

## Article

# A Unified Framework for Vehicle Detection, Tracking, and Counting Across Ground and Aerial Views Using Knowledge Distillation with YOLOv10-S

Md Rezaul Karim Khan  and Naphtali Rishen \* 

Knight Foundation School of Computing and Information Sciences (KFSCIS), Florida International University, Miami, FL 33199, USA; mkhan157@fiu.edu

\* Correspondence: rishen@cs.fiu.edu

## Highlights

### What are the main findings?

- The proposed unified and modular framework effectively integrates vehicle detection, multi-object tracking, and trajectory-based counting into a consistent end-to-end pipeline that works across both ground and aerial surveillance scenarios.
- Knowledge distillation strengthens the lightweight YOLOv10-S detector without architectural modification, improving temporal stability and enhancing overall system reliability across diverse viewpoints.

### What are the implications of the main findings?

- The study underscores the importance of evaluating traffic monitoring from a system-level perspective, where coordinated detection and tracking directly influence counting accuracy and robustness.
- The proposed framework provides a practical and scalable foundation for intelligent transportation applications, offering cross-domain adaptability and real-time feasibility for real-world traffic monitoring.

## Abstract

Accurate and reliable vehicle detection, tracking, and counting across different surveillance platforms are fundamental requirements for developing smart Traffic Management Systems (TMS) and promoting sustainable urban mobility. Recent advances in both ground-level surveillance and remote sensing using deep learning have opened new opportunities for extracting detailed vehicular information from high-resolution aerial and surveillance video data. Our research reported here aims to present a unified, real-time vehicle analysis framework that integrates lightweight deep learning-based detection, robust multi-object tracking, and trajectory-driven counting within a single modular pipeline. The proposed framework employs a “You Only Look Once” system, YOLOv10-S as the detection backbone and enhances its robustness through supervision-level knowledge distillation without introducing any architectural modifications. Temporal consistency is enforced using an observation-centric multi-object tracking algorithm (OC-SORT), enabling stable identity preservation under camera motion and dense traffic conditions. Vehicle counting is performed using a trajectory-based virtual gate strategy, reducing duplicate counts and improving counting reliability. Comprehensive experiments conducted on the UA-DETRAC and VisDrone benchmarks show that the proposed framework effectively balances detection performance, tracking robustness, counting accuracy, and real-time efficiency in both ground-based and aerial surveillance settings. Furthermore, cross-dataset evaluations under direct train–test transfer highlight the inherent challenges of domain shift while



Academic Editor: Tania Stathaki

Received: 21 January 2026

Revised: 5 March 2026

Accepted: 6 March 2026

Published: 9 March 2026

**Copyright:** © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and

conditions of the [Creative Commons](https://creativecommons.org/licenses/by/4.0/)

[Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

showing that knowledge distillation consistently improves robustness in detection, tracking identity consistency, and vehicle counting. Overall, this framework enables effective real-world traffic monitoring by adopting a scalable and practical system design, where reliability is prioritized over architectural complexity.

**Keywords:** vehicle detection; multi-object tracking; vehicle counting; intelligent transportation systems; knowledge distillation; deep learning; UAV-based traffic monitoring; cross-domain generalization

---

## 1. Introduction

The rapid expansion of urban populations places significant pressure on transportation infrastructure. As a result, accurate and reliable vehicle detection, tracking, and counting technology plays a crucial role in smart-city infrastructures and modern Intelligent Transportation Systems (ITS) [1]. Modern road networks, characterized by their scale and structural complexity, require constant monitoring and strategic management. Therefore, accurate and timely traffic information is essential for city planners and transportation authorities to reduce congestion, refine signal timing, increase road safety, and support long-term mobility planning. Over the past few years, advances in deep learning and computer vision [2] have significantly enhanced the ability to extract vehicular information from ground-based cameras [3] and aerial remote-sensing systems, including UAVs and drones [4]. These technologies offer both high spatial detail and time-varying information, making them especially valuable for analyzing detailed vehicle behavior in dense traffic conditions.

Despite significant progress in deep learning-based systems, real-world traffic environments still face considerable challenges. Traffic video streams often suffer from occlusion among closely packed vehicles, varying lighting conditions, shadows, weather disturbances, and motion blur caused by camera movement or fast-moving objects. Such complications reduce the effectiveness of object detectors, particularly in drone-based or aerial scenes, where vehicles are represented as very small objects against complex backgrounds [5,6]. The problem becomes even more difficult in multi-class settings, where vehicles with similar appearances, such as vans, pickup trucks, and small buses, must be differentiated under perspective distortion or low resolution [7]. While a broad range of detection models have been proposed for surveillance videos and UAV imagery, many still struggle to maintain reliable performance across diverse viewing conditions, particularly when accurate temporal association and stable object tracking are required.

Beyond detection, vehicle counting introduces additional complexity. Detection-based counting approaches are highly sensitive to missed or duplicated detections, while tracking-based counting requires consistent object identity association across frames [8]. In UAV-based traffic monitoring, the rapid motion of drones and abrupt viewpoint changes further complicate counting due to increased ID switches and tracking interruptions [9]. Several studies have explored detection-and-tracking pipelines for counting vehicles in traffic scenes [10], and others have proposed specialized models for small-object vehicle counting in aerial imagery [11]. Despite these efforts, combining accurate detection, robust multi-object tracking, and stable counting within a unified framework remains an underexplored area.

Another critical limitation in the prior work lies in generalization across datasets and viewpoints. Most studies evaluate detection or tracking performance within a single dataset or a fixed surveillance perspective, either ground-level or aerial. However, real-world

ITS deployments often require models to operate across heterogeneous camera setups, including fixed ground cameras and mobile aerial platforms. While cross-domain methods have been proposed to mitigate domain shifts between different environments, very few works systematically examine how a unified model [12,13] trained on one perspective (e.g., ground-level CCTV) performs when tested on a drastically different perspective (e.g., aerial UAV). This gap limits the practical deployability of existing systems.

In response to these challenges, the present work proposes a real-time framework with a unified yet modular design for vehicle detection, tracking, and counting that operates consistently across heterogeneous surveillance domains. YOLO (“You Only Look Once”) is one of the most popular frameworks for object detection in the field of computer vision. YOLOv10 [14] is a state-of-the-art, open-source software implementation of YOLO, together with trained models. YOLOv10-S is a lightweight version of YOLOv10 designed for speed and efficiency on low-power devices while still identifying objects accurately in real time. Our proposed framework follows a detection–tracking–counting paradigm, employing YOLOv10-S as a lightweight detection backbone, an observation-centric multi-object tracking algorithm (OC-SORT) [15] to enforce temporal consistency, and a trajectory-based counting strategy for traffic flow estimation. Without introducing architectural modifications, dataset-adaptive knowledge distillation is applied at the supervision level to enhance detection robustness while preserving real-time efficiency. Experimental results on the UA-DETRAC dataset [16] and VisDrone dataset [17] benchmarks confirm that the proposed framework performs effectively in both ground-level and aerial surveillance settings and is robust to domain and viewpoint differences. Certain individual components employed in this work, including YOLOv10-S, knowledge distillation, and OC-SORT, are established methods. The novelty of the present study lies in structured integration into a unified detection–tracking–counting pipeline designed for heterogeneous traffic surveillance scenarios. Unlike prior works that optimize detection, tracking, or counting independently, our framework systematically evaluates interdependence under a consistent experimental protocol across both ground-based and aerial domains. This system-level design enables controlled component-wise analysis and practical deployment without architectural modification. The core contributions of the present work:

- We design a modular yet unified framework that integrates vehicle detection, multi-object tracking, and trajectory-based counting, enabling consistent traffic analysis across both ground-based and aerial surveillance scenarios.
- Knowledge distillation is applied without modifying the detector architecture, improving detection robustness while preserving the efficiency of the lightweight YOLOv10-S model.
- An observation-centric tracking strategy (OC-SORT) is employed to mitigate identity switches and trajectory fragmentation under camera motion and abrupt viewpoint changes, resulting in more reliable vehicle counting.
- Extensive cross-domain evaluations on the UA-DETRAC (ground-based) and Vis-Drone (aerial) benchmarks validate the robustness and generalization ability of the proposed framework across diverse traffic scenarios.

## 2. Related Work

Automated traffic analysis has been widely studied by computer vision researchers, primarily focusing on vehicle detection, multi-object tracking, and vehicle counting. These tasks are often addressed independently and evaluated under fixed camera viewpoints or single datasets. Real-world traffic monitoring systems increasingly require unified frameworks that can operate across multiple scenarios, including both ground-based

surveillance and aerial imagery. This section discusses related studies in vehicle detection, tracking, and counting, highlighting robustness across varying viewpoints and datasets.

### 2.1. Vehicle Detection in Ground and Aerial Traffic Scenes

Deep learning-based detectors have become the leading approach for vehicle detection, driven by their strong representation learning and efficient inference speed. In traffic surveillance applications, single-stage detectors such as the YOLO family [18] are widely used due to their ability to combine high accuracy with efficient processing. Successive versions such as YOLOv5 [19] and YOLOv8 [20] introduced improved feature aggregation and training strategies, achieving strong performance in urban traffic scenes. Transformer-based detectors, including RT-DETR [21], have recently demonstrated competitive detection accuracy by leveraging global context modeling and efficient end-to-end training, particularly in structured traffic environments with stable camera viewpoints.

Vehicle detection in aerial imagery remains significantly more challenging compared to ground-based surveillance. In UAV scenarios, vehicles often appear as extremely small objects with limited pixel representation and high intra-class variation caused by altitude changes, camera motion, and viewing angle. A number of research efforts have concentrated on improving the recognition of small objects across ground and aerial images. Jin et al. [22] enhanced a YOLOv5-based detector by integrating a transformer module and an asymmetric focal loss, achieving improved performance for dense and occluded vehicle detection on the UA-DETRAC benchmark dataset. Chang et al. [23] improved a YOLO-based detection model by integrating enhanced feature extraction and attention mechanisms to effectively address fuzzy and small-target detection challenges in UAV imagery. Liu et al. [24] conducted a comprehensive survey and performance evaluation of deep learning-based small object detection methods, analyzing key challenges and solutions and comparing leading detectors such as YOLOv3, and SSD across benchmark datasets. The VisDrone-DET benchmark and challenge were introduced by Du et al. [25] to provide a comprehensive evaluation platform for UAV-based object detection and to highlight the significant performance gaps that remain in drone imagery analysis.

Although these detection methods achieve strong results within their respective domains, they often exhibit notable performance degradation when transferred across viewpoints, such as from ground-based cameras to UAV imagery, due to domain shifts in scale, perspective, and scene geometry.

### 2.2. Multi-Object Tracking for Traffic Analysis

Multi-object tracking (MOT) is essential for traffic analysis, as accurate vehicle counting and trajectory estimation rely on maintaining consistent identities across video frames. Most modern vehicle tracking methods adopt a tracking-by-detection paradigm, where detections are temporally associated using motion and appearance cues. N. Wojke et al. introduced DeepSORT [26], which integrates a re-identification network that extracts appearance features to support more reliable identity matching under occlusion. To enhance robustness in dense traffic scenes, Y. Zhang et al. proposed ByteTrack [27], demonstrating that associating both high- and low-confidence detections significantly improves tracking continuity. Building on this idea, N. Aharon et al. introduced BoT-SORT [28], which integrates improved motion modeling, camera motion compensation, and appearance-based association to further stabilize tracking. More recently, C. Cao et al. proposed OC-SORT [15], an observation-centric tracking framework that reduces reliance on appearance features and improves robustness under camera motion, making it well-suited for UAV-based and dynamic traffic surveillance scenarios.

These trackers have been widely evaluated on traffic benchmarks such as UA-DETRAC-MOT and VisDrone-MOT, representing ground-based and aerial monitoring settings, respectively. However, their cross-domain generalization ability across different viewpoints and sensing platforms remains insufficiently explored, motivating unified benchmarking across both datasets.

### 2.3. Vehicle Counting Approaches

Vehicle counting techniques are generally categorized into detection-based and tracking-based methods. Detection-based approaches count vehicles by summing frame-level detections, but they are prone to missed vehicles and repeated counting. Tracking-based counting improves robustness by associating detections over time and counting vehicles as they cross predefined virtual regions. Early UAV-based vehicle counting approaches relied on object detection combined with frame-level heuristics to reduce duplicate counts. More recent methods integrate detection with multi-object tracking to enhance reliability. Xiang et al. [9] proposed a UAV-based vehicle counting framework that integrates vehicle detection and online tracking to robustly handle both static and moving backgrounds in aerial videos, demonstrating effective performance in real-world highway monitoring scenarios. By combining YOLO-based object detection with multi-object tracking, Lu et al. [29] improved vehicle counting performance through more stable identity association across frames, minimizing duplicate detections and identity breaks. Mandal and Adu-Gyamfi [30] combined modern object detection and tracking methods into a single pipeline to enable accurate vehicle counting from traffic video data. Despite these advances, most vehicle counting methods remain highly dependent on accurate tracking performance and dataset-specific parameter tuning. Furthermore, evaluation is typically limited to a single dataset or viewpoint, and cross-dataset counting performance is rarely reported.

### 2.4. Cross-Domain and Cross-Perspective Generalization

Cross-domain vehicle detection has gained increasing attention as traffic monitoring systems are required to operate across diverse environments. To address domain shift, Xu et al. [31] presented a cross-domain car detection framework that integrates attention mechanisms for more robust detection. The C2FDA framework introduced by Zhang et al. [32] addresses cross-domain traffic object detection by providing a coarse-to-fine domain adaptation strategy that improves detector robustness and mitigates performance degradation when deployed in complex and unseen traffic environments. Despite progress in cross-domain detection, most existing studies focus exclusively on detection and require retraining or domain-specific adaptation. Unified evaluation of detection, tracking, and counting across datasets—without retraining—remains rare, particularly when transitioning between ground-based and aerial traffic scenes. This limitation highlights the need for generalizable traffic analysis frameworks capable of maintaining stable detection, tracking, and counting performance across heterogeneous viewpoints using fixed model weights.

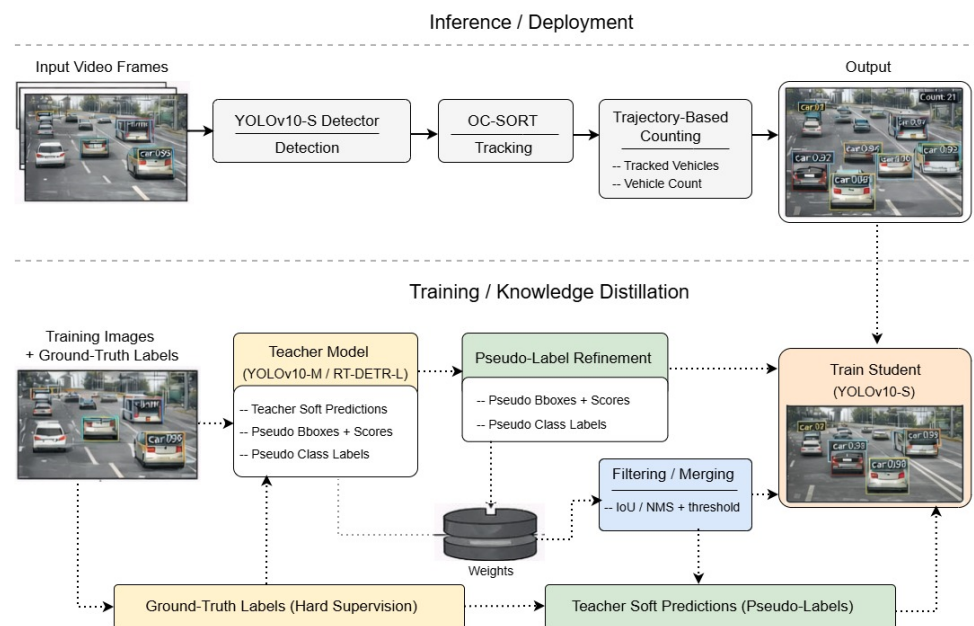
## 3. Methodology

In this work, we develop a modular and unified deep learning framework for vehicle detection, tracking, and counting, with consistent performance across ground-level surveillance and aerial remote-sensing scenarios. The methodology is structured to clearly separate detection, tracking, and counting as distinct but interconnected tasks, enabling fair evaluation and targeted optimization at each stage. The overall pipeline follows a detection–tracking–counting paradigm, where a unified detector produces frame-level vehicle predictions, a multi-object tracking module enforces temporal consistency, and a trajectory-

based counting module estimates vehicle flow. The overall architecture is designed to balance high accuracy with real-time efficiency, while maintaining reliable performance across datasets captured under different viewpoints, resolutions, and traffic densities.

### 3.1. Overall Framework Formulation

The proposed framework detects vehicles separately in each video frame and then links these detections over time, following a tracking-by-detection methodology to produce continuous trajectories. A lightweight convolutional detector serves as the front-end to ensure real-time feasibility, while a tracking module operates on top of the detector outputs to maintain identity consistency. Vehicle counting is performed as a post-processing step using tracked trajectories rather than raw detections, which significantly reduces duplicate counts caused by missed or fragmented detections as illustrated in Figure 1.



**Figure 1.** Overview of the Knowledge Distillation Framework for the YOLOv10-S Detector.

The performance improvements are achieved through a disciplined training strategy and Knowledge Distillation (KD) as discussed in Section 3.3, allowing the detector to benefit from stronger supervisory signals while preserving its original architecture and inference efficiency. This design decision makes it clear that the reported performance improvements are due to the proposed methodological pipeline rather than changes to the model architecture.

### 3.2. Vehicle Detection Backbone (YOLOv10-S)

The YOLOv10-S model [14] serves as the core detection backbone in our proposed framework for vehicle detection. YOLOv10-S is chosen for its lightweight design, fast inference speed, and reliable real-time detection performance, which make it suitable for deployment on resource-limited platforms such as edge devices and UAVs. Unlike earlier YOLO versions, YOLOv10-S eliminates the need for non-maximum suppression (NMS) while maintaining competitive detection accuracy.

The architecture of YOLOv10-S aligns with the typical YOLO framework, where the backbone, neck, and head serve as its core components.

**Backbone:** The backbone processes the input image to obtain meaningful features, combining local spatial details with global semantic context to support robust vehicle detection across multiple scales and perspectives.

Neck: The neck combines features from multiple scales to effectively handle vehicles of varying sizes and passes this information to the head. The neck plays a particularly important role in handling both large vehicles in ground-based surveillance scenes, as well as small, densely packed vehicles in aerial and remote-sensing imagery.

Head: The detection head produces bounding box coordinates along with confidence scores and vehicle class predictions in a single step. This joint prediction mechanism allows detection and classification to be performed in a single forward pass.

For an input frame  $I_t \in \mathbb{R}^{H \times W \times 3}$ , the detector outputs a set of predictions:

$$D_t = \{(b_i^t, c_i^t, s_i^t)\}_{i=1}^{N_t} \quad (1)$$

where  $b_i^t = (x_i^t, y_i^t, w_i^t, h_i^t)$  defines the bounding box described by its position and size,  $c_i^t$  refers to the predicted vehicle type, and  $s_i^t$  measures the confidence level of the detection. In our framework, the detector architecture is used as-is, without any structural changes or customized layers, ensuring architectural consistency across all experiments and datasets.

While this work mainly focuses on vehicle detection, tracking, and counting, category-level vehicle classification is also implicitly supported by the detector's multi-class prediction head. Each detected vehicle is automatically assigned a semantic class label (e.g., car, bus, van), which is later propagated through the tracking stage to enable class-aware analysis without introducing additional classification modules.

### 3.3. Knowledge Distillation Strategy for Enhancing Vehicle Detectors

Knowledge Distillation (KD) [33] refers to a training approach where a powerful, high-capacity teacher model guides a smaller, lightweight student model by sharing its learned knowledge. The objective is to enable the student model to approach teacher-level accuracy with reduced computational cost, memory usage, and latency, allowing efficient deployment in real-time or edge environments. As illustrated in Figure 1, the student model is trained using both ground-truth labels and soft supervision provided by the teacher.

Figure 1 presents the overall architecture of the proposed unified detection–tracking–counting framework, consisting of two main stages: training with knowledge distillation and inference/deployment. During training, the teacher model (YOLOv10-M or RT-DETR-L) processes the input images to generate soft predictions. These predictions are converted into pseudo-labels and further refined using IoU-based non-maximum suppression (NMS) and confidence threshold filtering to improve label reliability. The student detector (YOLOv10-S) is then trained using dual supervision signals, ground-truth annotations providing hard supervision, and refined pseudo labels providing soft supervision through knowledge distillation. During inference, only the distilled student detector is deployed for efficiency. The detected bounding boxes are passed to the OC-SORT tracker for identity association across frames, and the resulting trajectories are used by the trajectory-based virtual gate module to compute vehicle counts. This modular design enables clear separation between detection, tracking, and counting, while maintaining a unified evaluation framework.

A knowledge distillation approach is employed to boost the detection performance of the lightweight YOLOv10-S model without introducing additional architectural complexity. Within this framework, a more powerful teacher model produces high-confidence pseudo-labels for the training data, which are combined with the original ground-truth annotations to supervise the training of the YOLOv10-S student model.

Let  $\mathcal{D}_t^{GT}$  denote the ground-truth annotations and  $\mathcal{D}_t^T$  denote the teacher predictions at iteration  $t$ . The merged supervision set is defined as

$$\mathcal{D}_t^* = \mathcal{D}_t^{GT} \cup \mathcal{D}_t^T \setminus \mathcal{O}, \quad (2)$$

where  $\mathcal{O}$  represents duplicate detections removed using an IoU-based suppression rule. Ground-truth bounding boxes are always preserved to prevent supervision drift.

The student model is trained using the same optimization settings as the baseline detector, ensuring that performance gains are attributable solely to improved supervision rather than altered training conditions. Both same-family and cross-architecture (a transformer-based detector) distillation settings are examined to analyze their impact on detection generalization.

This distillation-driven enhancement forms a central component of the proposed methodology. Rather than altering the detector architecture, the approach leverages improved supervision to strengthen feature learning and generalization, while maintaining the real-time efficiency required for traffic surveillance applications.

### 3.4. Multi-Object Tracking Module (OC-SORT)

In this study, OC-SORT (Observation-Centric SORT) [15] is used as the multi-object tracking component within a tracking-by-detection framework, where frame-level vehicle detections are temporally associated to maintain identity consistency across consecutive video frames and enable reliable vehicle counting. Let

$$\mathcal{T} = \{T_k\}_{k=1}^K \quad (3)$$

represents the collection of tracked trajectories, where  $K$  indicates the total number of objects tracked within a video sequence. Each trajectory  $T_k$  corresponds to a single vehicle observed over multiple frames and is represented as an ordered sequence of bounding boxes:

$$T_k = \{b_k^1, b_k^2, \dots, b_k^{t_m}\}, \quad (4)$$

where  $b_k^t$  denotes the bounding box associated with object  $k$  at frame  $t$ , and  $t_m$  is the last frame in which the object is visible.

OC-SORT performs data association primarily based on spatial overlap and motion consistency between detections in consecutive frames. The cost used for association is calculated using the Intersection-over-Union (IoU) metric:

$$\text{IoU}(b_i^t, b_j^{t-1}) = \frac{|b_i^t \cap b_j^{t-1}|}{|b_i^t \cup b_j^{t-1}|}. \quad (5)$$

In contrast to simpler IoU-based trackers, OC-SORT explicitly models object motion using a velocity-aware state estimation and applies camera motion compensation to decouple object movement from global camera displacement. This design significantly improves tracking robustness under camera motion, abrupt viewpoint changes, and fast-moving platforms, which are common in aerial traffic surveillance.

Using motion continuity rather than appearance embeddings, OC-SORT achieves reliable identity preservation with low computational complexity. This makes it well-suited for large-scale traffic video analysis and real-time deployment scenarios, while providing more stable trajectories than other trackers in the presence of camera motion.

Unless otherwise specified, the OC-SORT tracking parameters, such as the IoU association threshold, maximum track age, and minimum initialization hits, follow the default configuration of the original implementation. No dataset-specific hyperparameter tuning is applied in our experiments, ensuring fair, consistent, and reproducible comparisons across all evaluated scenarios.

### 3.5. Trajectory-Based Vehicle Counting

Vehicle counting in this study is performed using a trajectory-based counting strategy rather than frame-level detection counting. By leveraging continuous object trajectories obtained from the multi-object tracking module, each vehicle is counted only once based on its motion history, effectively mitigating overcounting caused by repeated detections across consecutive frames.

The proposed approach adopts a tracking-based virtual gate mechanism, where predefined spatial regions are placed within the scene to monitor vehicle flow. These virtual gates may correspond to lane boundaries, road segments, or entry–exit lines, depending on the camera viewpoint and traffic layout. The location and orientation of each virtual gate remain fixed throughout the video sequence and serve as reference regions for counting vehicle crossings.

In this study, virtual gates are manually defined for each video sequence based on lane orientation and dominant traffic flow direction. They are not dynamically adjusted or automatically generated during evaluation.

Let  $G$  denote a predefined virtual gate region, and let  $T_k = \{b_k^1, b_k^2, \dots\}$  represent the trajectory of the  $k$ -th vehicle. The total vehicle count is computed as

$$C = \sum_{k=1}^K \mathbb{I}(\exists t \text{ s.t. } T_k(t) \cap G \neq \emptyset), \quad (6)$$

where  $\mathbb{I}(\cdot)$  represents an indicator function, taking the value 1 if the condition is satisfied and 0 if it is not. This formulation ensures that each vehicle contributes at most one count, even if it remains near or within the gate region for multiple frames.

To further improve counting reliability, short-lived or noisy trajectories are filtered during the counting stage, suppressing spurious tracks that may arise from false detections or brief localization errors. By decoupling tracking performance from counting logic, the proposed strategy enables robust and scalable vehicle flow estimation in dense traffic scenes.

Overall, the trajectory-based virtual gate approach provides a stable and computationally efficient solution for vehicle counting, making it well-suited for high-density traffic monitoring and UAV-based surveillance scenarios where accurate and consistent counting is critical.

For trajectory-based vehicle counting, a minimum trajectory length constraint is applied to suppress short-lived or unstable tracks. A vehicle is counted once when its refined trajectory intersects the predefined virtual gate region. These counting parameters remain fixed across all datasets, and no dataset-specific adjustment or tuning is performed.

### 3.6. Feature Representation

Feature extraction in the proposed framework is performed implicitly within the detector and tracking modules rather than as a standalone component. Spatial visual features are learned through intermediate convolutional layers of the YOLOv10-S backbone, enabling robust multi-scale representation of vehicles across varying resolutions and viewpoints. Temporal information is implicitly captured through persistent track identities generated by the OC-SORT tracker, which aggregates spatial observations across frames to improve temporal consistency.

No explicit feature-level supervision or standalone feature evaluation is conducted, as feature learning is reflected indirectly through detection accuracy, tracking stability, and vehicle counting performance.

### 3.7. Baseline Models for Comparison

To evaluate the effectiveness of the proposed framework, several state-of-the-art object detectors are selected as baseline models, including YOLOv5s [34], YOLOv8n [35], YOLOv10-M [14], and RT-DETR-L [36]. These models represent different detection paradigms, ranging from lightweight convolutional detectors to transformer-based architectures. To ensure a fair comparison, all baseline models follow the same training setup, including dataset splits, class definitions, input resolutions, and data augmentation. YOLOv10-S is used as the primary detection backbone, while YOLOv10-M and RT-DETR-L are additionally employed as teacher models in the knowledge distillation experiments.

For multi-object tracking and vehicle counting, OC-SORT [15] is adopted as the primary tracking algorithm, while ByteTrack [27], BoT-SORT [28], and DeepSORT [26] are evaluated as baseline trackers to analyze the impact of different motion association strategies on tracking and counting performance.

### 3.8. Training Strategy Across Datasets

The proposed framework is trained and evaluated using multiple benchmark datasets representing distinct sensing modalities and traffic viewpoints. UA-DETRAC-DET and VisDrone-DET are employed for vehicle detection experiments, enabling evaluation under both ground-based surveillance and aerial remote-sensing conditions. For tracking and counting analysis, the corresponding multi-object tracking benchmarks, UA-DETRAC-MOT and VisDrone-MOT, are used to ensure compatibility with standard MOT evaluation protocols. A unified training strategy is adopted across datasets, where the same detection architecture and learning framework are maintained while dataset-specific characteristics are handled through appropriate input resolution and supervision. This design avoids architectural bias and enables fair comparison across datasets. Both intra-dataset and cross-domain evaluation settings are considered to analyze generalization capability when transferring models between ground-level and aerial traffic scenes.

It is important to note that within the proposed framework, only the detection module (YOLOv10-S) undergoes training. The OC-SORT tracking algorithm and the trajectory-based counting strategy are deterministic inference-stage components and do not involve learnable parameters or participate in the training process. This design ensures modular separation between detection learning and downstream tracking and counting evaluation.

The proposed methodology integrates vehicle detection, tracking, and counting within a unified yet modular framework designed for both ground-based surveillance and aerial traffic monitoring. By maintaining architectural consistency, applying cross-dataset evaluation, and using a common tracking and counting strategy, the framework enables a rigorous and fair assessment of real-world applicability for Intelligent Transportation Systems.

## 4. Experiments and Results

In this section, we outline the experimental configuration and quantitative evaluation methods used to evaluate the performance of the proposed vehicle detection, tracking, and counting framework. The present work is evaluated separately using task-appropriate benchmarks and metrics, following the methodological design described in Section 3. All evaluations are conducted on unseen test data to avoid bias and ensure reliable performance assessment.

### 4.1. Datasets and Class Mapping

The robustness of the proposed framework is examined through experiments on two widely used public datasets that reflect different traffic monitoring viewpoints: UA-DETRAC [16], a ground-based fixed-camera dataset, and VisDrone, an aerial drone-

based dataset. Together, these datasets make it possible to evaluate model performance across different environmental conditions, viewpoints, and object scales. Representative examples from both datasets are shown in Figure 2, highlighting the fundamental differences between ground-level and aerial traffic monitoring scenarios.



**Figure 2.** Representative ground-based (UA-DETRAC benchmark) and aerial (VisDrone benchmark) traffic scenes used for evaluating vehicle detection, tracking, and counting across heterogeneous viewpoints.

#### 4.1.1. UA-DETRAC

The UA-DETRAC dataset [16] is a large-scale benchmark designed for multi-object vehicle detection and tracking in real-world traffic surveillance scenarios, jointly released by the University of Alberta and the University of Science and Technology of China, and is widely used in intelligent transportation research. It consists of long video sequences captured by fixed traffic cameras deployed at urban intersections and highways. The video sequences are recorded at 25 frames per second with a resolution of  $960 \times 540$  pixels and represent a wide spectrum of traffic conditions, including variations in lighting, weather (daylight, night, rainy), and nighttime scenarios. The dataset defines four vehicle categories: car, bus, van, and a generic other class.

For detection experiments in this study, the original UA-DETRAC-DET annotations are converted into the YOLO format to support unified training across all detection models. Images without valid bounding box annotations are excluded during preprocessing to ensure label consistency. To reduce temporal redundancy and mitigate overfitting caused by highly correlated consecutive frames, a temporal sampling strategy is applied by selecting one frame out of every five consecutive frames. After applying this sampling strategy, the processed UA-DETRAC-DET dataset consists of 18,810 training images, 4304 validation images, and 4562 test images, resulting in a total of 27,676 images. These splits remain fixed across all experiments to ensure fair and reproducible comparisons.

For tracking and vehicle counting, the UA-DETRAC-MOT benchmark is used, which preserves the original video sequences and identity annotations. This dataset enables trajectory-level evaluation under standard MOT protocols and is used exclusively for tracking and counting experiments, ensuring task-appropriate evaluation on continuous video data.

#### 4.1.2. VisDrone

The VisDrone dataset [17] is used to evaluate the proposed framework under aerial surveillance conditions. Collected using UAV-mounted cameras by the AISKYEYE research group at Tianjin University, the dataset supports object detection and tracking tasks across diverse real-world scenes, including urban roads, residential areas, highways, and open

spaces. Images are captured under varying lighting and weather conditions, with resolutions ranging from  $1920 \times 1080$  to  $2000 \times 1500$  pixels, posing significant challenges for detecting small and densely distributed objects.

