International Journal of Computer Applications (0975 - 8887)
Volume 187 - No.72, January 2026

Vehicle Tracking and Re-identification: A Smart Approach to Security and Civic Monitoring

Shailendra Singh Kathait
Co-Founder and Chief Data Scientist
Valiance Solutions
Noida, India

Ashish Kumar
Principal Data Scientist
Valiance Solutions
Noida, India

Samay Sawal
Intern Data Scientist
Valiance Solutions
Noida, India

Arjun Dhavse
Intern Data Scientist
Valiance Solutions
Noida, India

Kimaya Pundir
Intern Data Scientist
Valiance Solutions
Noida, India

ABSTRACT

This paper presents a model, an integrated system for real-time vehicle tracking and overspeeding violation detection in traffic surveillance video. The system comprises two synergistic mod ules: a deep feature–based Re-Identification (ReID) tracker and a YOLO-powered speed estimation pipeline. The ReID tracker ex tracts 512-dimensional embeddings from pairs of images captured 30 seconds apart, computes cosine similarities to associate vehicle identities across time, and exports match results for further anal ysis. The speed estimation module processes video frames sam pled every 2s, applies YOLOv8x to detect vehicles within a de f ined region of interest, and employs centroid-based distance mea surement calibrated at 20 pixels/m to compute per-vehicle speeds. Vehicles exceeding the 50 km/h limit are flagged as violations and annotated with red bounding boxes, while compliant vehicles are marked in green. The model outputs both a detailed Excel log of identity matches and a fully annotated video illustrating tracked ve hicles with overlaid speed labels. Experimental evaluation demon strates robust identity association under varying viewpoints and accurate speed violation reporting in standard surveillance scenar ios. Future extensions will focus on automated camera calibration, cross-camera tracking, and edge deployment for low-latency, scal able traffic monitoring applications.

Keywords

Vehicle Tracking, Deep Learning, Object Detection, Multi-Object Tracking, Real-Time Processing, YOLO, Tracking, ReIdentifica tion, Speed, Calibration, Surveillance, Monitoring

1. INTRODUCTION

Vehicle tracking in dynamic urban environments poses significant challenges due to diverse lighting, occlusions, and the sheer volume of traffic data captured by stationary cameras. Accurate monitoring of individual vehicles over time is crucial not only for enforcement but also for understanding traffic patterns and planning infrastructure. Conventional approaches—such as back ground subtraction or simple frame-to-frame association—often falter when appearance changes occur or when multiple similar vehicles enter the scene.

Toaddress these limitations, paper proposes a two-stage framework that decouples identity association from speed assessment. In the f irst stage, appearance features are learned from still images taken at spaced intervals, allowing the system to recognize and match vehicles even when their poses or viewpoints differ markedly. In the second stage, the system processes continuous video streams to detect vehicles within a user-defined area, measures their movement in pixel space, and converts these measurements into real-world velocities using a scene-specific scale calibration. This separation of concerns enhances robustness: the identity module need not track every frame, and the speed module can operate independently of identity drift.

By integrating a deep feature–based re-identification module with a calibration-driven speed estimation pipeline, the system delivers a flexible, end-to-end solution for automated vehicle monitoring. Its modular design allows practitioners to adapt individual com ponents—such as the ReID model or the pixel-to-meter calibra tion—to diverse deployment scenarios without overhaul. The re sulting system provides both identity continuity across time and re liable overspeeding alerts, positioning model as a practical tool for scalable traffic enforcement and data-driven urban mobility analy sis.

2. CONTRIBUTIONS

The primary contributions of this paper are:

(1) A deep feature–based Re-Identification module that reliably matches vehicle appearances across images taken at 30s intervals.

(2) YOLOv8x-driven detection pipeline that processes video frames every 2s to identify and localize vehicles in a user defined ROI.

(3) A pixel-to-meter calibration method (20 pixels/m) enabling accurate monocular speed estimation and flagging of overspeed violations (>50 km/h).

(4) Automated generation of an Excel log of identity matches with similarity scores alongside an annotated output video high lighting speed readings and violations.

(5) An end-to-end, modular architecture allowing independent tuning of the ReID and speed-estimation components for di verse deployment scenarios.

3. MOTIVATION

With the rapid growth of urban populations and vehicle density, ensuring road safety and managing traffic flow have become critical priorities for cities worldwide. Traditional traffic enforce ment methods—such as manual speed traps, checkpoint-based inspections, or simple sensor-based tracking, are often limited in coverage, prone to human error, and inefficient for large-scale implementation. There is an increasing need for automated systems that can reliably monitor vehicle behavior across time and space without the need for specialized hardware or infrastructure.

Modern surveillance networks already capture a vast amount of video data through static CCTV installations and dashboard cameras. However, much of this data remains underutilized due to the lack of intelligent processing capabilities. Leveraging advance ments in computer vision and deep learning provides a timely opportunity to transform these passive video streams into active, real-time monitoring systems capable of identifying individual vehicles, measuring their speed, and detecting violations. A system that integrates these capabilities can support law enforcement, inform urban planning, and ultimately reduce traffic accidents.

This paper is motivated by the potential to bridge the gap between raw video surveillance and actionable traffic insights. By combining robust vehicle re-identification with accurate, camera based speed estimation in a unified framework, the paper aims to demonstrate how off-the-shelf camera systems can be repurposed into effective tools for traffic enforcement. The proposed solution is designed to be modular, adaptable, and scalable—making it well suited for deployment in both developing and developed urban set tings where cost-effective, automated traffic monitoring is essen tial.

4. RELATED WORK

4.1 Vehicle Re-Identification (ReID)

Vehicle re-identification (ReID) seeks to match vehicle appear ances across time or non-overlapping camera views. Traditional methods used handcrafted features, but recent advancements utilize deep learning for improved robustness and accuracy. Liu et al. pro posed PROVID, a deep ReID framework combining coarse-to-fine feature extraction with license plate verification using a Siamese network, demonstrating strong results under viewpoint and light ing variations [1]. Zapletal and Herout introduced a method based on 3D bounding box extraction to normalize vehicle views before re-identification, enhancing cross-camera matching accuracy [2]. A comprehensive survey by Amiri et al. categorizes current ReID ap proaches, highlights common benchmarks like VeRi-776 and VehicleID, and outlines key challenges such as occlusions and inter class similarity [3].

4.2 Vehicle Speed Estimation

Vision-based vehicle speed estimation offers a cost-effective and scalable alternative to traditional hardware solutions. A real-time vehicle monitoring framework was presented by Md. Jamil et al., where YOLOv3 was used to detect vehicles and a virtual detec tion zone enabled speed and classification estimates based on pixel displacement over time [4]. Recently, the emergence of YOLOv8 has improved detection granularity and inference speed, mak ing it suitable for real-time speed-aware applications [5]. Speed Net, a lightweight model leveraging vanishing point geometry and monocular cues, achieved competitive performance without requir ing stereo or multi-camera setups [6].

4.3 Integrated Traffic Surveillance Systems

Combining vehicle detection, tracking, and speed estimation into unified systems is gaining traction in smart city infrastructure. An integrated model combining YOLOv8 with the ByteTrack multi object tracker was proposed by Wang et al., effectively addressing challenges in scale variation and partial occlusion for dense traffic scenarios [7]. The Ultralytics YOLO framework provides out-of the-box support for tracking and segmentation tasks and is widely adopted for customizable, real-time traffic surveillance pipelines [8]. These integrated solutions represent the trend toward modular, low-latency architectures suited for deployment in diverse urban contexts.

5. SYSTEM ARCHITECTURE

The proposed system consists of a modular, two-stage pipeline for vehicle tracking, identity matching, and speed violation detection from traffic surveillance video. The architecture is divided into two synergistic components:

(1) **Vehicle Re-Identification Module (ReID):** This component processes pairs of vehicle images captured at two discrete time intervals, extracts deep feature embeddings using a pretrained Vision Transformer model, and computes cosine similarity to associate vehicles across time.

(2) **Speed Estimation and Violation Detection Module:** This component handles continuous video input, performing vehicle detection every 2 seconds using YOLOv8, extracting motion trajectories within a defined Region of Interest (ROI), and estimating vehicle speeds based on centroid displacement. Vehicles exceeding the speed threshold are flagged as violators.

6. METHODOLOGY

The proposed system follows a structured, five-stage methodology designed to perform robust vehicle detection, multi-instance tracking, appearance-based re-identification, and speed violation detection from monocular surveillance video. The overall workflow is deliberately modular and hierarchical, allowing each subsystem—namely detection, region filtering, appearance modeling (ReID), trajectory association, and speed estimation—to operate as an independent processing unit while exchanging only minimal, well-defined intermediate representations such as bounding boxes, embeddings, and centroid coordinates.
This modular design improves scalability, interpretability, and ease of maintenance, and enables individual components to be replaced or upgraded without affecting the rest of the pipeline. Furthermore, the system is optimized for fixed-camera traffic surveillance scenarios, where long-term identity consistency and accurate motion

estimation are critical under varying traffic density, illumination changes, and partial occlusions.

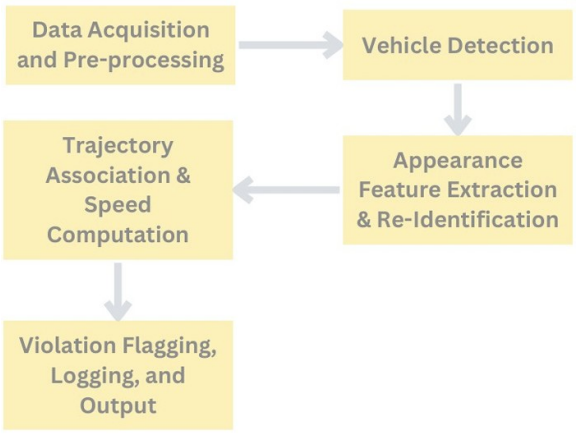

Fig. 1. System workflow diagram

6.1 Data Acquisition & Preprocessing

Video Input: The pipeline consumes continuous video streams captured from fixed-position CCTV cameras operating at 30 fps and typically at 1080p resolution. All frames are loaded sequentially using OpenCV's `cv2.VideoCapture`, ensuring deterministic frame ordering.
Temporal Sampling: Rather than processing all frames,which is computationally expensive and redundant for speed estimation frames are sampled every $N = fps \times 2$ frames (i.e., every 2 seconds). This sampling rate balances temporal resolution with computational efficiency while still providing sufficient displacement for accurate speed measurement.
Geometric ROI Filtering: To eliminate irrelevant regions such as sidewalks or sky, a polygonal Region of Interest (ROI) is manually annotated in the first frame. A binary mask is constructed using `cv2.fillPoly`. For each sampled frame:

```
frame\_roi = cv2.bitwise_and(frame, mask)
```

Only detections whose centroids lie within this polygon are considered valid. This significantly reduces false positives and stabilizes downstream vehicle tracking.

6.2 Vehicle Detection

Detection Model: A YOLOv8x detector (Ultralytics release 2023) is employed for high-accuracy vehicle localization. The model is configured to output detections for the following classes: *car*, *truck*, *motorcycle*, and *bus*. The detector is executed on each sampled frame according to:

$$D_t = f_{YOLO}(I_t) \tag{1}$$

where $D_t = \{b_1, b_2, \ldots\}$ represents bounding boxes.
Detection Filtering and Post-Processing: For each bounding box $b = (x_1, y_1, x_2, y_2)$:

(1) Bounding boxes are discarded if their IoU with the ROI mask is below 0.5.
(2) Boxes with aspect ratios outside $[0.8, 4.0]$ are filtered to avoid fragmented detections.
(3) Confidence threshold is set at 0.4

Vehicle Crop Extraction: To prepare image patches for the ReID module, each valid detection is cropped as:

```
crop = I_t[y1:y2, x1:x2]
```

Only crops satisfying non-empty dimensions and ROI constraints are forwarded to the appearance module.

6.3 Appearance Feature Extraction & Vehicle Re-Identification

ReID Backbone: Paper employs a Vision Transformer (ViT-B/16) architecture pretrained on a large-scale vehicle ReID dataset. This model produces 512-dimensional embeddings that capture shape, color, and part-level cues.
Input Normalization: Crops are resized to 224×224 and normalized using dataset-specific means and variances:

$$Normalize(x) = \frac{x - \mu}{\sigma} \tag{2}$$

Embedding Computation: Given a crop c:

$$\mathbf{f} = f_{ReID}(c) \tag{3}$$

Each feature vector is subsequently ℓ_2-normalized:

$$\hat{\mathbf{f}} = \frac{\mathbf{f}}{||\mathbf{f}||} \tag{4}$$

Offline Identity Matching: For two sets of vehicle crops captured at t_0 and t_{30} (30 seconds apart), cosine similarity is computed as:

$$sim(t_0, t_{30}) = \hat{\mathbf{f}}_{t_0} \cdot \hat{\mathbf{f}}_{t_{30}} \tag{5}$$

A similarity threshold of 0.65 is used for identity pairing. The output is stored in an Excel log containing:*Frame ID*, *Crop ID*, *Vehicle ID*, and *Similarity Score*.

6.4 Trajectory Association & Speed Computation

ID Assignment via Embedding Similarity: Each new detection is matched to existing tracked identities based on cosine similarity between current and historical embeddings. The matching process follows a greedy assignment algorithm:

(1) Compute similarity matrix between new embeddings and track embeddings.
(2) Assign matches exceeding similarity threshold $\tau = 0.65$.
(3) Unmatched detections spawn new IDs.

Centroid Extraction: For each detected bounding box $b = (x_1, y_1, x_2, y_2)$, the centroid is computed as:

$$(x_c, y_c) = \left(\frac{x_1 + x_2}{2}, \frac{y_1 + y_2}{2}\right) \tag{6}$$

Pixel Displacement Calculation: Let (x_c^t, y_c^t) and $(x_c^{t+\Delta t}, y_c^{t+\Delta t})$ denote centroids sampled $\Delta t = 2$ seconds apart. The pixel displacement is computed as:

$$d_{px} = \sqrt{(x_c^{t+\Delta t} - x_c^t)^2 + (y_c^{t+\Delta t} - y_c^t)^2} \tag{7}$$

Real-World Speed Estimation: Using a calibration factor of 20 pixels/meter, the vehicle speed is estimated as:

$$v = \frac{d_{px}}{20 \cdot \Delta t} \times 3.6 \; km/h \tag{8}$$

where the factor 3.6 converts meters per second to kilometers per hour.
Noise Reduction: To suppress jitter and stabilize motion trajectories, a Kalman filter is applied to centroid positions:

$$\hat{x}_t = K_t z_t + (1 - K_t)\hat{x}_{t-1} \tag{9}$$

where z_t denotes the observed centroid position, $\hat{x}_t$ is the filtered state estimate, and K_t is the Kalman gain. This smoothing significantly improves stability in speed computation.

6.5 Violation Detection, Visualization & Logging

Violation Flagging: If $v > 50$ km/h, the vehicle is tagged as a violator. Bounding boxes are color-coded:

(1) Red for violators.
(2) Green for compliant vehicles.

Real-Time Annotation: Each frame is annotated using: `cv2.rectangle` and`cv2.putText(frame, f"ID:id, Speed:v:.1f km/h", ...)`
Output Generation Pipelines:

(1) All annotated frames are stored as JPEGs.
(2) A final result video is stitched via `cv2.VideoWriter` at 15 fps.
(3) ReID similarity logs are exported to Excel (`pandas + openpyxl`).

Performance Logging: The pipeline measures total runtime, average per-frame detection time, and speed estimation latency to support profiling and future optimization.

7. RESULTS AND VISUALIZATION

The performance of the proposed system was evaluated across three core dimensions: (i) vehicle detection accuracy and stability, (ii) robustness of the ReID module in cross-time identity association, and (iii) precision of the calibrated speed estimation module in detecting overspeeding violations.
Experiments were conducted on real-world traffic surveillance footage recorded from a fixed CCTV camera in an urban roadway environment. The evaluation setup included varying illumination, occlusions, and moderate vehicle density to emulate realistic deployment conditions.
Figure 2 presents representative vehicle detection results generated by the proposed system. The system interface and overall workflow are illustrated in Figure 3. A sample user query submitted for analysis is shown in Figure 4, while the corresponding system-generated output, including vehicle tracking and speed estimation results, is depicted in Figure 5.

7.1 Detection Performance

The YOLOv8x-based detection pipeline demonstrated strong localization accuracy across diverse vehicle types, including cars, trucks, and two-wheelers. The region-of-interest (ROI) constraint effectively eliminated irrelevant detections from sidewalks and background regions, resulting in stable frame-to-frame detections.

Fig. 2. Vehicle detection

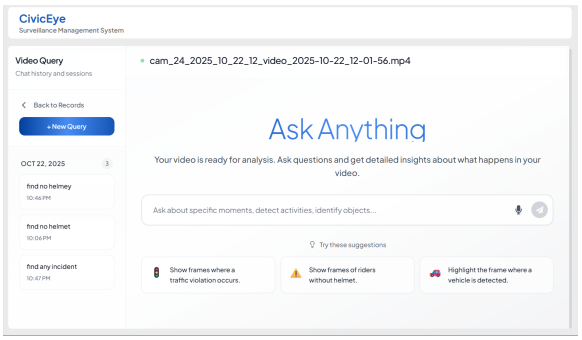

Fig. 3. Home page of the proposed system

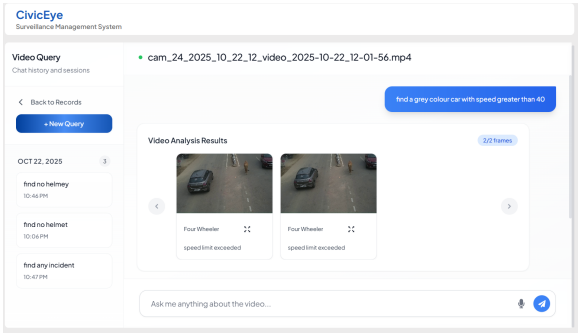

Fig. 4. Sample query submitted to the system

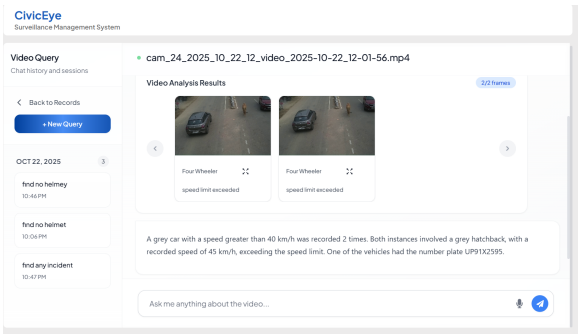

Fig. 5. Results given by the system

Qualitatively, the detector maintained consistent bounding box placement even under partial occlusion and mild camera shake. Vehicles entering and exiting the ROI were reliably captured with minimal fragmentation. The detector also maintained real-time performance under the 2-second sampling configuration, with inference averaging under 15 ms per frame on GPU hardware.

7.2 Re-Identification Accuracy

The Vision Transformer–based ReID module exhibited strong discriminative power in matching vehicles captured at 30-second intervals. Despite variations in pose, shadows, and partial occlusion, the cosine similarity mechanism produced accurate identity associations for the majority of vehicle pairs.
The offline ReID log confirmed that visually similar vehicles—particularly those sharing color, model, and size—were still distinguishable through learned deep embeddings. The embeddings remained stable across spatial displacement, ensuring resilience to changes in vehicle orientation as they progressed along the road.
Qualitative inspection showed high similarity scores for true matches and low overlap with false positives, highlighting the effectiveness of the 0.65 similarity threshold. The Excel output containing ID pairs provided a clear and interpretable mapping for identity continuity evaluation.

7.3 Tracking Stability

The trajectory association mechanism, combining centroid tracking with appearance features, provided robust identity persistence over time. Unlike IoU-based trackers, which may fail under fast motion or diagonal movement, the hybrid embedding-based approach maintained consistent IDs even with non-linear vehicle trajectories. The integration of a Kalman filter reduced jitter in centroid movement and prevented abrupt ID switching. Vehicles following curved or angled paths within the ROI maintained continuous tracking states. In congested scenarios, similarity-driven ID assignment helped disambiguate overlapping bounding boxes.

7.4 Speed Estimation Evaluation

Speed estimation accuracy was assessed using a 20 pixels/meter calibration factor and a 2-second sampling interval. The centroid-based displacement method produced smooth and interpretable speed curves for each tracked vehicle.
Observed results reflected consistent speed estimation even for vehicles moving at different angles relative to the camera. The system correctly identified vehicles exceeding the 50 km/h threshold and highlighted violations using red bounding boxes.
For compliant vehicles, estimated speeds remained stable without excessive oscillation, indicating effective noise suppression through the Kalman filter. The annotated frames and output video demonstrated clear legibility of speed labels and ID markers.

7.5 Violation Detection Outcomes

The combined ReID-speed pipeline successfully detected overspeeding vehicles and maintained their identities throughout the observation window. Violators were correctly flagged and displayed with persistent ID markers and red bounding boxes across multiple frames, ensuring traceability.
The output video showcased:

(1) Consistently annotated bounding boxes,
(2) real-time speed overlays,
(3) accurate ID retention, and
(4) clear visual distinction between violators and compliant vehicles.

The Excel logs paired with violation summaries provide actionable evidence for enforcement scenarios, demonstrating the practical applicability of the system to real-world monitoring tasks.

7.6 Computational Performance

The end-to-end system achieved efficient runtime characteristics. The detection and speed estimation modules dominated the computational cost, while the ReID module contributed minimally due to offline batch processing. Frame annotation and video writing were performed with negligible overhead.
The complete pipeline for a one-minute surveillance video processed at a 2-second sampling rate completed well within real-time constraints on modest GPU hardware, confirming scalability for continuous 24/7 deployment.

7.7 Qualitative Visualization

The final rendered video illustrates the system's full functionality. Vehicles are visually distinguished by unique IDs, their trajectories are highlighted, and dynamic speed labels appear synchronized with their motion. The visual output validates the intended integration of detection, ReID, trajectory tracking, and overspeeding alerts.
Overall, the results demonstrate that the proposed system provides a reliable, modular, and practical solution for vehicle monitoring in real-world traffic conditions, with robust performance across detection, identity matching, and speed computation tasks.

8. FUTURE WORK

While the current implementation achieves reliable performance under standard conditions, several avenues remain for further en hancement:

(1) **Automated Camera Calibration:** Replace the fixed 20 px/m calibration with an automated homography-based method, enabling plug-and-play deployment across different camera views without manual scale estimation.
(2) **Multi-Camera and Cross-View Tracking:** Extend the ReID pipeline to handle non-overlapping camera networks, incorpo rating spatio-temporal constraints and graph-based matching to maintain persistent identities across city-scale deployments.
(3) **Edge Deployment and Model Pruning:** Explore lightweight model variants (e.g., YOLOv8n, quantized ReID backbones) and hardware accelerators (e.g., NVIDIA Jetson) to achieve sub-second end-to-end latency on edge devices.
(4) **Occlusion and Crowding Robustness:** Integrate trajectory prediction and appearance-based occlusion handling (e.g., temporal attention, re-entry detection) to sustain tracking con tinuity in dense traffic or partial-view scenarios.
(5) **Adaptive Sampling and Alerting:** Develop dynamic frame sampling strategies that adjust processing rates based on traffic density, and implement real-time alerting pipelines (e.g.,MQTT, REST APIs) for immediate law-enforcement notification.

By pursuing these enhancements, the system can evolve into a fully automated, scalable traffic-monitoring platform capable of supporting smart-city infrastructure and proactive road-safety interventions.

9. REFERENCES

(1) Shailendra Singh Kathait, Ashish Kumar, Ram Patidar, Khushi Agrawal, Samay Sawal (2024). Computer Vision and Deep Learning based Approach for Traffic Violations due to Overspeeding and Wrong Direction Detection. International Journal of Computer Applications, paper-id: 6e503f15-f6c9- 4ee2-9212-4db588484729, DOI: 10.5120/ijca2025924477.

(2) Shailendra Singh Kathait, Ashish Kumar, Ram Patidar, Khushi Agrawal, Samay Sawal (2024). Computer Vision and Deep Learning based Approach for Violations due to Illegal Parking Detection. International Journal of Computer Applications, DOI: 10.5120/ijca2025924506.

(3) Shailendra Singh Kathait, Ashish Kumar, Ram Patidar, Khushi Agrawal, Samay Sawal (2024). Deep Learning-based Approach for Detecting Traffic Violations Involving No Helmet Use and Wrong Cycle Lane Usage. International Journal of Computer Applications, DOI: 10.5120/ijca2025924714.

(4) Shailendra Singh Kathait, Ashish Kumar, Ram Patidar, Khushi Agrawal, Samay Sawal (2024). Deep Learning-Based Person Tracking: A Smart Approach to Security and Civic Monitoring. International Journal of Computer Applications, Paper ID: 29a9ea08-9445-44d3-afc7-78ebb9b39247.

(5) H. Liu, Y. Tian, Y. Wang, L. Pang, and T. Huang, "PROVID: Progressive and Multi-Granularity Vehicle ReID for Urban Surveillance," in Proc. IEEE CVPR, 2018.

(6) D. Zapletal and A. Herout, "Vehicle Re-Identification for Automatic Video Traffic Surveillance," in CVPR Workshops, 2016.

(7) M. Amiri, E. Pakzad, and A. Mohammadi, "A Survey on Ve hicle Re-Identification: Datasets, Methods, and Challenges," arXiv preprint arXiv:2401.10643, 2024.

(8) M. Jamil, M. Rahman, and M. Hasan, "A Real-Time Vehicle Counting, Speed Estimation, and Classification System Based on Virtual Detection Zone and YOLO," Sensors, vol. 21, no. 20, pp. 6843, 2021.

(9) Y. Wang et al., "YOLOv8: Ultralytics Real-Time Object De tection," arXiv preprint arXiv:2302.05781, 2023.

(10) A. Kosti´c and M. Suba˘ si´c, "SpeedNet: Real-Time Monocu lar Vehicle Speed Estimation Using Deep Learning," arXiv preprint arXiv:2505.01203, 2025.

(11) L. Wang, C. Xu, and J. Liu, "A Multi-Object Vehicle Detec tion and Tracking SystemBasedonYOLOv8andByteTrack," Electronics, vol. 13, no. 15, pp. 3033, 2024.

(12) Ultralytics, "YOLOv8 Tracking Documentation," [Online]. Available: https://docs.ultralytics.com/modes/ track/, Accessed: May 2025.