

This part evaluates the performance of the integrated surveillance system as a whole under real-time operation, both in processing speed and tracking robustness. Under live testing on a system installed with a CUDA-enabled GPU, the system processed at about an average of 8.1 frames per second (FPS) when it was tracking two simultaneous streams of cameras.

Figure 11. Result of monitoring for 2 Camera

The system demonstrated high effectiveness in identifying and tracking individuals broadcast live from two different sources (a computer camera and a phone). The combination of the fast YOLOv8 detector and the efficient SORT tracker proved effective in maintaining consistency of the individuals' tracking IDs. A known individual was correctly identified from both sources using two unique tracking IDs (ID:1 and ID:20), confirming the integrity of the system architecture. The application remained stable throughout the testing period, with all modules working as intended.

Discussion

The system effectively addresses the shortcomings of traditional surveillance by combining high-speed detection with accurate and reliable recognition and tracking. The use of YOLOv8 and SORT has proven effective as the system's backbone. However, the observed processing speed limits scalability in resource-constrained environments. Moreover, despite the high recognition accuracy, extreme facial poses and difficult lighting conditions still pose challenges. Therefore, enhancing the system's ability to handle such situations should be prioritized. Overall, the results demonstrate the validity of the integrated approach and its practical applicability in secure and sensitive environments, where instant recognition and forensic evidence support are critical.

Conclusions

This research presents a comprehensive and intelligent multi-camera facial recognition and tracking system with real-time capability for security. High accuracy and stable performance are achieved by integrating the YOLOv8 face detection algorithm, SORT tracking with high efficiency, and Dlib feature extraction with an SVM classifier. The system's modular design, coupled with an intuitive desktop application, enables live monitoring, automatic alerts, and forensic analysis with video evidence management. Experimental analysis confirms the system's face detection and recognition ability with high accuracy and speed, with recognition accuracy of more than 97% on several benchmark datasets.

Declaration of competing interest

The authors declare that they have no any known financial or non-financial competing interests in any material discussed in this paper.

Funding information

No funding was received from any financial organization to conduct this research.

Author contribution

The contribution to the paper is as follows: Mogeeb A. Saeed, Osama M. Sallam: study conception and design; Mohanad W. Alghazali: data collection; Salman A. Ahmed, Ayman K. Al-Kuhlani: analysis and interpretation of results; Mogeeb A. Saeed: draft preparation. All authors approved the final version of the manuscript.

References

[1] Markets and Markets, “Facial Recognition Market by Offering, Application, Organization Size, Deployment Mode, Vertical and Region - Global Forecast to 2028,” Markets and Markets Research Private Ltd., Delray Beach, FL, USA, Tech. Rep. TC 3421, 2023. [Online]. Available: https://www.marketsandmarkets.com/PressReleases/facial-recognition.asp

[2] Honey and S. O. Singh, “Facial Feature Detection in Real-Time: A Novel Approach with MTCNN Deep Learning,” Int. J. Adv. Netw. Appl., vol. 14, no. 6, pp. 6076–6081, 2023.

[3] G. Kabanda, “Face Recognition in Machine Learning: A Framework for Dimensionality Reduction Algorithms,” Int. J. Adv. Netw. Appl., vol. 14, no. 2, pp. 5396–5407, 2022, DOI:10.35444/IJANA.2022.14211

[4] Y. Li, Y. Xie, and X. Lu, “Multi-face recognition and dynamic tracking based on reinforcement learning algorithm,” MATEC Web Conf., vol. 336, p. 06006, 2021, DOI: 10.1051/matecconf/202133606006

[5] N. Azatbekuly, A. Mukhanbet, and S. D. Bekele, “Development of an Intelligent Video Surveillance System Based on YOLO Algorithm,” in Proc. SIST 2024 - IEEE 4th Int. Conf. Smart Inf. Syst. Technol., 2024, pp. 498–503, DOI:10.1109/SIST61555.2024.10629617

[6] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, “Simple online and realtime tracking,” in Proc. IEEE Int. Conf. Image Process. (ICIP), Aug. 2016, pp. 3464–3468, DOI:10.1109/ICIP.2016.7533003.