In this study, we focus exclusively on vehicle-related categories relevant to traffic analysis, including cars, vans, trucks, buses, and motorcycles, while excluding non-vehicular object classes. Unlike UA-DETRAC-DET, the VisDrone-DET dataset already provides sparsely sampled and diverse frames suitable for detection tasks. Therefore, no additional temporal down-sampling was applied, and all available frames were retained during training and evaluation. After preprocessing and ensuring annotation quality, the VisDrone-DET dataset used for evaluation includes 6286 training, 543 validation, and 1578 test images, summing to 8407 images.

For tracking and vehicle counting evaluation, the VisDrone-MOT dataset is used, which provides sequence-level annotations with consistent object identities across frames. This dataset is specifically designed to evaluate tracking performance under aerial conditions characterized by camera motion, scale variation, and dense object distributions. By using VisDrone-MOT for tracking and counting, the proposed framework is evaluated under realistic UAV surveillance scenarios where temporal consistency and motion robustness are critical.

For both datasets, the test splits consist of unseen data and are used exclusively for final performance evaluation, while the validation splits are employed only for model selection and hyperparameter tuning. This experimental protocol ensures unbiased assessment and prevents any form of data leakage.

#### 4.2. Implementation Details

Experiments were performed on a high-performance computing server equipped with four NVIDIA A100 GPUs (80 GB HBM2 each), providing sufficient resources for large-scale deep learning training and evaluation. All runs utilized CUDA-based GPU acceleration with CUDA 12.8 and NVIDIA driver 545.23.08. The software environment was configured using PyTorch 2.9.0, running within a Python-based virtual environment. The underlying system architecture is x86\_64, powered by a multi-socket Intel CPU configuration with 96 logical processing cores, enabling efficient data loading and parallel processing during training.

For vehicle detection, all models were trained using the AdamW optimizer together with a cosine learning-rate schedule and a brief warm-up period. The initial learning rate was adjusted based on each model's capacity, while a consistent weight-decay value was used across all experiments to support regularization. Training was conducted for up to 200 epochs, with early stopping guided by validation performance to reduce overfitting. Lightweight YOLO-based models were trained with larger effective batch sizes of 32 to improve gradient stability, whereas transformer-based models were trained with smaller batch sizes of 16 due to higher memory requirements.

All models trained on the UA-DETRAC-DET dataset employed an input image size of 960 pixels, while for VisDrone-DET, a larger input resolution of 1344 pixels was used to better preserve small object details. Data augmentation was applied during training using conservative strategies suitable for traffic scenes, including random horizontal flipping, limited mosaic augmentation, HSV jitter, scale variation, and random cropping, while validation and testing are performed without augmentation.

For tracking and counting experiments, the same trained detector weights are reused, and only the tracking algorithm is varied. This design isolates the effect of the tracking strategy and ensures that differences in tracking and counting performance are not influenced by detection variability.

### 4.3. Evaluation Metrics

The proposed vehicle detection–tracking–counting framework is evaluated using standard object detection and multi-object tracking metrics, complemented by counting accuracy and computational efficiency.

**Detection Metrics:** To evaluate detection performance, we employ standard metrics including Average Precision ( $AP$ ), mean Average Precision ( $mAP$ ), Precision ( $P$ ), Recall ( $R$ ), and F1-score.

Precision represents the ratio of true vehicle detections to the total number of predicted detections, as expressed in Equation (7)

$$P = \frac{TP}{TP + FP} \quad (7)$$

In this context,  $TP$  represents the count of true positive detections, whereas  $FP$  corresponds to the number of false positives.

Recall indicates how many of the actual vehicles present in the ground truth are successfully detected, as shown in Equation (8)

$$R = \frac{TP}{TP + FN} \quad (8)$$

where  $FN$  refers to false-negative detections, while Precision and Recall jointly explain the balance between incorrect detections and missed objects.

To provide a balanced measure that combines Precision and Recall, the F1-score is computed as defined in Equation (9)

$$F1 = \frac{2 \cdot P \cdot R}{P + R} \quad (9)$$

To assess detection performance across varying confidence thresholds, the Precision–Recall (P–R) curve is employed, and the Average Precision ( $AP$ ) is defined as the area beneath the curve for each class, as presented in Equation (10)

$$AP = \int_0^1 P(R) dR. \quad (10)$$

The  $mAP$  metric is calculated by averaging the Average Precision results for each vehicle category. Two evaluation settings are considered in this study. The first, denoted as  $mAP@0.5$ , counts a detection as correct as one where the predicted and ground-truth bounding boxes achieve an IoU above 0.5. The second,  $mAP@0.5:0.95$ , aggregates performance over multiple IoU thresholds ranging from 0.50 to 0.95 at intervals of 0.05, thereby providing a stricter assessment of localization accuracy.

**Tracking Metrics:** To evaluate multi-object vehicle tracking performance, we used MOTA, IDF1, and IDSW.

MOTA (Multiple Object Tracking Accuracy) measures overall tracking accuracy by combining the effects of missed detections, false positives, and identity switches into a single metric, as defined in Equation (11)

$$MOTA = 1 - \frac{\sum_t (FN_t + FP_t + IDSW_t)}{\sum_t GT_t} \quad (11)$$

where  $GT_t$  refers to the ground-truth object count at time  $t$ , and  $FN_t$ ,  $FP_t$ , and  $IDSW_t$  measure missed detections, false positives, and identity switches, respectively.

Identity F1-score (IDF1) measures the consistency of object identities throughout the tracking sequence and is defined as the harmonic mean of identity precision and identity recall, shown in Equation (12)

$$\text{IDF1} = \frac{2 \cdot \text{IDTP}}{2 \cdot \text{IDTP} + \text{IDFP} + \text{IDFN}}, \quad (12)$$

where  $\text{IDTP}$ ,  $\text{IDFP}$ , and  $\text{IDFN}$  denote the numbers of identity true positives, identity false positives, and identity false negatives, respectively.

In addition, the total number of identity switches ( $\text{IDSW}$ ) is reported to quantify the frequency of identity changes during tracking. A lower  $\text{IDSW}$  value indicates better temporal consistency, which is essential for accurate trajectory reconstruction and vehicle counting in dense traffic scenes.

Counting Metrics and Runtime Efficiency: Since accurate counting is a key objective in traffic monitoring applications, both error-based and accuracy-based metrics are employed.

The Absolute Counting Error (ACE) is defined as the normalized percentage difference between the predicted and ground-truth vehicle counts, as shown in Equation (13)

$$\text{ACE} = \frac{|C^{\text{pred}} - C^{\text{gt}}|}{C^{\text{gt}}} \times 100 \quad (13)$$

where  $C^{\text{gt}}$  and  $C^{\text{pred}}$  denote the ground-truth and predicted vehicle counts, respectively.

Counting Accuracy (CA) is used as a normalized metric for evaluating counting performance, as shown in Equation (14)

$$\text{CA} = 1 - \frac{|C^{\text{pred}} - C^{\text{gt}}|}{C^{\text{gt}}}. \quad (14)$$

Both ACE and CA are computed for each video sequence individually and then averaged across all evaluated sequences. Due to independent averaging across sequences, CA is not strictly equal to  $1 - \text{ACE}/100$  when reported in aggregated form.

The GT Count and Pred Count columns in the tables denote aggregated totals across sequences and are provided for reference only; they are not directly used to compute CA and ACE.

Lower ACE and higher CA values indicate more reliable vehicle counting performance.

In addition to accuracy, runtime performance is reported using the average inference latency per image, measured in milliseconds ( $\text{infer\_ms}$ ), and the corresponding processing speed in frames per second (FPS), defined as shown in Equation (15)

$$\text{FPS} = \frac{1000}{\text{infer\_ms}}. \quad (15)$$

These runtime metrics are obtained directly from the evaluation outputs of the testing scripts and indicate whether the system can operate under near real-time constraints in traffic surveillance scenarios.

#### 4.4. Detection Results and Comparison

##### 4.4.1. Results on UA-DETRAC-DET Dataset

Detection performance was first assessed on the UA-DETRAC-DET test set to evaluate the framework's effectiveness in ground-based traffic surveillance scenarios. For this dataset, RT-DETR was chosen as the teacher model for knowledge distillation because of its strong ability to capture global context and its proven robustness in fixed-camera traffic environments, while remaining compatible with the YOLOv10-S student model.

Overall Detection Performance: Table 1 summarizes the overall detection results on the UA-DETRAC-DET test set. Among all evaluated models, the YOLOv10-S variant distilled from RT-DETR achieves the strongest performance, reaching an mAP@0.5 of 77.54% and an mAP@0.5:0.95 of 62.02%, while still maintaining real-time inference at 160.46 FPS. Compared to the standard YOLOv10-S baseline, the distillation process yields improvements of 1.87 percentage points in mAP@0.5 and 1.74 percentage points in mAP@0.5:0.95, indicating more precise object localization under stricter IoU criteria without increasing computational cost.

When compared with YOLOv10-M, the distilled YOLOv10-S achieves comparable accuracy at a higher inference speed, making it more suitable for real-time traffic monitoring. RT-DETR-L demonstrates strong recall but suffers from significantly lower inference speed, limiting its practicality for real-time deployment. Lightweight baselines such as YOLOv5s and YOLOv8n achieve higher throughput, but exhibit reduced localization accuracy. Overall, the results indicate that supervision-level knowledge distillation enables YOLOv10-S to achieve a well-balanced trade-off between accuracy and efficiency.

**Table 1.** Overall quantitative comparison of detection performance on the UA-DETRAC-DET test set.

Model	mAP@0.5 (%)	mAP@0.5:0.95 (%)	P (%)	R (%)	FPS
YOLOv5s [34]	68.45	54.12	74.82	61.49	211.19
YOLOv8n [35]	73.49	57.97	85.15	65.52	<b>236.64</b>
YOLOv10-S [14]	75.67	60.28	<b>87.34</b>	65.64	160.63
YOLOv10-M [14]	75.53	59.73	83.75	69.07	141.14
RT-DETR-L [36]	77.00	60.03	76.71	<b>78.79</b>	37.56
YOLOv10-S (KD from YOLOv10-M) [14,33]	74.46	59.15	81.71	64.94	160.57
<b>YOLOv10-S (KD from RT-DETR) [14,33]</b>	<b>77.54</b>	<b>62.02</b>	85.14	71.07	160.46
Reference methods are included for contextual comparison, and their results are reported as in the original publications.					
YOLOv8-S [37]	64.20	46.60	71.80	58.60	–
YOLOv8-S+CG_Down+C2f_DRB+Soft-NMS [37]	73.20	55.40	82.40	62.30	–
YOLOv8-M [37]	64.00	49.70	–	–	135.00
YOLOv9-S [38]	65.10	48.00	–	–	–
Faster-RCNN [39]	66.30	38.80	–	–	39.00
RetinNet [40]	56.60	32.10	–	–	59.00
YOLO-FA [41]	70.00	50.30	–	–	163.00

Note: The best results in each metric are highlighted in bold.

Comparison with Reference Methods: Beyond the methods evaluated in this study, Table 1 also lists several representative reference detectors to provide contextual comparison. Within the same study by You et al. [37], YOLOv8-S achieves 64.20% mAP@0.5 and 46.60% mAP@0.5:0.95, its enhanced variant with CG-Down, C2f-DRB, and Soft-NMS improves these scores to 73.20% and 55.40%, and YOLOv8-M reports 64.0% mAP@0.5 at 135 FPS, however, the proposed YOLOv10-S (KD from RT-DETR) surpasses all these variants without introducing extra architectural components. Similarly, YOLOv9-S [38] achieves 65.1% mAP@0.5 and 48.0% mAP@0.5:0.95, remaining significantly below the proposed method, particularly under higher IoU evaluation criteria. Two-stage detectors such as Faster R-CNN [39] reach 66.3% mAP@0.5 but operate at only 39 FPS, limiting real-time applicability, while RetinaNet [40] shows weaker localization performance with an mAP@0.5:0.95 of 32.1%. Although YOLO-FA [41] achieves competitive speed (163 FPS), its localization accuracy remains lower than YOLOv10-S (KD from RT-DETR), particularly under stricter IoU thresholds. Overall, these comparisons show that the proposed method strikes a better balance between detection accuracy and real-time performance, without adding architectural complexity.

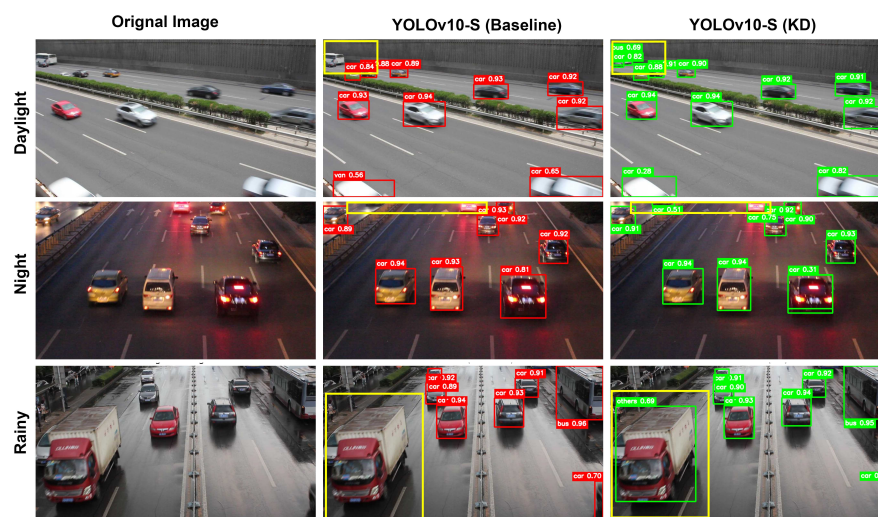
Class-Wise Detection Performance: As shown in Table 2, the class-wise results on the UA-DETRAC-DET test set indicate that YOLOv10-S distilled from RT-DETR achieves better performance than the baseline for every vehicle category, particularly at stricter IoU levels. For the car class, mAP@0.5 increased from 89.82% to 90.11% and mAP@0.5:0.95 from 65.25% to 65.78%, indicating more stable detection in dense traffic scenes. The improvement of

precision and recall indicates more stable detection and reduced missed vehicles in dense traffic scenarios. The bus category shows more pronounced improvements, with mAP@0.5 increasing from 81.87% to 86.03% and mAP@0.5:0.95 from 75.45% to 78.16%, reflecting better localization of larger vehicles under occlusion. Improvements for the van class are modest but consistent, particularly in recall (66.51% to 72.05%) and mAP@0.5:0.95 (58.20% to 58.57%). Overall, the results indicate that distillation from RT-DETR enhances localization robustness across both frequent and challenging vehicle classes without modifying the detector architecture, which is beneficial for downstream tracking and vehicle counting tasks.

**Table 2.** Class-wise detection performance on the UA-DETRAC-DET test set for YOLOv10-S and YOLOv10-S (KD from RT-DETR).

Class	YOLOv10-S (Baseline)				YOLOv10-S (KD from RT-DETR)			
	mAP@0.5 (%)	mAP@0.5:0.95 (%)	P (%)	R (%)	mAP@0.5 (%)	mAP@0.5:0.95 (%)	P (%)	R (%)
Car	89.82	65.25	90.14	77.61	90.11	65.78	86.05	81.71
Bus	81.87	75.45	88.27	74.95	86.03	78.16	93.74	78.08
Van	77.28	58.20	82.45	66.51	77.57	58.57	78.10	72.05

Figure 3 presents qualitative comparisons on representative UA-DETRAC-DET test images captured under daylight, nighttime, and rainy conditions. While the baseline YOLOv10-S (red boxes) performs well in clear scenes, it exhibits missed or weaker detections in challenging regions highlighted in yellow. The knowledge-distilled model (green boxes) improves detection completeness and localization stability, particularly under low illumination, motion blur, and wet-road reflections. Overall, the qualitative comparison confirms that supervision-level knowledge distillation enhances robustness across varying environmental conditions without altering the network architecture.



**Figure 3.** Visualization of baseline and knowledge-distilled YOLOv10-S detection results on the UA-DETRAC-DET test set under different environmental conditions (Daylight, Night, and Rainy).

#### 4.4.2. Results on VisDrone-DET Dataset

To assess how well the framework performs in aerial surveillance settings, detection experiments were conducted on the VisDrone-DET test set. VisDrone-DET presents a significantly more challenging scenario than the UA-DETRAC-DET dataset, due to high camera altitudes, large-scale variations, dense object distributions, and frequent occlusions. For VisDrone-DET, we selected YOLOv10-M as the teacher model for knowledge distillation due to its stronger performance under aerial imagery and scale variation, while preserving architectural compatibility with the YOLOv10-S student.

**Overall Detection Performance:** As shown in Table 3, the lightweight YOLO-based detectors maintain high inference speed but exhibit limited localization accuracy under these conditions, particularly at stricter IoU thresholds. Within this experiment, the baseline YOLOv10-S achieves 55.77% mAP@0.5 and 36.24% mAP@0.5:0.95 at 131.58 FPS, reflecting a reasonable balance between accuracy and speed. After applying knowledge distillation from YOLOv10-M, the student model improves to 59.14% mAP@0.5 and 38.79% mAP@0.5:0.95, corresponding to absolute gains of +3.37% and +2.55%, respectively, while simultaneously increasing inference speed to 174.57 FPS.

**Table 3.** Overall quantitative comparison of detection performance on the VisDrone-DET test set.

Model	mAP@0.5 (%)	mAP@0.5:0.95 (%)	P (%)	R (%)	FPS
YOLOv5s [34]	55.18	35.74	61.85	54.06	182.33
YOLOv8n [35]	51.91	33.33	60.26	50.79	217.14
YOLOv10-S [14]	55.77	36.24	63.42	53.23	131.58
YOLOv10-M [14]	59.09	38.77	64.58	56.61	85.45
RT-DETR-L [36]	54.56	35.83	68.09	55.59	27.05
<b>YOLOv10-S (KD from YOLOv10-M) [14,33]</b>	<b>59.14</b>	<b>38.79</b>	65.69	56.61	174.57
Reference methods are included for contextual comparison, and their results are reported as in the original publications.					
YOLOv8n [42]	26.70	14.80	38.40	29.20	130.00
RT-DETR-UAVs [42]	39.20	23.10	58.40	40.70	89.00
Zhang et al. [43]	50.80	31.00	63.10	49.30	–
FFCA-YOLO [44]	33.10	–	42.90	31.60	89.00
Gold-YOLO [45]	32.70	–	41.00	31.30	112.00
OSD-YOLOv10 [46]	33.40	–	43.90	32.50	136.00

Compared with earlier lightweight baselines such as YOLOv5s and YOLOv8n, which achieve even higher frame rates, exceeding 180 FPS, but their detection accuracy remains noticeably lower. In contrast, the heavier YOLOv10-M reaches similar detection accuracy (59.09% mAP@0.5) but operates at a much lower speed of 85.45 FPS. The distilled YOLOv10-S therefore achieves comparable accuracy with more than twice the throughput. Although RT-DETR-L shows strong recall performance at 55.59%, its low inference speed of 27.05 FPS poses a major challenge for real-time UAV deployment. Overall, the quantitative results demonstrate that distillation enables YOLOv10-S to narrow the accuracy gap with larger models while preserving a substantially higher inference rate, which is critical for real-time aerial traffic monitoring.

**Comparison with Reference Methods:** In addition to the methods evaluated in this work, Table 3 includes several representative reference detectors for contextual comparison. We can see that, compared to the distilled YOLOv10-S, YOLOv8n [42] achieves an mAP@0.5 of 26.7% and an mAP@0.5:0.95 of 14.8%, which are substantially lower, indicating limited robustness under aerial surveillance conditions. Similarly, RT-DETR-UAVs [42] improves detection accuracy to 39.2% mAP@0.5 and 23.1% mAP@0.5:0.95, but still remains significantly below the distilled model, particularly under stricter IoU evaluation criteria. Among CNN-based UAV-oriented detectors, Zhang et al. [43] demonstrated 50.8% mAP@0.5 and 31.0% mAP@0.5:0.95, achieving better localization accuracy compared to earlier lightweight approaches but still not better than our distilled YOLOv10-S. Other UAV-oriented detectors, including OSD-YOLOv10 [46], FFCA-YOLO [44], and Gold-YOLO [45], achieve moderate accuracy gains but do not outperform the proposed method in terms of the overall accuracy–efficiency trade-off. These comparisons further confirm that YOLOv10-S (KD from YOLOv10-M) achieves superior detection robustness without introducing additional architectural complexity.

**Class-Wise Detection Performance:** The class-wise detection performance of YOLOv10-S (Baseline) and YOLOv10-S (KD from YOLOv10-M) on the VisDrone-DET test set is represented in Table 4. In the Car class, knowledge-distilled YOLOv10-S demonstrates clear performance gains, with mAP@0.5 increasing by 2.14% and mAP@0.5:0.95 by 1.92%, accom-

panied by higher precision and recall. This indicates reliable detection of the most frequent vehicle type under aerial viewpoints. The Bus category also shows notable improvements, with mAP@0.5 increasing from 63.20% to 65.09% and mAP@0.5:0.95 from 46.36% to 47.99%, reflecting better localization of large vehicles in complex scenes. The distilled model improves both localization accuracy and recall for vans and trucks, which suffer from scale variation and background clutter in UAV imagery. Improvements are also seen for motorcycles, though performance remains limited due to their small size and weak visual cues. Overall, the results indicate that knowledge distillation consistently improves detection accuracy across all vehicle categories, especially for medium and large vehicles, particularly under the challenging conditions of aerial surveillance.

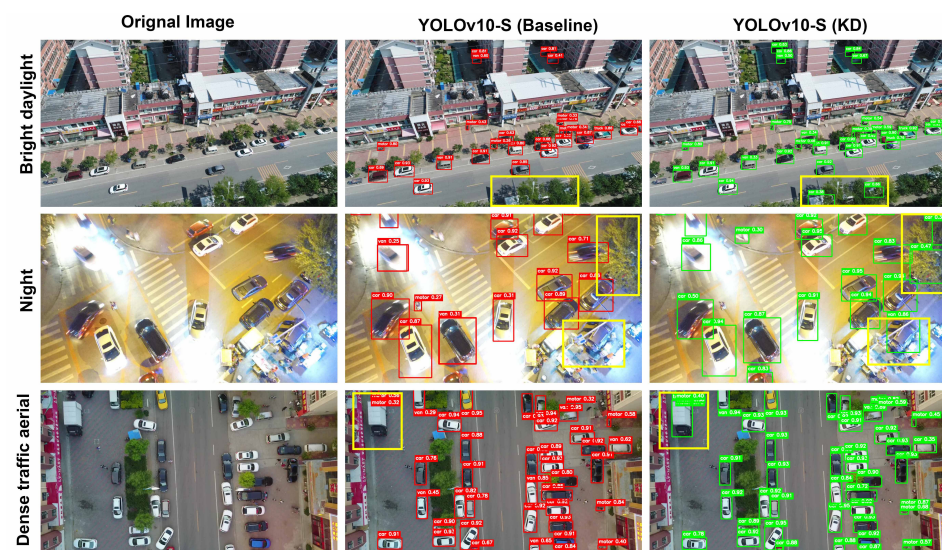
**Table 4.** Class-wise detection performance on the VisDrone-DET test set for YOLOv10-S and YOLOv10-S (KD from YOLOv10-M).

Class	YOLOv10-S (Baseline)				YOLOv10-S (KD from YOLOv10-M)			
	mAP@0.5 (%)	mAP@0.5:0.95 (%)	P (%)	R (%)	mAP@0.5 (%)	mAP@0.5:0.95 (%)	P (%)	R (%)
Car	80.97	53.23	76.36	77.27	<b>83.11</b>	<b>55.15</b>	<b>78.38</b>	<b>79.96</b>
Van	45.60	31.92	52.22	45.10	47.51	33.53	56.16	47.36
Truck	46.44	31.13	55.66	46.97	54.03	36.82	59.16	52.54
Bus	63.20	46.36	75.67	54.76	65.09	47.99	77.76	56.22
Motor	42.61	18.56	57.20	42.05	45.96	20.49	57.01	46.98

Note: The best results in each metric are highlighted in bold.

Overall, these quantitative characteristics make distilled YOLOv10-S a more reliable detection backbone under aerial surveillance conditions.

Figure 4 illustrates qualitative detection results on representative VisDrone-DET aerial scenes, including bright daylight, nighttime illumination, and dense traffic conditions. Compared to the baseline model, the distilled YOLOv10-S demonstrates improved small-object detection capability and more stable localization in high-density regions. In dense aerial scenes, the baseline model (red boxes) occasionally misses closely packed vehicles or produces fragmented detections, as highlighted in the yellow regions. The knowledge-distilled model (green boxes) reduces missed detections and maintains more consistent bounding box placement, particularly in occluded and cluttered environments. These qualitative observations further confirm the robustness and improved detection stability achieved through knowledge distillation.



**Figure 4.** Visualization of baseline and knowledge-distilled YOLOv10-S detection results on the VisDrone-DET test set across diverse aerial scenarios (Bright Daylight, Night, and Dense Traffic).

#### 4.5. Tracking Results

##### 4.5.1. Results on UA-DETRAC-MOT Dataset

Although the distilled YOLOv10-S achieves the best frame-level detection accuracy, tracking performance is primarily influenced by temporal detection consistency rather than per-frame localization alone; therefore, detector–tracker selection is evaluated independently in the following experiments. This subsection reports multi-object vehicle tracking results on the UA-DETRAC-MOT test split using 100 video sequences.

Table 5 compares different tracking algorithms using a fixed YOLOv10-S detector trained via knowledge distillation from RT-DETR, ensuring that performance differences are attributable to the tracker. OC-SORT stands out among the evaluated methods, showing the best overall performance with top IDF1 (88.50%) and MOTA (82.23%) scores, along with a comparatively low number of identity switches (888 IDSW). In contrast, BoT-SORT attains comparable IDF1 (88.36%) but exhibits substantially higher identity fragmentation (1975 IDSW), and ByteTrack shows lower overall accuracy. DeepSORT performs poorly in dense traffic scenarios, yielding a MOTA of only 26.01%. This behavior is primarily caused by frequent identity switches and trajectory fragmentation under dense traffic conditions, leading to significantly increased false positives and missed associations. These results indicate that OC-SORT provides the most reliable balance between tracking accuracy and identity stability under realistic traffic conditions.

**Table 5.** Comparison of multi-object tracking algorithms on the UA-DETRAC-MOT dataset using a fixed YOLOv10-S detector trained via knowledge distillation from RT-DETR.

Tracker	IDF1 (%) ↑	MOTA (%) ↑	IDSW ↓
ByteTrack [27]	87.08	75.87	869
BoT-SORT [28]	88.36	80.82	1975
DeepSORT [26]	71.42	26.01	1986
<b>OC-SORT [15]</b>	<b>88.50</b>	<b>82.23</b>	<b>888</b>

Note: ↑ indicates higher values are better, while ↓ indicates lower values are preferred. Bold values highlight the best performance, and the selected tracker used in our framework.

Further analyzes the impact of different detectors when paired with OC-SORT as present in the Table 6. YOLOv8n achieves the highest IDF1 (90.20%), while YOLOv10-S provides comparable tracking accuracy (IDF1 88.48%, MOTA 83.83%) with a favorable balance between performance and model complexity. The distilled YOLOv10-S variants exhibit mixed behavior, indicating that the benefits of knowledge distillation are not always reflected in isolated tracking metrics and are better assessed at the system level. Overall, the results support the selection of YOLOv10-S with OC-SORT as a stable and efficient configuration for downstream vehicle counting.

**Table 6.** Tracking performance of different detectors on UA-DETRAC-MOT when paired with the OC-SORT tracker.

Detector	IDF1 (%) ↑	MOTA (%) ↑	IDSW ↓
YOLOv5s	88.63	82.16	806
YOLOv8n	90.20	83.51	490
YOLOv10-S	88.48	83.83	986
YOLOv10-M	87.83	82.95	994
RT-DETR-L	87.35	84.65	1900
YOLOv10-S (KD from YOLOv10-M)	86.69	80.49	1224
YOLOv10-S (KD from RT-DETR)	88.50	82.23	888

Note: ↑ indicates higher values are better, while ↓ indicates lower values are preferred.

#### 4.5.2. Results on VisDrone-MOT Dataset

In this subsection, we report the results of multi-object vehicle tracking on the VisDrone-MOT test set using 17 video sequences.

Table 7 first compares different multi-object tracking algorithms on VisDrone-MOT using a fixed YOLOv10-S detector trained via knowledge distillation from YOLOv10-M, allowing the effect of the tracker to be evaluated independently of the detector. Among all evaluated trackers, OC-SORT shows the most balanced performance, combining the top IDF1 (46.45%) and MOTA (34.11%) with fewer identity switches (1031 IDSW) than ByteTrack and DeepSORT. BoT-SORT achieves a competitive IDF1 (41.75%) and fewer identity switches (573 IDSW), but its overall tracking accuracy remains lower than OC-SORT. DeepSORT performs poorly in this aerial scenario. In the VisDrone-MOT aerial environment, DeepSORT produces negative MOTA values due to severe identity fragmentation under rapid camera motion and scale variation. Since negative MOTA is dominated by accumulated detection and association errors and does not provide meaningful comparative insight, it is omitted from Table 7. These results confirm that OC-SORT provides more reliable identity association and temporal consistency for UAV-based traffic scenes.

**Table 7.** Comparison of multi-object tracking algorithms on the VisDrone-MOT dataset using a fixed YOLOv10-S detector trained via knowledge distillation from YOLOv10-M

Detector	IDF1 (%) ↑	MOTA (%) ↑	IDSW ↓
ByteTrack [27]	36.49	16.02	1426
BoT-SORT [28]	41.75	14.35	573
DeepSORT [26]	25.04	–	1896
<b>OC-SORT [15]</b>	<b>46.45</b>	<b>34.11</b>	<b>1031</b>

Note: ↑ indicates higher values are better, while ↓ indicates lower values are preferred. Bold values highlight the best performance, and the selected tracker used in our framework.

As shown in Table 8, the effect of using different detectors alongside OC-SORT is further examined. The results show that tracking performance varies substantially across detectors, reflecting differences in detection stability under aerial conditions. YOLOv10-S, when distilled from YOLOv10-M, demonstrates the best overall tracking performance, attaining an IDF1 score of 46.45% and a MOTA of 34.11%, and outperforming detectors across different model scales. In comparison, YOLOv8n achieves moderate performance with an IDF1 of 41.81%, while the baseline YOLOv10-S attains an IDF1 of 38.87%. Although RT-DETR-L provides strong detection capability, its tracking performance is limited by a high number of identity switches (1420 IDSW), indicating reduced identity stability in aerial scenes.

**Table 8.** Performance comparison of different detectors on the VisDrone-MOT dataset under the OC-SORT tracking framework

Detector	IDF1 (%) ↑	MOTA (%) ↑	IDSW ↓
YOLOv5s	42.22	29.50	<b>840</b>
YOLOv8n	41.81	30.44	854
YOLOv10-S	38.87	28.31	960
YOLOv10-M	35.99	24.26	1015
RT-DETR-L	36.20	16.32	1420
YOLOv10-S (KD from YOLOv10-M)	<b>46.45</b>	<b>34.11</b>	1031

Note: ↑ indicates higher values are better, while ↓ indicates lower values are preferred. Bold values highlight the best performance.

Overall, the results on the VisDrone dataset demonstrate that tracking performance in UAV-based traffic environments is highly sensitive to detection stability and temporal

consistency. By combining a knowledge-distilled YOLOv10-S detector with the OC-SORT tracker, the proposed framework achieves more stable trajectories and improved tracking accuracy under challenging aerial conditions.

#### 4.6. Counting Accuracy

##### 4.6.1. On UA-DETRAC-MOT Dataset

As shown in Table 9, vehicle counting performance on the UA-DETRAC-MOT dataset is evaluated using a trajectory-based virtual gate counting method built on the refined OC-SORT tracking pipeline. Compared with the other evaluated detectors, the RT-DETR–distilled YOLOv10-S achieves the most accurate counting results, with the highest CA (89.20%) and the lowest ACE (11.17%). Its predicted vehicle count (8753) is the closest to the ground-truth count (8256), indicating improved trajectory stability and reduced duplicate or missed counts. This demonstrates that knowledge distillation contributes to more consistent detections over time, which directly benefits trajectory-based vehicle counting. In contrast, the baseline YOLOv10-S exhibits lower counting accuracy, with a CA of 85.97% and an ACE of 14.95%, despite achieving strong detection performance. This degradation is primarily caused by trajectory fragmentation and identity instability, which lead to over-counting when vehicles cross the virtual gate multiple times. Similar trends are observed for other lightweight detectors, such as YOLOv5s and YOLOv8n, which achieve high inference speed but suffer from increased counting error due to less stable tracking.

**Table 9.** Vehicle counting performance on the UA-DETRAC-MOT dataset using trajectory-based virtual gate counting with the refined OC-SORT tracking pipeline.

Detector	CA (%) ↑	ACE (%) ↓	GT Count	Pred Count
YOLOv5s	88.97	11.45	8256	8804
YOLOv8n	88.46	12.48	8256	8755
YOLOv10-S	85.97	14.95	8256	8971
YOLOv10-M	86.67	13.84	8256	8912
RT-DETR-L	85.17	15.41	8256	9027
YOLOv10-S (KD from YOLOv10-M)	88.01	13.12	8256	8746
<b>YOLOv10-S (KD from RT-DETR)</b>	<b>89.20</b>	<b>11.17</b>	8256	8753

Note: ↑ indicates higher values are better, while ↓ indicates lower values are better. Bold values highlight the best results among the compared methods.

##### 4.6.2. Results on VisDrone-MOT Dataset

According to Table 10, YOLOv10-S distilled from YOLOv10-M demonstrates superior counting performance compared to other detectors, with a CA of 72.96% and an ACE of 27.03%. Its predicted vehicle count (1665) is the closest to the ground-truth count (2211), demonstrating improved trajectory smoothness and more consistent gate-crossing behavior under challenging aerial conditions. In contrast, baseline detectors exhibit substantially lower counting accuracy. YOLOv10-S achieves a CA of 63.48%, while YOLOv8n and YOLOv10-M produce comparable results with CA values of 60.90% and 60.65%, respectively. Although these detectors perform reasonably at the detection stage, their trajectories are more fragmented in aerial scenes, leading to missed or duplicate counts. RT-DETR-L shows the weakest counting performance, with a CA of 52.01% and the highest ACE (49.61%), indicating that strong detection capability alone does not guarantee reliable vehicle counting in UAV-based traffic environments.

Overall, the UA-DETRAC-MOT and VisDrone-MOT results highlight that vehicle counting accuracy is strongly influenced by trajectory stability rather than frame-level detection accuracy alone. By improving temporal detection consistency through knowledge

distillation and combining it with refined OC-SORT tracking, the proposed framework achieves significantly more reliable vehicle counting under ground-level and complex aerial surveillance conditions.

**Table 10.** Vehicle counting performance on the VisDrone-MOT dataset using trajectory-based virtual gate counting with the refined OC-SORT tracking pipeline

Detector	CA (%) ↑	ACE (%) ↓	GT Count	Pred Count
YOLOv5s	63.03	39.96	2315	1551
YOLOv8n	60.90	39.09	2301	1430
YOLOv10-S	63.48	36.51	2324	1491
YOLOv10-M	60.65	39.34	2353	1631
RT-DETR-L	52.01	49.61	2455	2523
<b>YOLOv10-S (KD from YOLOv10-M)</b>	<b>72.96</b>	<b>27.03</b>	2211	1665

Note: ↑ indicates higher values are better, while ↓ indicates lower values are better. Bold values highlight the best results among the compared methods.

#### 4.7. Cross-Dataset Generalization

To evaluate how well the proposed framework generalizes, we perform cross-domain experiments by directly transferring the trained model to a new domain without any retraining or domain-specific adjustments. The models are learned from a source dataset and assessed on a target dataset that presents notable differences in viewpoints and scene structure. Specifically, the UA-DETRAC dataset represents ground-based fixed-camera traffic surveillance, while the VisDrone dataset represents aerial UAV-based traffic scenes.

Table 11 reports cross-domain detection performance under direct domain transfer. When models trained on UA-DETRAC-DET are evaluated on VisDrone-DET, detection accuracy drops substantially due to severe differences in scale, perspective, and background complexity. The baseline YOLOv10-S achieves an mAP@0.5 of 12.41%, while the knowledge-distilled YOLOv10-S improves this to 14.85%. Similar trends are observed for stricter localization criteria, where mAP@0.5:0.95 increases from 8.60% to 10.67%. A similar trend is observed in the reverse direction. When trained on VisDrone-DET and evaluated on UA-DETRAC-DET, the baseline YOLOv10-S attains an mAP@0.5 of 28.13%, while the distilled model improves detection accuracy to 29.46%. These results suggest that knowledge distillation consistently enhances cross-domain detection robustness, even without retraining, by producing more stable and transferable feature representations.

**Table 11.** Cross-domain detection performance under direct train–test transfer without retraining.

Train → Test	Model	mAP@0.5 (%)	mAP@0.5:0.95 (%)	FPS
UA-DETRAC-DET → VisDrone-DET	YOLOv10-S (Baseline)	12.41	8.60	149.32
UA-DETRAC-DET → VisDrone-DET	YOLOv10-S (KD)	<b>14.85</b>	<b>10.67</b>	<b>155.22</b>
VisDrone-DET → UA-DETRAC-DET	YOLOv10-S (Baseline)	28.13	18.87	156.73
VisDrone-DET → UA-DETRAC-DET	YOLOv10-S (KD)	<b>29.46</b>	<b>20.26</b>	<b>157.14</b>

Note: The best results in each metric are highlighted in bold.

Further evaluates cross-domain performance for tracking and vehicle counting using a fixed OC-SORT tracker as shown in Table 12. Under UA-DETRAC-MOT → VisDrone-MOT transfer, tracking performance is limited, with IDF1 increasing from 8.72% to 13.12% when using the distilled detector, accompanied by an increase in counting accuracy from 12.50% to 17.31%. For the VisDrone-MOT → UA-DETRAC-MOT setting, tracking and counting performance are notably higher, with IDF1 improving from 55.24% to 57.96%, and corresponding improvements are observed in counting accuracy. Due to severe domain mismatch, MOTA becomes negative in several cross-domain tracking settings and is

therefore omitted, as it is dominated by detection errors and does not provide meaningful comparative insight in this scenario.

Overall, although absolute performance remains limited under direct cross-domain transfer, the results consistently show improved detection, tracking identity consistency, and vehicle counting accuracy when using the knowledge-distilled YOLOv10-S compared to the baseline model.

**Table 12.** Cross-domain tracking and vehicle counting performance under direct train–test transfer using OC-SORT with detectors trained on the source domain.

Train → Test	Model	IDF1 (%) ↑	MOTA (%) ↑	IDSW ↓	CA (%) ↑	ACE (%) ↓
UA-DETRAC-MOT → VisDrone-MOT	YOLOv10-S (Baseline)	8.72	4.90	149	12.50	87.50
UA-DETRAC-MOT → VisDrone-MOT	YOLOv10-S (KD)	<b>13.12</b>	<b>7.72</b>	219	<b>17.31</b>	<b>82.69</b>
VisDrone-MOT → UA-DETRAC-MOT	YOLOv10-S (Baseline)	55.24	–	2202	37.77	<b>88.28</b>
VisDrone-MOT → UA-DETRAC-MOT	YOLOv10-S (KD)	<b>57.96</b>	–	<b>2023</b>	34.27	102.86

Note: ↑ indicates higher values are better, while ↓ indicates lower values are better. The best results in each metric are highlighted in bold.

## 5. Discussion

This work investigates a unified detection–tracking–counting framework for traffic surveillance across heterogeneous environments, including ground-based fixed cameras and aerial UAV imagery. The experimental results demonstrate that reliable vehicle analysis requires not only strong frame-level detection but also stable temporal association, particularly for downstream tasks such as tracking and counting.

The detection experiments show that supervision-level knowledge distillation improves localization accuracy and robustness without modifying the detector architecture or increasing computational cost. Across both UA-DETRAC and VisDrone datasets, the distilled YOLOv10-S consistently outperforms its non-distilled counterpart under stricter IoU thresholds. Although teacher models generally possess greater representational capacity, a distilled student model can occasionally match or surpass the teacher on specific benchmarks. This behavior may be attributed to the regularizing effect of soft labels in knowledge distillation, which encourages smoother decision boundaries and mitigates overfitting. In certain scenarios, the larger teacher model may slightly overfit to the training distribution, whereas the lightweight student benefits from constrained capacity and refined supervision. Consequently, the distilled YOLOv10-S demonstrates competitive or superior performance under specific dataset conditions. However, the results also indicate that improved detection accuracy does not always lead to better isolated tracking performance, highlighting the need to evaluate detection and tracking independently within a unified pipeline. Tracking results on both datasets confirm the importance of trajectory stability and motion-aware association. OC-SORT consistently provides a strong balance between identity preservation and tracking accuracy under both fixed-camera and aerial settings, outperforming alternative trackers in challenging scenarios involving camera motion and dense traffic. These findings suggest that lightweight, observation-centric tracking strategies are well-suited for real-world traffic surveillance.

Vehicle counting experiments further emphasize that accurate counting is a system-level problem. Detectors with strong frame-level performance may still produce unreliable counts if trajectories are fragmented or unstable. On UA-DETRAC-MOT, the distilled YOLOv10-S combined with OC-SORT achieves the highest counting accuracy, while on VisDrone-MOT, counting performance remains more challenging due to aerial scene complexity. In both cases, improved temporal consistency leads to more reliable gate-based counting. Cross-domain experiments highlight the difficulty of directly transferring traffic models across drastically different viewpoints without retraining. Although absolute performance degrades under domain shift, the distilled YOLOv10-S consistently exhibits

improved robustness across detection, tracking, identity consistency, and vehicle counting. These results demonstrate that knowledge distillation can mitigate, but not fully overcome, cross-domain performance degradation without additional adaptation.

Overall, the findings indicate that a carefully designed, modular detection–tracking–counting framework can achieve reliable real-time performance across diverse traffic environments. Instead of introducing additional architectural complexity, the proposed approach emphasizes better supervision to support practical and scalable real-world traffic monitoring.

The findings highlight the importance of system-level integration and controlled evaluation rather than architectural modification. By jointly analyzing detection, tracking, and counting within a unified framework, we demonstrate how component interactions influence downstream performance and enable reproducible cross-domain comparison.

While the proposed framework does not introduce new architectural modules, it enables a structured component-wise evaluation through controlled experimental comparisons. Specifically, the effect of supervision-level knowledge distillation is assessed by comparing baseline and distilled YOLOv10-S detectors across detection, tracking, and counting metrics. The contribution of the tracking module is isolated by pairing the same detector with different tracking algorithms, allowing direct evaluation of identity association stability. Furthermore, the impact of the trajectory-based counting strategy is quantified by analyzing counting accuracy under consistent detection and tracking conditions. Collectively, these controlled comparisons function as an implicit ablation study, clearly demonstrating the individual and combined contributions of supervision enhancement, motion-centric tracking, and trajectory-level counting to the overall system performance.

## 6. Conclusions

Modern intelligent transportation systems are highly dependent on accurate vehicle detection, tracking, and counting. This work presents a single real-time framework capable of detecting, tracking, and counting vehicles in a wide range of traffic surveillance scenarios. By combining a lightweight YOLOv10-S detector with supervision-level knowledge distillation and a robust tracking-based counting strategy, the proposed framework achieves robust performance without relying on architectural modifications or dataset-specific tuning. Extensive evaluations on the UA-DETRAC and VisDrone benchmarks demonstrate that improving supervision quality and temporal consistency is critical for stable traffic analysis, particularly under challenging aerial and dynamic conditions. The results further indicate that knowledge distillation enhances detection robustness and contributes to more reliable counting behavior while preserving real-time efficiency. Cross-dataset experiments reveal the inherent difficulty of direct domain transfer; however, the proposed framework consistently shows improved robustness compared to non-distilled baselines, highlighting its potential for practical deployment where retraining may not be feasible.

As a next step, the framework will be extended with explicit vehicle classification [47] to provide more detailed traffic composition analysis and to support advanced applications such as congestion profiling and emission estimation. Additional research directions include improving cross-domain adaptability through lightweight self-supervised [48] or adaptive strategies [49], as well as integrating contextual cues and motion priors to further enhance trajectory stability and counting reliability in large-scale or highly congested traffic scenes. These extensions have the potential to broaden the applicability of the proposed framework to more complex and data-scarce traffic monitoring environments.

**Author Contributions:** Conceptualization, M.R.K.K.; methodology, M.R.K.K.; formal analysis, M.R.K.K.; investigation, M.R.K.K.; writing—original draft preparation, M.R.K.K.; writing—review

and editing, N.R. and M.R.K.K.; visualization, M.R.K.K.; supervision, N.R.; funding acquisition, N.R. All authors have read and agreed to the published version of the manuscript.

**Funding:** This material is based in part upon work supported by the National Science Foundation under Grant Nos. CNS-2018611 and CNS-1920182.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The datasets used in this study (UA-DETRAC dataset [16] and VisDrone dataset [17]) are publicly available from their respective official repositories.

**Acknowledgments:** The authors would like to thank the reviewers and the editor for their valuable comments and constructive suggestions, which helped improve the quality and clarity of this manuscript.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

1. Li, C.; Xu, P. Application on traffic flow prediction of machine learning in intelligent transportation. *Neural Comput. Appl.* **2021**, *33*, 613–624. [[CrossRef](#)]
2. Song, H.; Liang, H.; Li, H.; Dai, Z.; Yun, X. Vision-based vehicle detection and counting system using deep learning in highway scenes. *Eur. Transp. Res. Rev.* **2019**, *11*, 51. [[CrossRef](#)]
3. Madhavi, G.B.; Bhavani, A.D.; Reddy, Y.S.; Kiran, A.; Chitra, N.T.; Reddy, P.C.S. Traffic congestion detection from surveillance videos using deep learning. In Proceedings of the 2023 International Conference on Computer, Electronics & Electrical Engineering & Their Applications (IC2E3), Srinagar Garhwal, India, 8–9 June 2023; pp. 1–5.
4. Srivastava, S.; Narayan, S.; Mittal, S. A survey of deep learning techniques for vehicle detection from UAV images. *J. Syst. Archit.* **2021**, *117*, 102152. [[CrossRef](#)]
5. Hua, W.; Chen, Q. A survey of small object detection based on deep learning in aerial images. *Artif. Intell. Rev.* **2025**, *58*, 1–67. [[CrossRef](#)]
6. Xiao, J.; Cheng, H.; Sawhney, H.; Han, F. Vehicle detection and tracking in wide field-of-view aerial video. In Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 13–18 June 2010; pp. 679–684.
7. Li, X.; Li, X.; Pan, H. Multi-scale vehicle detection in high-resolution aerial images with context information. *IEEE Access* **2020**, *8*, 208643–208657. [[CrossRef](#)]
8. Liu, Z.; Zhang, W.; Gao, X.; Meng, H.; Tan, X.; Zhu, X.; Xue, Z.; Ye, X.; Zhang, H.; Wen, S.; et al. Robust movement-specific vehicle counting at crowded intersections. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Seattle, WA, USA, 13–19 June 2020; pp. 614–615.
9. Xiang, X.; Zhai, M.; Lv, N.; El Saddik, A. Vehicle counting based on vehicle detection and tracking from aerial videos. *Sensors* **2018**, *18*, 2560. [[CrossRef](#)] [[PubMed](#)]
10. Zhu, J.; Sun, K.; Jia, S.; Li, Q.; Hou, X.; Lin, W.; Liu, B.; Qiu, G. Urban traffic density estimation based on ultrahigh-resolution UAV video and deep neural network. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2018**, *11*, 4968–4981. [[CrossRef](#)]
11. Ye, J.; Fu, C.; Zheng, G.; Paudel, D.P.; Chen, G. Unsupervised domain adaptation for nighttime aerial tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 8896–8905.
12. Wang, K.; Pu, L.; Dong, W. Cross-domain adaptive object detection based on refined knowledge transfer and mined guidance in autonomous vehicles. *IEEE Trans. Intell. Veh.* **2023**, *9*, 1899–1908. [[CrossRef](#)]
13. Song, W.; Li, S.; Chang, T.; Hao, A.; Zhao, Q.; Qin, H. Cross-view contextual relation transferred network for unsupervised vehicle tracking in drone videos. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Snowmass Village, CO, USA, 1–5 March 2020; pp. 1707–1716.
14. Wang, A.; Chen, H.; Liu, L.; Chen, K.; Lin, Z.; Han, J.; Ding, G. Yolov10: Real-time end-to-end object detection. *Adv. Neural Inf. Process. Syst.* **2024**, *37*, 107984–108011.
15. Cao, J.; Pang, J.; Weng, X.; Khirodkar, R.; Kitani, K. Observation-centric sort: Rethinking sort for robust multi-object tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 9686–9696.
16. Wen, L.; Du, D.; Cai, Z.; Lei, Z.; Chang, M.C.; Qi, H.; Lim, J.; Yang, M.H.; Lyu, S. UA-DETRAC: A new benchmark and protocol for multi-object detection and tracking. *Comput. Vis. Image Underst.* **2020**, *193*, 102907. [[CrossRef](#)]

17. Zhu, P.; Wen, L.; Du, D.; Bian, X.; Fan, H.; Hu, Q.; Ling, H. Detection and tracking meet drones challenge. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *44*, 7380–7399. [[CrossRef](#)] [[PubMed](#)]
18. Chen, S.; Lin, W. Embedded system real-time vehicle detection based on improved YOLO network. In Proceedings of the 2019 IEEE 3rd Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC), Chongqing, China, 11–13 October 2019; pp. 1400–1403.
19. Zhang, Y.; Guo, Z.; Wu, J.; Tian, Y.; Tang, H.; Guo, X. Real-time vehicle detection based on improved yolo v5. *Sustainability* **2022**, *14*, 12274. [[CrossRef](#)]
20. Guo, H.; Zhang, Y.; Chen, L.; Khan, A.A. Research on vehicle detection based on improved YOLOv8 network. *arXiv* **2024**, arXiv:2501.00300. [[CrossRef](#)]
21. Bihanda, Y.G.; Faticah, C.; Yuniarti, A. Multi-vehicle tracking and counting framework in average daily traffic survey using rt-detr and bytetrack. *IEEE Access* **2024**, *12*, 121723–121737. [[CrossRef](#)]
22. Jin, Z.; Zhang, Q.; Gou, C.; Lu, Q.; Li, X. Transformer-based vehicle detection for surveillance images. *J. Electron. Imaging* **2022**, *31*, 051602–051602. [[CrossRef](#)]
23. Chang, Y.; Li, D.; Gao, Y.; Su, Y.; Jia, X. An improved YOLO model for UAV fuzzy small target image detection. *Appl. Sci.* **2023**, *13*, 5409. [[CrossRef](#)]
24. Liu, Y.; Sun, P.; Wergeles, N.; Shang, Y. A survey and performance evaluation of deep learning methods for small object detection. *Expert Syst. Appl.* **2021**, *172*, 114602. [[CrossRef](#)]
25. Du, D.; Zhu, P.; Wen, L.; Bian, X.; Lin, H.; Hu, Q.; Peng, T.; Zheng, J.; Wang, X.; Zhang, Y.; et al. VisDrone-DET2019: The vision meets drone object detection in image challenge results. In Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops, Seoul, Republic of Korea, 27–28 October 2019.
26. Wojke, N.; Bewley, A.; Paulus, D. Simple online and realtime tracking with a deep association metric. In Proceedings of the 2017 IEEE International Conference on Image Processing (ICIP), Beijing, China, 17–20 September 2017; pp. 3645–3649.
27. Zhang, Y.; Sun, P.; Jiang, Y.; Yu, D.; Weng, F.; Yuan, Z.; Luo, P.; Liu, W.; Wang, X. Bytetrack: Multi-object tracking by associating every detection box. In *Proceedings of the European Conference on Computer Vision*; Springer: Cham, Switzerland, 2022; pp. 1–21.
28. Aharon, N.; Orfaig, R.; Bobrovsky, B.Z. BoT-SORT: Robust associations multi-pedestrian tracking. *arXiv* **2022**, arXiv:2206.14651.
29. Lu, J.; Xia, M.; Gao, X.; Yang, X.; Tao, T.; Meng, H.; Zhang, W.; Tan, X.; Shi, Y.; Li, G.; et al. Robust and online vehicle counting at crowded intersections. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 21–25 June 2021; pp. 4002–4008.
30. Mandal, V.; Adu-Gyamfi, Y. Object detection and tracking algorithms for vehicle counting: A comparative analysis. *J. Big Data Anal. Transp.* **2020**, *2*, 251–261. [[CrossRef](#)]
31. Xu, H.; Lai, S.; Li, X.; Yang, Y. Cross-domain car detection model with integrated convolutional block attention mechanism. *Image Vis. Comput.* **2023**, *140*, 104834. [[CrossRef](#)]
32. Zhang, H.; Luo, G.; Li, J.; Wang, F.Y. C2FDA: Coarse-to-fine domain adaptation for traffic object detection. *IEEE Trans. Intell. Transp. Syst.* **2021**, *23*, 12633–12647. [[CrossRef](#)]
33. Gou, J.; Yu, B.; Maybank, S.J.; Tao, D. Knowledge distillation: A survey. *Int. J. Comput. Vis.* **2021**, *129*, 1789–1819. [[CrossRef](#)]
34. Jocher, G. *Ultralytics YOLOv5*; Zenodo: Geneva, Switzerland, 2020. [[CrossRef](#)]
35. Jocher, G.; Chaurasia, A.; Qiu, J. *Ultralytics YOLOv8*; Zenodo: Geneva, Switzerland, 2023.
36. Lv, W.; Zhao, Y.; Chang, Q.; Huang, K.; Wang, G.; Liu, Y. RTDETRv2: All-in-One Detection Transformer Beats YOLO and DINO. *arXiv* **2024**, arXiv:2407.17140. [[CrossRef](#)]
37. You, L.; Chen, Y.; Xiao, C.; