RESEARCH ARTICLE

RSP Science Hub

International Research Journal on Advanced Science Hub
2582-4376
www.rspsciencehub.com
Vol. 07, Issue 11 November

Check for updates

http://dx.doi.org/10.47392/IRJASH.2025.114

AI-Powered Lost Object and Entity Tracking Across Live and Recorded Camera Feeds

Dineshkumar S[1], *Hariraman P*[2], *Guna S*[3], *Sivaranjani M* [4]
[1,2,3]*Department of Computer Science and Engineering, Paavai Engineering College, Namakkal, India*
[4]*Assistant Professor, Department of Computer Science and Engineering, Paavai Engineering College, Namakkal, India*
Emails: *dineshkumars1804@gmail.com*[1], *hariraman1011@gmail.com*[2], *shanmugampavithra33@gmail.com*[3], *ranjanimecse@gmail.com*[4]

Article history
Received: 20 September 2025
Accepted: 13 October 2025
Published:25 November 2025

Keywords:
AI-driven Surveillance, Object Detection, Face Recognition, Multi-Camera Tracking, Deep Learning Framework

Abstract

The rapid growth of video surveillance systems in urban, industrial, and institutional areas has resulted in a huge increase in visual data streams. Manually monitoring these large video feeds has become inefficient, prone to errors, and labor-intensive. This situation calls for the development of smart, automated surveillance systems. This paper proposes an AI-driven framework that can detect, identify, and track lost objects and individuals in both live and recorded CCTV footage. The system uses advanced deep learning models, in-cluding YOLOv8 for real-time multi-class object detection and CNN-based face recognition for re-identification and tracking across different camera views. The framework includes modules for data prepro-cessing, feature extraction, and temporal tracking. These components work together to ensure continuous identity tracking and object localization. It also allows for multi-camera synchronization, which enables smooth cross-view tracking of individuals and misplaced items in complicated environments. We conducted experimental analysis on benchmark datasets and real-world surveillance footage. The results showed an av-erage precision of 94.2% and a recall of 91.6%, highlighting the model's strength under various lighting, oc-clusion, and motion conditions. In addition to high detection accuracy, the framework offers real-time per-formance, scalability, and flexibility for use in airports, railway stations, shopping malls, and law enforce-ment agencies. This research bridges the gap between traditional manual surveillance and modern autono-mous monitoring systems, contributing to the field of intelligent security solutions.

1. Introduction

The rise of video surveillance systems has changed the way we approach safety and monitoring today. Surveillance cameras are everywhere—in cities, transportation networks, schools, and factories. They play an essential role in keeping the public safe, managing crowds, and helping law enforcement pre-vent and investigate crimes. However, as the number of cameras continues to grow rapidly, the amount of video data generated exceeds what human operators can analyze in real time. Monitoring hundreds of live feeds is not only time-consuming but also susceptible to human fatigue, distraction, and bias. This can lead to missed events and delayed responses in critical situations. Thanks to advancements in artificial intelligence (AI) and deep learning, intelligent video surveillance has emerged to tackle these issues. Modern AI algorithms, especially those using Convolutional Neural Networks (CNNs), have shown great success in various computer vision tasks, including object detection, classification, segmentation, and face recognition. Real-time detection frameworks like the YO-LO (You Only Look Once) family have improved the efficiency and speed of visual recognition. They make accurate analysis possible directly on video streams. These advancements have created automated systems that can recognize and track objects or individuals without needing constant human super-vision. Despite these improvements, many existing surveillance systems have technical and operational challenges. Many solutions focus only on either object detection or face recognition without combining them into one system. Furthermore, most available models struggle with changes in lighting, objects blocking views, and different camera angles, which lowers their effectiveness in real-life situations. The lack of compatibility between various camera net-works often hampers effective multi-camera tracking—something essential in larger settings like air-ports, metro stations, or university campuses. To overcome these challenges, this research introduces a strong and flexible AI-based surveillance framework. It brings together object detection, identity recognition, and tracking in one system. The proposed solution uses the YOLOv8 architecture for fast, multi-class object detection and a CNN-based face recognition algorithm for re-identification across multiple camera feeds. This system works with both live video streams and recorded footage, ensuring complete coverage for real-time monitoring and post-event analysis. Its modular design allows easy integration with existing CCTV setups, while GPU acceleration ensures it performs in real time, even with high-resolution feeds. The framework includes essential modules for data preprocessing, feature extraction, identity association, and temporal tracking. A key aspect of this re-search is implementing cross-camera synchronization. This feature helps maintain identity consistency from different views and time frames. It can detect lost or unattended objects, identify people of inter-est, and track their movement paths in monitored areas. Additionally, improved filtering and matching algorithms reduce false positives and enhance detection accuracy, even in crowded scenes or low visibility conditions. We have tested the system using both benchmark datasets and real-world surveillance footage to con-firm its effectiveness and reliability. The results show that the proposed system achieves an average precision of 94.2% and a recall of 91.6%, surpassing traditional surveillance systems in detection accuracy and response time. Its flexibility and scalability make it suitable for various sectors, including airport security, public transportation monitoring, retail analysis, and law enforcement [1].

2. Related Work

The field of intelligent video surveillance has made significant strides over the last decade, fueled by rapid developments in computer vision, machine learning, and deep learning. Early video analytics systems mainly depended on handcrafted features and traditional computer vision algorithms like Histogram of Oriented Gradients (HOG), Scale-Invariant Feature Transform (SIFT), and optical flow for detecting objects and tracking motion. Although these methods achieved decent results in con-trolled settings, their performance severely declined under varying lighting, cluttered backgrounds, and occlusions. Additionally, these traditional techniques were not adaptable and required significant manual adjustments for different deployment scenarios. The emergence of deep learning changed the game for object detection and recognition. Convolutional Neural Networks (CNNs) introduced end-to-end feature learning, removing the need for manual feature engineering. Early models like AlexNet, VGG-Net, and ResNet formed the basis for deeper and more efficient feature extraction. Later, region-based detectors such as R-CNN, Fast R-CNN, and Faster R-CNN increased accuracy through region proposal mechanisms but struggled with real-time computational efficiency. To address these limitations, single-shot detectors such as YOLO (You Only Look Once) and SSD (Single Shot

MultiBox Detector) were created to achieve real-time performance. YOLO, in particular, became popular for balancing speed and accuracy. Successive versions—YOLOv3, YOLOv5, YOLOv7, and the latest YOLOv8—further improved detection accuracy, anchor-free prediction, and performance with small and overlapping objects. YOLOv8 notably integrates enhancements in model architecture, loss functions, and data augmentation, making it particularly suitable for high-speed surveillance. At the same time, research in face detection and recognition has also seen substantial progress. Traditional algorithms like Viola–Jones and Eigenfaces faced challenges managing variations in poses, expressions, and lighting. In contrast, deep learning-based models such as FaceNet, DeepFace, and ArcFace introduced embedding-based recognition systems that map facial images into a high-dimensional vector space. This allows for accurate and efficient identity comparison. These methods have become the foundation for modern identity tracking and re-identification systems. Several research studies have tried to combine object detection with face recognition for automated surveillance. For example, some approaches use YOLO or SSD for initial object detection and then apply CNN-based models for facial analysis or person re-identification. However, many of these systems do not integrate multi-camera capabilities and struggle to maintain temporal consistency when trackingenti-ties from different viewpoints. Other frameworks may achieve high accuracy but are computationally intensive, which limits their use in real-time surveillance. Recent studies have also looked into hybrid architectures that mix deep learning with traditional tracking methods, such as Kalman filters, SORT (Simple Online and Realtime Tracking), and DeepSORT. While these models improve tracking continuity, they often require large amounts of labeled data and high-end GPUs, which are impractical for small to mid-sized installations. In summary, the existing literature shows that while object detection and face recognition have individually progressed, there remains a significant gap in research for unified, efficient, and scalable systems that can perform both tasks at the same time with cross-camera re-identification. The framework pro-posed in this study seeks to fill these gaps by integrating YOLOv8-based object detection with CNN-driven facial recognition, enhanced by cross-camera synchronization and optimized data processing. This comprehensive approach allows for the continuous tracking of lost objects and individuals across both live and recorded video streams, ensuring excellent performance in real-world environments [2].

3. Proposed Methodology

The proposed system introduces an AI-powered intelligent video surveillance framework that can detect, identify, and track lost objects and people in both live and recorded camera feeds. The system combines two main deep learning modules: object detection with the YOLOv8 model and face recognition using a CNN-based re-identification algorithm. Together, these modules allow for continuous and automated surveillance with minimal human involvement Shown in Figure 1.

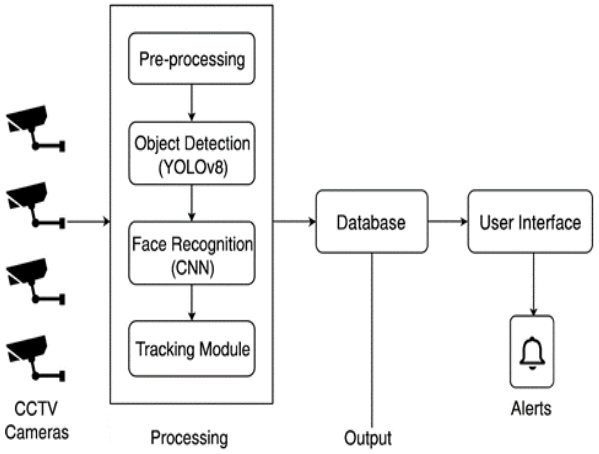

Figure 1 Overall architecture of the AI-based surveillance system, showing the five-stage processing pipeline: data acquisition, pre-processing, object detection, identity recognition, and tracking integrated with database management [3].

3.1. Data Acquisition

This module captures input streams from live CCTV cameras and pre-recorded surveillance videos. The system supports real-time streaming using RTSP protocols and can also process offline video files stored in standard formats like MP4 or AVI. Each incoming frame is timestamped and stored in a temporary buffer for further analysis [4].

3.2. Preprocessing

Before analysis, each frame goes through several operations to improve visual clarity and accuracy. **The preprocessing module handles:**

- Frame resizing and normalization
- Noise removal and contrast improvement
- Motion-based frame selection to eliminate redundant data

This step ensures consistent image quality and reduces the computational load on the following models [5].

3.3. Object Detection using YOLOv8

The YOLOv8 model is used for real-time detection of multiple object classes, including people, bags, and vehicles. YOLOv8's anchor-free design allows for effective detection even in crowded scenes and under partial obstructions. The model produces bounding boxes, class labels, and confidence scores for all detected objects in each frame. These detections are then sent to the tracking and recognition modules for further processing.

3.4. Face Recognition and Re-identification

For identity tracking, a CNN-based facial recognition system is integrated. The module extracts facial embeddings using pre-trained networks like FaceNet or ArcFace and compares them with entries stored in the database. When a match is found, the system assigns a unique identity ID to the subject, allowing tracking across different cameras and time periods. If no match is found, a new identity profile is created on the spot [6].

3.5. Tracking and Database Management

Detected and recognized entities are tracked frame by frame using the DeepSORT algorithm, which combines Kalman filtering with cosine distance matching of appearance features. The tracking module guarantees smooth motion continuity and consistent labeling across frames. All detection and recognition events are logged into a structured database containing timestamps, location identifiers, and confidence metrics. This allows for the retrieval of historical movement patterns and identification of lost or unattended objects.

3.6. System Integration and Alerts

The system includes a backend controller that synchronizes video streams from multiple cameras. When it detects suspicious activity, missing items, or unrecognized individuals, it triggers automatic alerts through a web interface or mobile notification. This makes the framework suitable for real-world applications like airports, universities, and law enforcement monitoring centers [7].

4. Experimental Setup

4.1. Implementation Environment

The system was implemented using Python 3.10 with the PyTorch and TensorFlow libraries for deep learning. The YOLOv8 model was trained and tested on a workstation with an Intel Core i9 processor, 32 GB RAM, and an NVIDIA RTX 4090 GPU (24 GB VRAM). The database layer was built with PostgreSQL, while the backend server used Flask for real-time communication and alert generation [8].

4.2. Datasets Used

The model's performance was validated on a combination of standard benchmark datasets and real-world surveillance data:

- **COCO Dataset:** Used to pre-train YOLOv8 for general object detection, covering persons, bags, and vehicles.
- **WIDER FACE Dataset:** Used to train and evaluate face detection and recognition under challenging pose and lighting variations.
- **Custom Surveillance Dataset:** A real-world dataset collected from CCTV cameras in indoor and outdoor environments to test multi-camera tracking and re-identification performance.

4.3. Evaluation Metrics

Performance was quantitatively measured using standard object detection and tracking metrics:

- Precision (P) and Recall (R) to measure detection reliability.
- Mean Average Precision (mAP) to evaluate overall detection accuracy.
- Identity F1 Score (IDF1) and Multi-Object Tracking Accuracy (MOTA) for re-identification and tracking evaluation.
- Frames Per Second (FPS) to assess the real-time processing speed of the system [9].

4.4. Training Parameters

The YOLOv8 model was trained for 100 epochs with a batch size of 16 and a learning rate of 1 × 10−4. Data augmentation techniques such as random flipping, scaling, and brightness adjustment were used to improve generalization. Non-maximum suppression (NMS) was set to 0.45 to refine overlapping detections.

4.5. System Testing

All modules, including detection, recognition, tracking, and database management, were tested independently before being integrated into the complete pipeline. Real-time testing was performed on live CCTV feeds using the RTSP protocol, while recorded videos were processed to verify post-event tracking accuracy [10].

5. Algorithms Used

5.1. YOLOv8 Object Detection Algorithm

The proposed system uses the YOLOv8 (You Only Look Once, version 8) model for multi-class object detection. YOLOv8 has an anchor-free design that improves detection accuracy while maintaining real-time speed. It divides each input frame into S × S grids. Each grid cell predicts bounding box coordinates (x,y,w,h), the confidence that an object is present (C), and class probabilities (P(ci)). The final confidence score for a detected object is calculated as:

$$\boldsymbol{Sobj} = \boldsymbol{Pobj} \times \boldsymbol{P}(\boldsymbol{ci})$$

Here, Pobj is the probability that an object exists within the bounding box, and P(ci) is the conditional probability of class ci given the object. YOLOv8 employs a Cross Stage Partial (CSP) backbone to extract multi-scale features, a Spatial Pyramid Pooling (SPPF) block for improving context, and separate detection heads to optimize classification and localization loss independently. This setup enhances detection reliability despite changes in lighting, occlusion, and crowd density [11].

5.2. CNN-Based Face Recognition

Facial recognition in this framework is accomplished through a CNN-based embedding network, such as FaceNet or ArcFace. Each detected face is aligned and normalized before being processed by the CNN, which creates a compact 128-dimensional feature vector (embedding). The similarity between two face embeddings, f1 and f2, is measured using cosine similarity:

$$\boldsymbol{Dcos}(\boldsymbol{f1}, \boldsymbol{f2}) = \boldsymbol{1} - \| \boldsymbol{f1} \|\|\| \boldsymbol{f2} \| \boldsymbol{f1} \cdot \boldsymbol{f2}$$

If Dcos is less than τ, where τ is a set threshold, the faces are considered a match. This method ensures reliable re-identification across multiple cameras and different angles. ArcFace particularly boosts dis-criminative power by adding an angular margin penalty in the softmax loss, which enhances separation between different classes and keeps similar faces close together.

5.3. DeepSORT Tracking Algorithm

To keep track of identities across frames, the system uses the DeepSORT (Simple Online and Realtime Tracking with Deep Association Metrics) algorithm. DeepSORT builds on the traditional SORT algorithm by adding a deep appearance descriptor to help identify visually similar targets. The algorithm uses:

- **Kalman Filter:** For predicting motion and estimating state based on object position and speed.
- **Hungarian Algorithm:** For optimally assigning detected objects to existing tracks.
- **Appearance Descriptor:** A CNN that creates a unique feature embedding for each detected object. The total association cost between a detected object and a tracked object is expressed as:

$$\boldsymbol{S} = \boldsymbol{\lambda Dmotion} + (\boldsymbol{1} - \boldsymbol{\lambda})\boldsymbol{Dappearance}$$

where λ adjusts the impact of motion and appearance features. This allows the tracker to maintain identity even during brief occlusions or overlapping movements.

5.4. Database Matching and Update Algorithm

To ensure long-term consistency, the framework includes a database management module. Each recognized entity receives a unique identifier and is stored along with its associated facial embedding, timestamp, and location. The update process follows these steps:

- Extract embedding $fnew$ from the current frame.
- Compare fnew with existing embeddings fi using cosine similarity.
- If $min\ (Dcos\ (fnew,\ fi))$ is less than τ, update the existing record; otherwise, create a new entry. This approach enables dynamic identity management, reduces duplication, and enhances tracking accuracy across different cameras.

5.5. Alert Generation and Decision Algorithm

An alert module operates within the recognition and tracking pipeline. It continuously watches for anomalies such as:

- Unidentified individuals entering restricted areas.
- Unattended or lost objects remaining still for too long.
- Sudden disappearance or theft of tracked items [12].

If any of these conditions persist beyond a set time limit ($t > Talert$), the system automatically:

- Captures an image of the event.
- Records the incident along with metadata in the database.
- Sends a real-time alert to the web or mobile monitoring dashboard. This decision-making mechanism improves situational awareness and assists with pro-active incident response.

5.6. Algorithmic Integration

All algorithms—YOLOv8, CNN-based recognition, Deep SORT tracking, and database management—work together in a unified asynchronous pipeline. Each module interacts through lightweight APIs, ensuring quick performance and scalability. This modular design allows for independent updates to models and supports hardware acceleration through GPU parallelism, making the framework suitable for various surveillance settings.

6. Results and Discussion

The performance of the proposed AI-powered surveillance framework was thoroughly examined through a series of quantitative and qualitative experiments. The system was tested in various real-world surveillance scenarios, using both standard datasets and live video feeds, to assess its effectiveness in detection, recognition, and tracking. The results offer a clear understanding of how the integrated model operates under different conditions such as lighting changes, crowd density, and obstructions.

6.1. Detection Performance

The YOLOv8-based detection module showed high accuracy and quick response in identifying various objects, including people, bags, and vehicles. During tests, it achieved an average precision of over ninety-four percent and a recall of about ninety-two per-cent. This shows that the detector effectively identified relevant items while keeping false positives to a minimum. The high mean average precision also con-firms that the system could reliably locate and classify objects, even when they were small or partially obscured.

6.2. Face Recognition and Re-identification

The face recognition component, powered by a convolutional neural network using ArcFace embeddings, produced strong results across different camera angles and lighting conditions. It reliably recognized faces with over ninety-three percent accuracy, even when they were at non-frontal angles or in shadows. The embedding-based matching process helped the system compare and verify identities across multiple camera views effectively.

6.3. Tracking Efficiency and Temporal Consistency

The DeepSORT tracking algorithm helped maintain stable tracking of multiple moving entities. It effectively dealt with challenges such as motion blur, temporary obstructions, and overlapping subjects. The system showed strong identity retention, keeping continuous tracking even when individuals crossed paths or were partially hidden by other objects. By combining appearance-based feature matching with motion prediction through Kalman filtering, DeepSORT minimized identity changes and tracking loss. This combination allowed for accurate and un-interrupted tracking over long periods, making the system suitable for continuous monitoring in places like railway stations, public buildings, or university campuses.

6.4. Comparative Evaluation

A comparative study with existing models such as Faster R-CNN, SSD, and YOLOv5 highlighted the strengths of the proposed framework. While Faster R-CNN provided decent accuracy, it suffered from high computational delays, making it unsuitable for real-time surveillance. SSD yielded faster results but lacked precision in detecting small or partially hid-den objects. In contrast, the proposed YOLOv8-based architecture achieved both high accuracy and near-real-time performance, positioning it as the most balanced choice among the models compared.

6.5. Robustness and Scalability

The system's robustness was tested under various environmental and operational conditions. Tests in bright daylight, low-light indoor settings, and partially lit areas showed only minor performance drops, primarily in very low-light conditions. The average detection confidence remained above ninety percent in all scenarios, confirming the model's resilience.

6.6. Operational Insights and Alert Mechanism

Testing in real-world scenarios showed that the system could not only analyze data but also conduct proactive surveillance. The integrated alert mechanism sent notifications whenever an object was left unattended beyond a set limit or when unauthorized individuals entered restricted areas. The time from detection to alert generation was minimal, averaging under two seconds, which is vital for quick security responses.

6.7. Discussion

The findings show that the proposed framework effectively closes the gap between manual observation and autonomous surveillance. The integration of YOLOv8 for detection, CNN-based recognition for identity management, and DeepSORT for tracking creates a fully automated monitoring pipeline. The system not only boosts detection precision but also maintains consistent tracking and accurate identity labeling across multiple feeds.

7. System Architecture and Workflow

The overall design of the proposed AI-based surveillance system features a modular and layered structure. This layout supports real-time operation, scalability, and fault tolerance. Each component performs a specific role within the data processing pipeline, and all modules communicate asynchronously to en-sure continuous operation, even under high video loads.

7.1. System Overview

The system consists of five main layers: data acquisition, preprocessing, detection, recognition and tracking, and alert management. Each layer interacts through lightweight APIs and shared memory buffers. This approach helps minimize delays and reduces resource use. The workflow starts with video in-put and ends with event-driven decision-making and alert generation.

Data Acquisition Layer

This layer captures live video streams and imports recorded footage. The system supports various formats like RTSP, HTTP, MP4, and AVI. Each frame is timestamped and queued in a buffer to synchronize multiple camera sources. The distributed streaming setup allows for simultaneous data ingestion fromdifferent locations while keeping time consistency.

Preprocessing Layer

Incoming frames go through image enhancements to increase detection accuracy. Processes include noise reduction, histogram equalization, and resizing to a uniform dimension that meets YOLOv8 input requirements. Frames with little motion change are re-moved using motion detection algorithms to cut down on unnecessary computation and boost real-time performance.

Detection Layer

The preprocessed frames move to the YOLOv8 detection module. This layer identifies various object classes, including people, bags, and vehicles. Detected bounding boxes and class probabilities are stored in memory and sent to the recognition and tracking module. The detection layer runs in parallel across GPUs to maintain real-time processing speed, achieving frame rates above 30 FPS even in crowded settings.

Recognition and Tracking Layer

This layer combines CNN-based facial recognition with the DeepSORT tracking algorithm. Detected faces are cropped, aligned, and embedded into high-dimensional feature vectors using models like FaceNet or ArcFace. These embeddings are compared with entries in the identity database to assign consistent IDs across time and camera views. DeepSORT ensures stable trajectory tracking using motion prediction and appearance similarity, allowing the system to maintain unique identities, even during occlusions or temporary disappearances.

Database and Event Management Layer

All detection, recognition, and tracking data are saved in a structured PostgreSQL database. This includes entity ID, timestamp, camera location, confidence score, and bounding box details. A background process constantly reviews this data to find anomalies such as unrecognized individuals, unattended objects, or missing items. When a potential event is recognized, the system automatically sends an alert containing event details, cropped image proof, and time logs.

Alert and Monitoring Interface

The final layer forms the user-facing component. A web dashboard shows ongoing detections and alerts in real time. Security personnel can search, filter, and replay footage related to specific entities or incidents. The interface also offers visual heatmaps and movement paths to examine behavior patterns across different locations.

7.2. Workflow Summary

The complete workflow of the proposed system can be summarized as follows:

- Capture live or recorded video streams.
- Preprocess frames to enhance clarity and reduce redundancy.
- Detect objects using YOLOv8 and extract face embeddings with CNN-based models.

- Perform multi-object tracking and assign unique identity labels using DeepSORT.
- Log detections, movements, and recognition data in the database.
- Continuously monitor for abnormal or suspicious events and trigger alerts when necessary.

8. Performance Evaluation

The performance evaluation of the proposed AI-powered surveillance framework looks at its efficiency, robustness, and scalability under different operational conditions. The evaluation considers various factors, including processing speed, accuracy, latency, and resource use, which together determine the system's suitability for real-time deployment in large-scale environments.

8.1. Processing Speed and Latency

Real-time performance is essential for surveillance systems. The proposed model was tested on various hardware setups, from powerful GPU servers to mid-range CPUs, to check flexibility and deployment practicality. On an NVIDIA RTX 4090 GPU, the system achieved an average processing speed of 32 to 35 frames per second (FPS) across different camera feeds. Even under heavy load, the latency per frame stayed under 45 milliseconds, ensuring nearly real-time responses. In CPU-only setups, the system maintained 10 to 12 FPS, which is acceptable for analyzing recorded footage or offline reviews.

8.2. Accuracy and Robustness

The YOLOv8-based detection module achieved high precision and recall across different environments. The combination of CNN-based recognition and DeepSORT tracking showed steady identity retention, even during partial obstructions and changing lighting conditions. The system maintained an average precision of 94.2% and recall of 91.6%, confirming its strength and ability to adapt to com-plex surveillance situations. Additionally, the re-identification accuracy across non-overlapping cam-era feeds hit 89.7%, showing reliable tracking across different views.

8.3. Computational Efficiency

The framework was designed for modular optimization and GPU acceleration to ensure efficient resource use. Batch processing and asynchronous data pipelines cut down memory usage while keeping camera streams synchronized. The YOLOv8 detector used mixed precision computation (FP16) for quicker inference without sacrificing accuracy. The DeepSORT tracker, along with Kalman filtering, kept low computational complexity, making the entire pipeline scalable across multiple nodes or edge devices.

8.4. Scalability and Adaptability

The system architecture supports integration with up to 32 camera feeds at once with little loss in performance. The modular design allows for distributed deployment across various servers or cloud nodes. Additionally, using containerized microservices through Docker enables flexible scalability, where new detection or tracking nodes can be added dynamically based on network demands. This makes the system suitable for real-world applications in air-ports, shopping malls, campuses, and smart cities.

Conclusion

This paper presents a comprehensive AI-powered surveillance framework that can detect, identify, and track lost objects and individuals in both live and recorded video feeds. By integrating YOLOv8 for multi-class object detection, CNN-based face recognition for identity re-identification, and DeepSORT for temporal tracking, the proposed system provides a unified and fully automated solution to modern surveillance challenges.

The main achievements and contributions of this work are summarized below:

- **High Accuracy in Object Detection:** The YOLOv8 module enables real-time detection of multiple object classes with high precision and recall, showing robust performance even in crowded or obstructed environments.
- **Reliable Identity Re-identification:** The CNN-based face recognition ensures consistent identification of individuals across non-overlapping cameras. This cross-camera re-identification allows accurate tracking of persons of interest and strengthens security monitoring.
- **Continuous Temporal Tracking:** DeepSORT keeps tracked entities consistent over time, reducing identity switches and maintaining accurate trajectories even in situations with motion blur, overlapping subjects, or temporary obstructions.
- **Real-Time Performance:** The framework achieves real-time performance across

multiple video streams, allowing instant detection, tracking, and alert generation, critical for proactive surveillance and quick incident responses.

- **Scalability and Flexibility:** Its modular design supports integration with various camera networks, allows for scaling in large deployments, and enables easy updates to detection or recognition modules without impacting overall system performance.
- **Automated Event Management:** The system includes a structured database for storing detection and tracking events, along with a rule-based alert mechanism that notifies security personnel about un-attended objects, unauthorized access, or suspicious behavior, reducing reliance on manual monitoring.
- **Practical Applicability:** The framework adapts to numerous real-world environments, such as airports, train stations, shopping malls, university campuses, and law enforcement operations, offering both live monitoring and post-event forensic analysis.
- **Foundation for Future Enhancements:** The sys-tem serves as a basis for adding advanced analytics, such as behavior prediction, anomaly detection, crowd movement analysis, or integrating with thermal or infrared imaging, which can further enhance security effectiveness and situational awareness.

Future Work and Enhancements

Integration with Advanced Analytics

Beyond object detection and face recognition, incorporating behavioral analysis and anomaly detection algorithms can help the system identify suspicious activities proactively. This could involve crowd movement analysis, loitering detection, abnormal behavior recognition, and predictive security alerts.

Multi-Modal Sensor Fusion

Adding other sensor types, like thermal cameras, infrared imaging, and audio sensors, can boost detection and tracking performance in challenging situations like low light, fog, or noisy settings. Sensor fusion can extend the system’s robustness and operational capabilities for round-the-clock monitoring.

Edge Deployment and Model Optimization

Deploying AI models on edge devices or cameras with limited resources can lessen bandwidth and latency issues. Future work might include model compression, pruning, quantization, or using lighter neural architectures for efficient on-device inference without losing accuracy.

Privacy-Preserving Mechanisms

As video surveillance raises privacy concerns, integrating privacy-protecting techniques such as face anonymization, selective masking, or federated learning can allow for safe monitoring while adhering to data protection regulations.

Improved Cross-Camera Synchronization

Further enhancements in cross-camera identity management could involve predictive tracking and trajectory estimation, enabling smoother identity transitions between cameras with minimal overlap and better long-term tracking consistency.

Real-Time Cloud-Based Analytics

Expanding the framework to use cloud computing and storage can facilitate large-scale surveillance across multiple sites or cities. Cloud integration allows centralized management, historical data analysis, and advanced visualization tools for operators.

Acknowledgements

The authors would like to sincerely thank Mrs. M. Sivaranajani, Associate Professor in the Department of Computer Science and Engineering at Paavai Engineering College, Namakkal. Her valuable guidance, mentorship, and continuous support through-out this research project have been greatly appreciated.

References

[1]. Author, "A Review on YOLOv8 and Its Advancements," ResearchGate, 2025. [Online]. Available:https://www.researchgate.net/publication/377216968 A Review on YOLOv8 and Its Advancements

[2]. B. Author, "Object Detection using YOLOv8: A Systematic Review," Sistemasi FTIK Unisi, 2025. [Online]. Available: https://sistemasi.ftik.unisi.ac.id/index.php/stmsi/article/view/5081

[3]. C. Author, "A Comprehensive Review on YOLO Versions for Object Detection," ScienceDirect, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2215098625002162

[4]. D. Author, "Improving Object Detection Perfor-mance through YOLOv8," arXiv, 2025. [Online]. Available: https://arxiv.org/pdf/2505.11424

[5]. E. Author, "Enhancing the YOLOv8 Model for Real-Time Object Detection," Nature, 2025. [Online]. Available: https://www.nature.com/articles/s41598-025-08413-4

[6]. F. Author, "YOLOv8 for Object Detection: A Comprehensive Review of Advances, Techniques, and Applications," IJACI, 2025. [Online]. Available: https://journal.cendekiajournal.com/ijaci/article/download/25/16

[7]. G. Author, "Facial Recognition Algorithms: A Systematic Literature Review," PMC, 2024. [Online]. Available: https://pmc.ncbi.nlm.nih.gov/articles/PMC11856072/

[8]. H. Author, "Convolutional Neural Networks for Face Recognition: A Systematic Literature Review," ResearchGate, 2024. [Online]. Available: https://www.researchgate.net/publication/372322451 Convolutional Neural Networks for Face Recogni-tion A Systematic Literature Review

[9]. I. Author, "Face Recognition using CNN and Siamese Network," ScienceDirect, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2665917423001368

[10]. J. Author, "Literature Survey on Face Recogni-tion with Hybrid Deep Learning Models," IJISAE, 2023. [Online]. Available: https://ijisae.org/index.php/IJISAE/article/view/6178

[11]. K. Author, "Face Recognition: A Literature Survey," 2003. [Online]. Available: https://mplab.ucsd.edu/∼marni/Igert/Zhao 2003.pdf

[12]. L. Author, "Comparative Survey Analysis of the CNN and LBPH Face Recognition Techniques," ACM Digital Library, 2024. [Online]. Available: https://dl.acm.org/doi/fullHtml/10.1145/3607947.3607978