[7] D. Jayaram, S. Vedagiri, and M. Ramachandra, “Framework for multiple person identification using YOLOv8 detector: a transfer learning approach,” Int. J. Electr. Comput. Eng., vol. 14, no. 3, pp. 2790–2802, 2024, DOI: http://doi.org/10.11591/ijece.v14i3.pp2790-2802

[8] A. Rajagopalan, “Real-Time Deep Learning-Based Face Recognition System,” Master's thesis, St. Cloud State University, St. Cloud, MN, USA, 2022.

[9] G. C. Suguna, H. S. Kavitha, and S. Sunita, “Face Recognition System for Real Time Applications Using Svm Combined With Facenet and Mtcnn,” Int. J. Electr. Eng. Technol., vol. 12, no. 6, pp. 328–335, 2021, **DOI:** 10.34218/IJEET.12.6.2021.031

[10] A. W. Majeed, S. H. Shaker, and A. A. Saeid, “A Real Time Face Recognition and Tracking Framework Using Lightweight Convolutional Neural Network,” BIO Web Conf., vol. 97, p. 00029, 2024, DOI:10.1051/bioconf/20249700029

[11] Y. Jiang and Z. Wang, “Real-Time Multiple Face Identity Recognition Based on Face-tracking Feature Extraction,” Journal of Network Intelligence, vol. 8, no. 2, 2023, pp. 626–638.

[12] M. Sri Durga, N. Leelavathy, K. Siva, B. V. N. Surya Prathap, K. Sudheer Babu, E. C. Alex Mondesir, Group Face Recognition: Identifying and Tracking Multiple Individuals, Proceedings of the International Conference on Computational Innovations and Emerging Trends (ICCIET 2024), Advances in Computer Science Research 112, pp.1209-1218, 2024, DOI:10.2991/978-94-6463-471-6_116

Figures

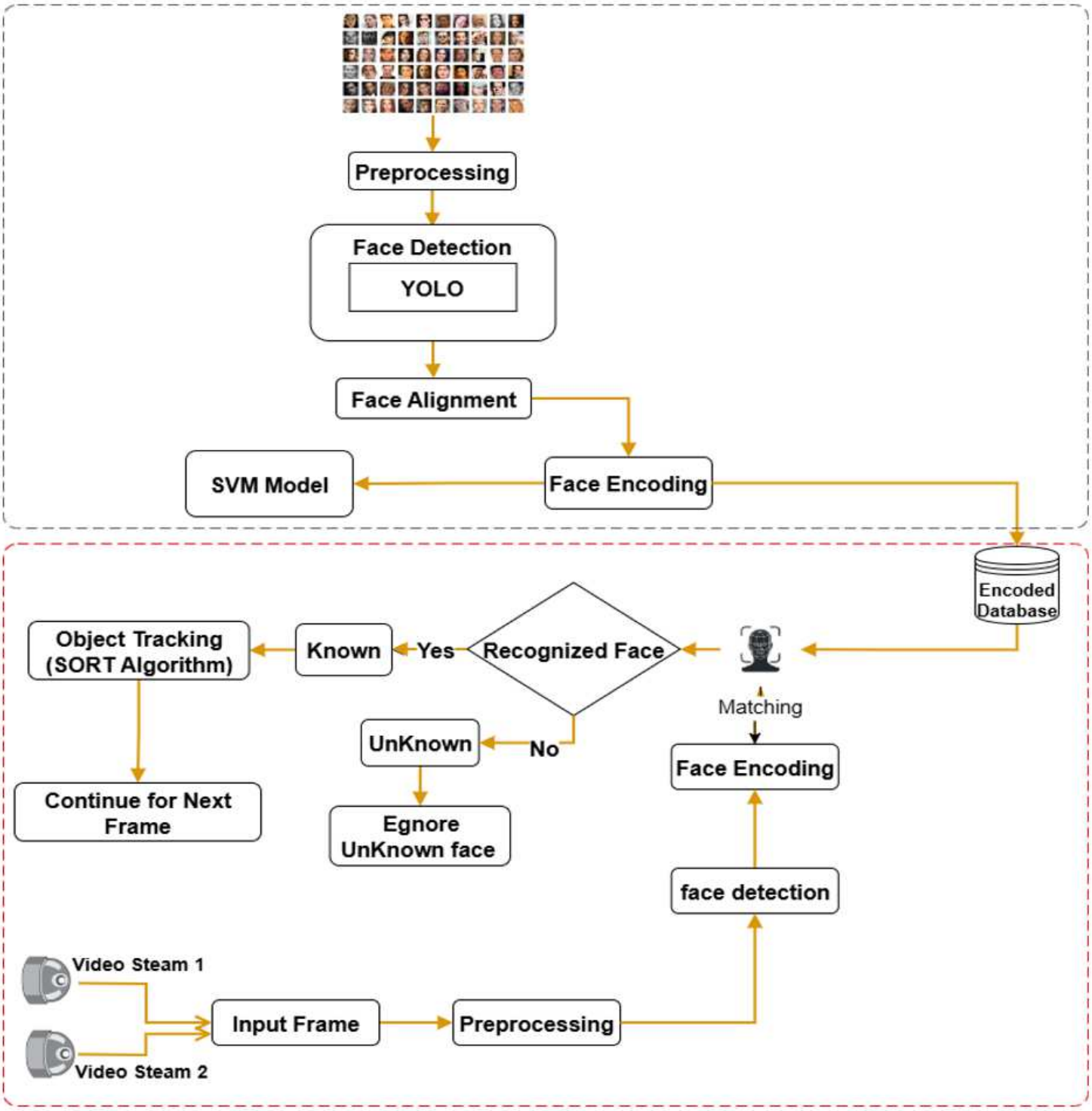


Figure 1

Overall Architecture of the Proposed Face Recognition System



Figure 2

Face Detection Process



Figure 3

All Alignment Process

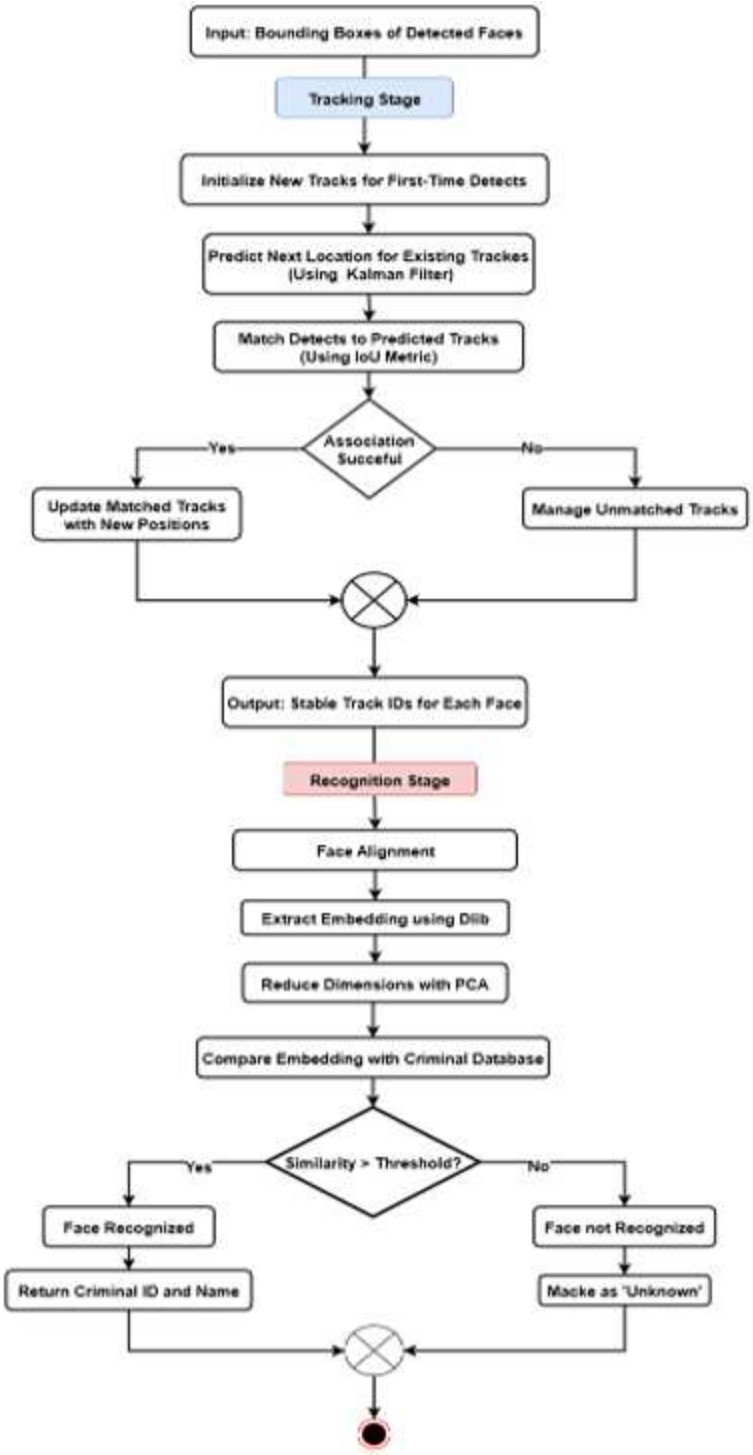


Figure 4

Tracking and Recognition Methodology

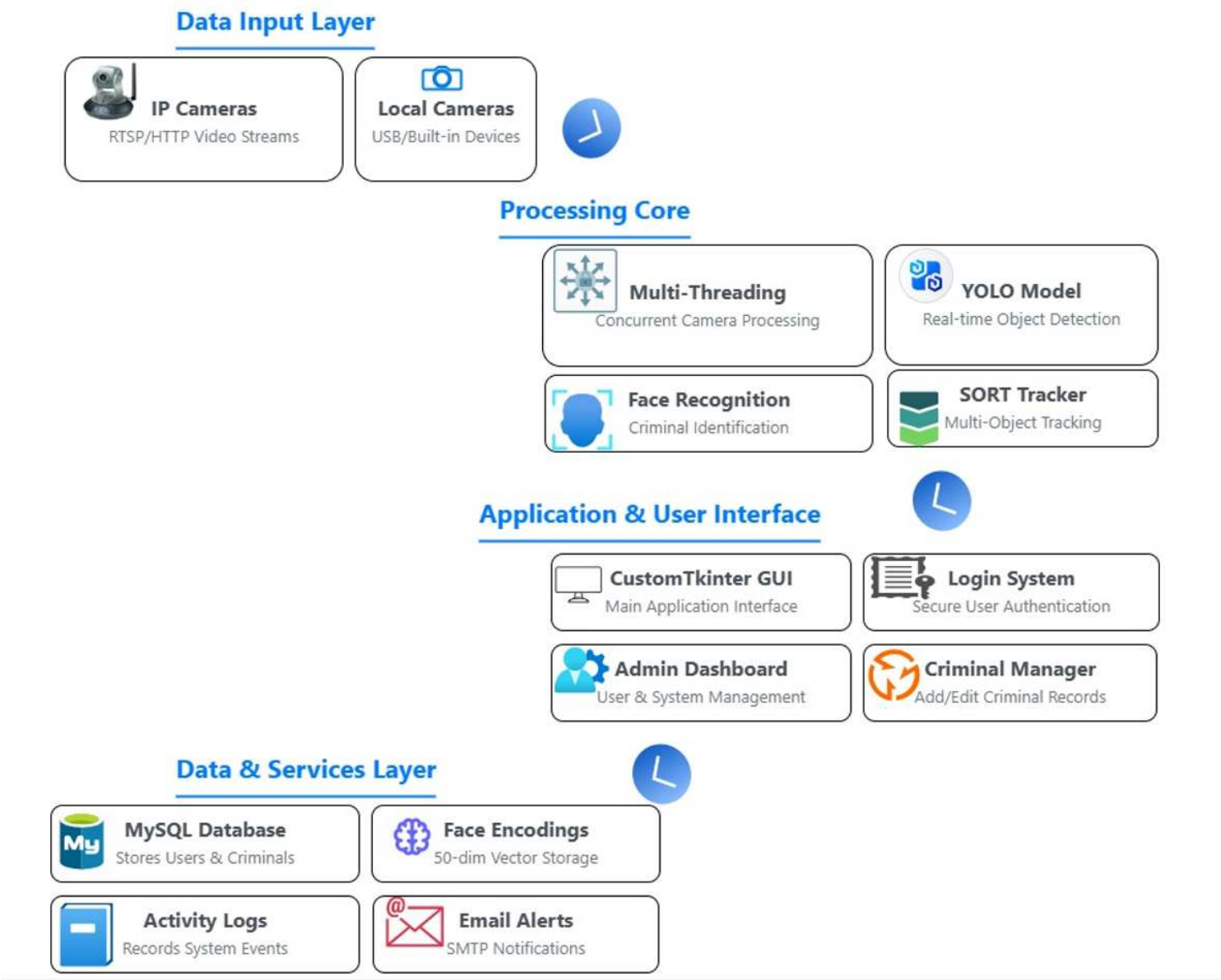


Figure 5

System Architecture

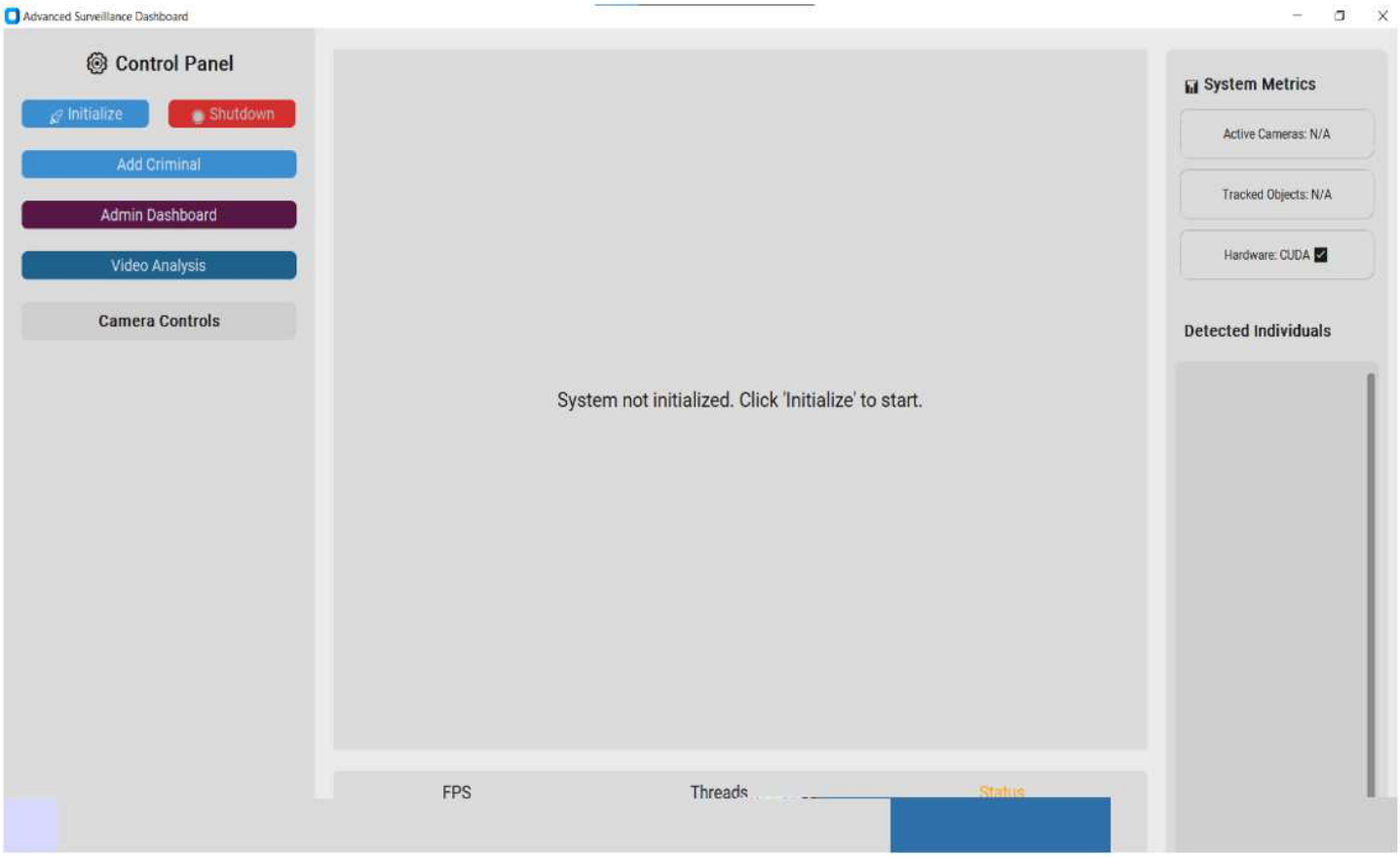


Figure 6

Main Dashboard

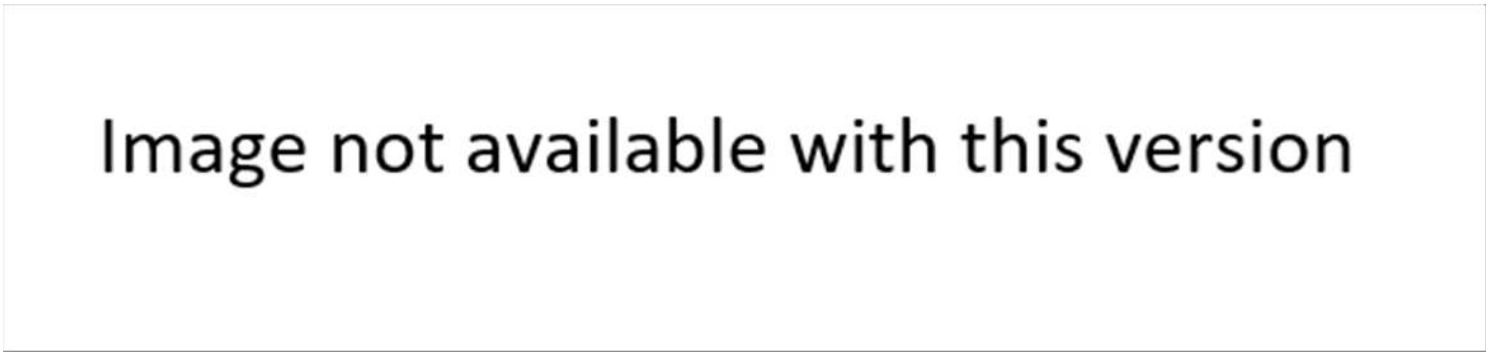


Figure 7

Add a criminal (suspect)

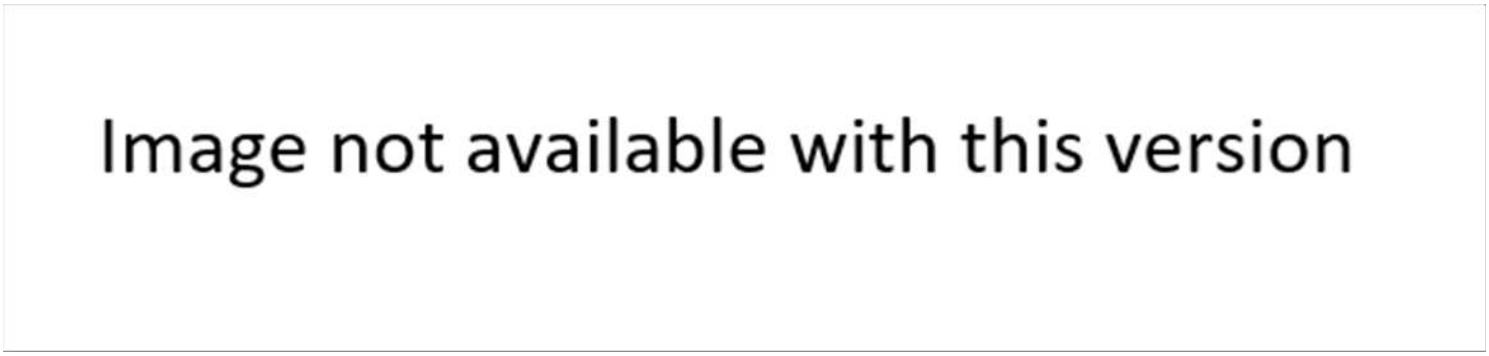


Figure 8

Criminal Management

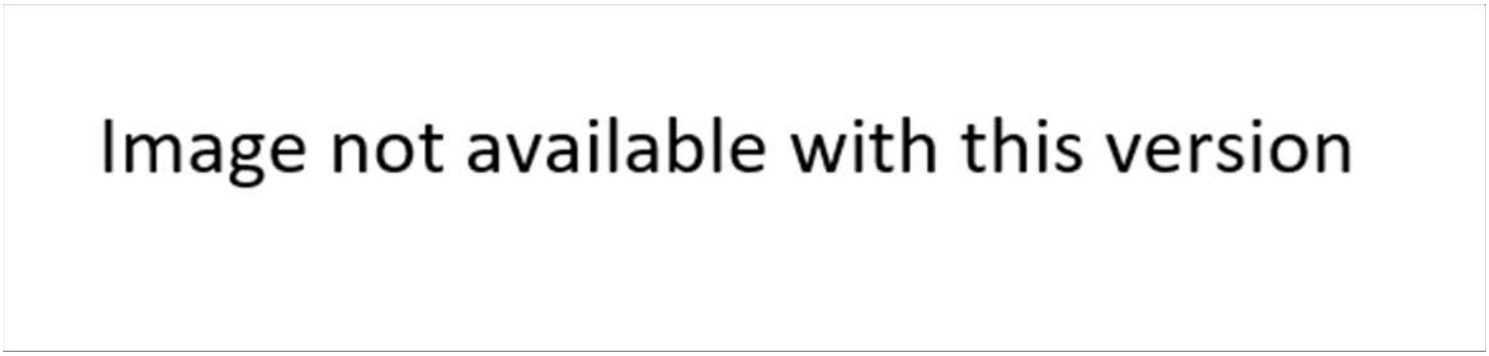


Figure 9

All Result in Validation Of YOLOv8s Model

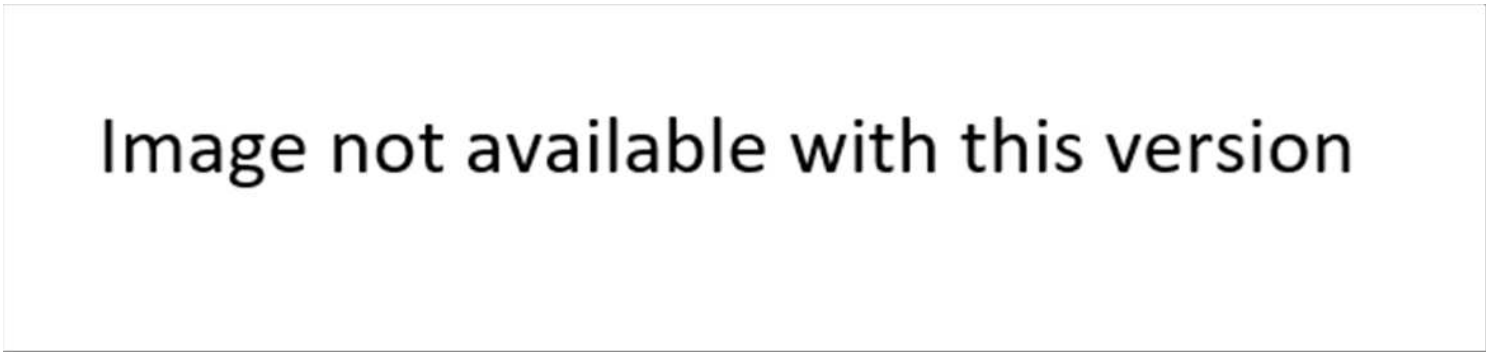


Figure 10

Confusion Matrix in Validation Task Of YOLOv8s Model

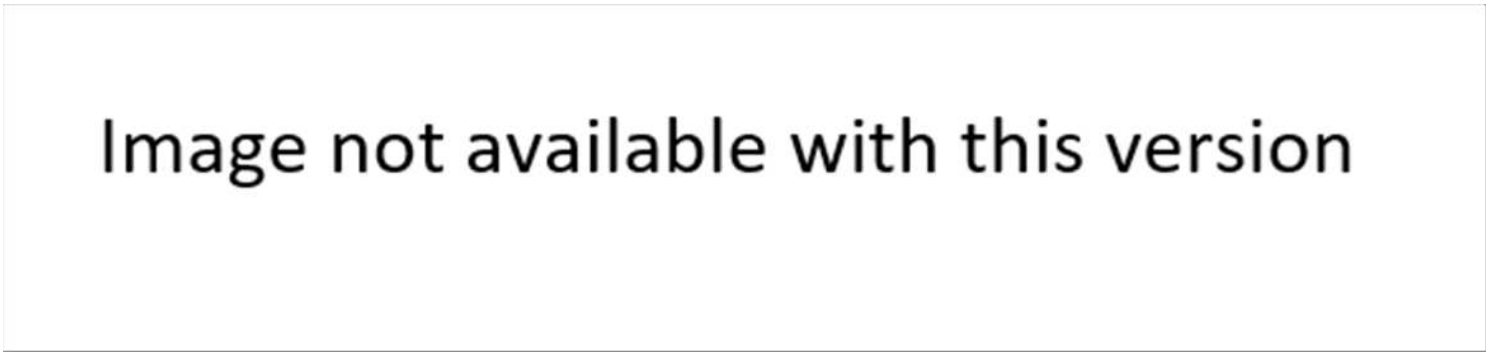


Figure 11

Result of monitoring for 2 Camera