

Person Activities Tracking and Detection Using Surveillance System

Mr. Vrushabh Digraje¹, Prof. Prasad Bhosle², Ms. Prajakta Patil³, Ms. Trupti Wavhal⁴,

Ms. Sushala Jagtap⁵

Department Of Computer Engineering, Trinity College Of Engineering And Research Pune, India.

Abstract - In modern institutions such as colleges, offices, airports, and public buildings, monitoring and locating individuals in real time has become a significant challenge. Traditional surveillance systems rely heavily on manual monitoring, which is inefficient, time-consuming, and prone to human error. This paper proposes an intelligent “Person Activities Tracking and Detection Using Surveillance System” capable of detecting, tracking, and identifying individuals using artificial intelligence, computer vision, and deep learning techniques. The proposed system integrates YOLOv8 for real-time person detection, DeepSORT for multi-object tracking, and InsightFace for facial recognition and identity verification. The framework allows authorized users to upload a reference image and continuously track the target individual across multiple live CCTV feeds in real time. The system also displays the current location and live video feed of the detected person. OpenCV-based video processing and RTSP camera streaming are used for seamless surveillance monitoring and real-time analysis. Experimental results demonstrate improved detection accuracy, stable identity preservation, and efficient tracking performance under varying environmental conditions such as occlusion, lighting variation, and crowded scenes. The proposed system enhances surveillance automation, improves security monitoring, reduces manual effort, and enables faster response during emergency situations.

Key Words: Artificial Intelligence, Computer Vision, Real-Time Surveillance, YOLOv8, DeepSORT, InsightFace, Face Recognition, Person Detection, Multi-Object Tracking, CCTV Monitoring, RTSP Streaming, Identity Verification, OpenCV, Deep Learning.

1. INTRODUCTION

The rapid increase in surveillance cameras across educational institutions, offices, transportation systems, and public places has generated a massive amount of real-time video data that requires continuous monitoring and analysis [1]. Traditional surveillance systems mainly function as recording devices and rely heavily on human operators to manually observe multiple CCTV feeds. This process is labor-intensive, inefficient, and highly prone to human error, especially in crowded or large-scale environments where identifying and locating a specific individual becomes difficult and time-consuming [2].

Recent advancements in artificial intelligence, computer vision, and deep learning have enabled the development of

intelligent surveillance frameworks capable of automatically detecting, tracking, and recognizing individuals in real time [3]. Several research studies have introduced approaches using SSD, YOLOv3, DeepSORT, Kalman Filter, HOG, CNN, and facial recognition techniques to improve surveillance automation and tracking accuracy [4][5]. Although existing systems have achieved significant progress in person detection and tracking, they still face challenges such as identity switching, occlusion handling, lighting variation, crowded scene instability, and high computational requirements, which affect their real-world performance and scalability [6].

To overcome these limitations, the proposed “Person Activities Tracking and Detection Using Surveillance System” integrates YOLOv8 for real-time person detection, DeepSORT for multi-object tracking, and InsightFace for face recognition and identity verification. The system allows authorized users to upload a reference image of a target individual and continuously track the person across multiple live CCTV feeds. OpenCV-based video processing and RTSP camera streaming are used for seamless real-time surveillance monitoring and location identification. The proposed framework enhances surveillance automation, improves security monitoring efficiency, reduces manual observation effort, and enables faster response during emergency situations.

2. Related Work

Several research studies have focused on developing intelligent surveillance systems for real-time person detection, tracking, and recognition using computer vision and deep learning techniques [1][2]. These systems aim to improve surveillance automation, reduce manual monitoring effort, and enhance security operations in crowded or sensitive environments.

Sercan Çağdaş Tekkök et al. developed a face detection, tracking, and recognition framework using Single Shot MultiBox Detector (SSD), SORT tracking algorithm, and deep learning-based face recognition techniques [1]. The system was designed to detect masked and unmasked faces in real time and maintain tracking using Kalman Filter-based prediction. Histogram of Oriented Gradients (HOG) and EigenFace methods were used for face recognition. The framework achieved efficient detection performance; however, the system experienced identity switching when individuals exited and re-entered the camera frame. In addition, the recognition accuracy decreased under poor lighting conditions and occlusion caused by masks or side-profile faces.

Another research work proposed an image-processing-based real-time human detection and tracking system using OpenCV, HOG feature extraction, CNN embeddings, facial landmark detection, and Support Vector Machine (SVM) classifiers [2]. The framework successfully identified individuals and maintained timestamp-based tracking information. Although the system improved person detection accuracy, it suffered from limitations such as sensitivity to illumination changes, pose variations, and crowded scene instability. The computational complexity of frame-by-frame analysis also reduced its scalability for large surveillance environments.

Research based on YOLOv3 and Kalman Filter algorithms introduced real-time face tracking systems capable of maintaining stable tracking under motion blur and temporary occlusion conditions [3][5]. YOLOv3 provided high-speed object detection, while the Kalman Filter predicted object trajectories across video frames. These systems achieved good tracking performance for linear motion; however, the framework struggled with nonlinear movement patterns and long-term occlusion handling. Furthermore, the absence of advanced appearance-based re-identification techniques reduced tracking stability in multi-person surveillance scenarios.

Several studies also implemented SSD and DeepSORT-based surveillance systems for crowd behavior analysis and person tracking across multiple camera feeds [5][6]. DeepSORT improved identity preservation using appearance descriptors and motion prediction techniques, while SSD enabled fast multi-scale object detection. These frameworks provided efficient crowd monitoring and trajectory analysis capabilities. However, the systems still faced challenges such as ID switching, tracking instability in dense crowds, low-light performance issues, and dependency on high-performance GPU hardware for real-time processing.

The analysis of existing surveillance systems indicates that although significant improvements have been achieved in object detection and person tracking, many frameworks still suffer from challenges related to identity preservation, lighting variation, crowded environments, occlusion handling, and computational efficiency. To overcome these limitations, the proposed "Person Activities Tracking and Detection Using Surveillance System" integrates YOLOv8 for real-time person detection, DeepSORT for multi-object tracking, and InsightFace for accurate face recognition and identity verification. The proposed framework aims to provide stable real-time tracking, improved recognition accuracy, and efficient surveillance automation across multiple CCTV camera feeds.

3. Methodology

A. RTSP Camera Streaming

RTSP (Real Time Streaming Protocol) is used for establishing communication between CCTV/IP cameras and the surveillance framework. It enables continuous live video streaming from multiple surveillance cameras for real-time

monitoring and analysis.[6] The captured surveillance feed acts as the primary input source for the framework.



Fig. 1 RTSP-based live CCTV camera streaming

B. OpenCV for Video Processing

OpenCV is used as the primary computer vision library for handling real-time video processing operations. It captures video frames from live CCTV streams and performs preprocessing operations such as resizing, normalization,[5] and color conversion. These preprocessing techniques improve image quality and optimize the input frames for deep learning models.

C. YOLOv8 for Person Detection

YOLOv8 (You Only Look Once Version 8) is used for real-time person detection within surveillance video streams. The model processes the complete image in a single forward pass and detects human objects by generating bounding boxes along with confidence scores. YOLOv8 Nano is selected because of its lightweight architecture, fast inference speed, and efficient detection performance suitable for real-time surveillance applications.[3]

The detection model identifies multiple individuals simultaneously and performs accurate person localization even in crowded environments.

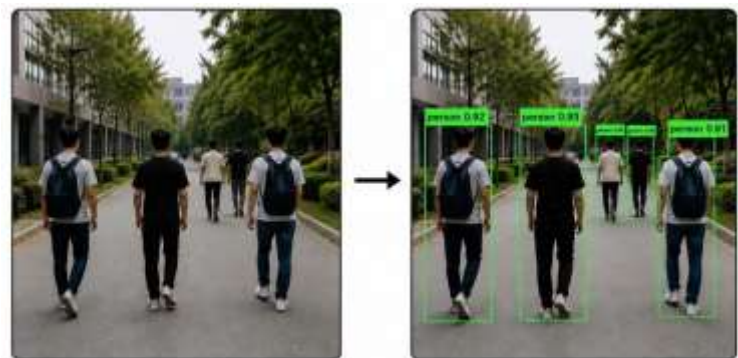


Fig. 2 YOLOv8-based real-time person detection

D. DeepSORT for Object Tracking

DeepSORT (Deep Simple Online and Realtime Tracking) is used for multi-object tracking and identity preservation across consecutive video frames. The tracking algorithm combines Kalman Filter-based motion prediction with appearance descriptor matching to maintain continuous tracking of detected individuals.[7]

Each detected person is assigned a unique tracking ID, which remains consistent during movement across surveillance frames.[3] DeepSORT improves tracking stability under partial occlusion, crowded scenes, and camera movement conditions.

E. InsightFace for Face Recognition

InsightFace and the FaceAnalysis model (buffalo_s) are used for facial recognition and identity verification. The framework extracts high-dimensional facial embeddings from detected face regions and compares them with uploaded reference image embeddings for person identification.[8]

The model supports front-face, side-profile, and partially visible face recognition using facial landmark alignment and deep embedding extraction techniques. This improves recognition accuracy under pose variation and lighting changes.



Fig. 3 InsightFace-based facial recognition

F. Cosine Similarity for Identity Verification

Cosine similarity is used for comparing facial embeddings extracted from uploaded reference images and live surveillance feeds. The similarity score determines whether the detected face matches the target individual stored in the database.[10]

Higher similarity values indicate stronger identity matching and improve recognition reliability within the surveillance framework.

4. Implementation

Step 1: Environment Setup

The implementation process begins with configuring the development environment using Python as the primary programming language. Visual Studio Code is used for developing and managing the surveillance framework.

FastAPI is implemented for backend processing and API communication, while React.js is used for frontend interface development and real-time monitoring visualization.

Several libraries are installed to support artificial intelligence and computer vision operations. OpenCV is used for video processing, PyTorch provides deep learning support, YOLOv8 is implemented for real-time person detection, InsightFace is used for facial recognition, and DeepSORT is integrated for multi-object tracking and identity preservation. NumPy and Pillow are used for image processing and numerical computations during surveillance analysis.

The framework supports both CPU and GPU-based execution. GPU acceleration improves model inference speed and enables efficient real-time surveillance processing across multiple camera feeds.

Step 2: RTSP Camera Connection

RTSP (Real Time Streaming Protocol) is implemented for establishing communication between CCTV/IP cameras and the surveillance framework. RTSP enables continuous live video streaming from surveillance cameras to the processing server in real time.

The framework supports multiple IP-based surveillance cameras such as CP PLUS and Panasonic cameras. OpenCV VideoCapture functions are used to establish RTSP connections and continuously receive surveillance frames from connected cameras.

The RTSP stream acts as the primary input source for the proposed framework. Continuous video transmission allows the system to monitor surveillance environments dynamically and process live video feeds without interruption.



Fig. 4 RTSP camera integration and live video streaming process

Step 3: Video Frame Acquisition using OpenCV

OpenCV is used for real-time video frame acquisition and preprocessing operations. The framework continuously captures surveillance frames from connected CCTV/IP

cameras and processes them before forwarding them to artificial intelligence models.

Several preprocessing techniques are implemented to improve image quality and optimize model performance:

- Frame resizing
- Image normalization
- Color space conversion
- Frame enhancement
- Noise reduction

These preprocessing operations reduce computational complexity and ensure compatibility between input frames and deep learning models such as YOLOv8 and InsightFace.



YOLOv8-based real-time person detection



Fig. 5 OpenCV-based video frame acquisition

Step 4: Person Detection using YOLOv8

YOLOv8 Nano is implemented as the primary object detection model for detecting persons within surveillance video streams. YOLOv8 is a deep convolutional neural network architecture capable of performing object localization and classification simultaneously in a single forward propagation step.

The model processes each surveillance frame and divides the image into multiple detection regions. It predicts bounding boxes, object confidence scores, and class probabilities for detected human objects. Non-Maximum Suppression (NMS) is applied to remove duplicate detections and retain only the highest-confidence bounding boxes.

YOLOv8 Nano is selected because of its lightweight architecture, fast inference speed, and low computational overhead suitable for real-time surveillance applications. The model performs accurate multi-person detection even under crowded environments, pose variation, and illumination changes.

Step 5: Multi-Object Tracking using DeepSORT

After person detection, DeepSORT (Deep Simple Online and Realtime Tracking) is implemented for continuous multi-object tracking and identity preservation across consecutive surveillance frames.

Each detected individual is assigned a unique tracking ID, which remains consistent during movement throughout the surveillance environment. The DeepSORT framework combines:

- Kalman Filter-based motion prediction
- Hungarian Algorithm for data association
- Deep appearance descriptor matching



Fig. 6 DeepSORT-based multi-object tracking implementation

The Kalman Filter predicts future object trajectories based on previous movement patterns, while the Hungarian Algorithm associates predicted tracks with newly detected objects. Appearance descriptors extracted from detected individuals improve tracking stability and reduce identity switching during:

- Crowded scenes
- Partial occlusion
- Camera movement
- Temporary disappearance

DeepSORT significantly improves long-term tracking consistency and enables continuous monitoring of individuals across multiple surveillance frames.

Step 6: Face Detection and Embedding Extraction

Face regions are extracted from detected persons using InsightFace and the FaceAnalysis model (buffalo_s). The framework performs face localization, facial landmark detection, face alignment, and embedding generation for recognition and identity verification.

Facial landmark alignment detects important facial points such as:

- Eyes
- Nose
- Mouth corners
- Jawline

The detected face is geometrically aligned to normalize head rotation and pose variation before feature extraction. This significantly improves recognition accuracy for:

- Frontal faces
- Side-profile faces
- Tilted faces
- Partially visible faces

After alignment, high-dimensional facial embeddings are generated using deep convolutional neural networks. These embeddings represent unique facial characteristics and are used for identity matching operations.



Fig.7 face detection and facial embedding extraction

Step 7: Reference Image Upload and Matching

Authorized users upload a reference image through the frontend monitoring interface. The uploaded image undergoes facial embedding extraction using the same InsightFace framework to maintain embedding consistency.

The framework compares reference image embeddings with live surveillance embeddings generated from detected faces. The comparison process helps identify whether the detected

surveillance face corresponds to the uploaded target individual.

This mechanism enables automated person identification and real-time tracking within surveillance environments.



Fig. 8 Reference image upload and facial embedding

Step 8: Identity Verification using Cosine Similarity

Cosine similarity is implemented for comparing facial embeddings generated from uploaded reference images and live surveillance feeds. The similarity function calculates angular similarity between embedding vectors and generates a similarity confidence score.

Higher similarity values indicate stronger identity correlation between:

- Uploaded reference image
- Live-detected surveillance face

Threshold-based similarity comparison is used to verify identity matching accurately. Cosine similarity improves recognition robustness and reduces false-positive identification during surveillance operations.



Camera	Similarity	Time
College Camera 1	0.97	12:07:48
College Camera 1	0.98	12:07:49
College Camera 1	0.98	12:07:50
College Camera 1	0.98	12:07:55
College Camera 1	0.97	12:07:56
College Camera 1	0.97	12:08:01
College Camera 1	0.97	12:08:06
College Camera 1	0.97	12:08:10
College Camera 1	0.98	12:08:14

Fig. 9 Similarity-based identity verification

5. Conclusion

The proposed intelligent surveillance framework successfully integrates artificial intelligence, deep learning, and computer vision techniques for real-time person detection, tracking, and recognition using CCTV/IP camera feeds. The framework utilizes YOLOv8 for accurate person detection, DeepSORT for continuous multi-object tracking, InsightFace for facial recognition, and cosine similarity for identity verification, while OpenCV and RTSP streaming enable efficient real-time video processing and surveillance monitoring. The implementation results demonstrate that the system can accurately detect and track multiple individuals simultaneously while maintaining stable identity preservation across consecutive video frames. The facial recognition module effectively identifies target individuals using uploaded reference images and supports frontal-face, side-profile, and partially visible face recognition under varying environmental conditions. The framework significantly reduces manual monitoring effort, improves surveillance automation, and enhances security operations in environments such as colleges, offices, airports, transportation systems, and public places. Although the system performs efficiently in real-time conditions, factors such as extreme occlusion, low-resolution surveillance feeds, and poor illumination may slightly affect recognition accuracy and tracking stability. Future enhancements may include edge AI deployment, cloud-based surveillance integration, low-light optimization, and advanced behavior analysis techniques for improving large-scale intelligent surveillance applications.

REFERENCES

- [1] S. C. Tekkök, M. E. Soyunmez, B. Bostancı, and P. O. Ekim, "Face Detection, Tracking and Recognition with Artificial Intelligence," IEEE, 2021.
- [2] Saket Ranjan, Sarthak Tyagi, Shubhi Gupta, Mandeep Kaur, and Mayank Kumar Goyal, "Image Processing-Based Real-Time Detection and Tracking of Human," 2023 3rd International Conference on Technological Advancements in Computational Sciences (ICTACS), IEEE, 2023.
- [3] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," arXiv preprint arXiv:1804.02767, 2018.
- [4] Glenn Jocher et al., "YOLOv8 by Ultralytics," Ultralytics, 2023.
- [5] N. Wojke, A. Bewley, and D. Paulus, "Simple Online and Realtime Tracking with a Deep Association Metric," IEEE International Conference on Image Processing (ICIP), 2017.
- [6] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "ArcFace: Additive Angular Margin Loss for Deep Face Recognition," Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019.
- [7] Jiankang Deng et al., "RetinaFace: Single-stage Dense Face Localisation in the Wild," arXiv preprint arXiv:1905.00641, 2020.
- [8] Guo, Jia, et al., "InsightFace: 2D and 3D Face Analysis Project," GitHub Repository, 2022.
- [9] G. Bradski, "The OpenCV Library," Dr. Dobb's Journal of Software Tools, 2000.
- [10] A. Paszke et al., "PyTorch: An Imperative Style, High-Performance Deep Learning Library," Advances in Neural Information Processing Systems (NeurIPS), 2019.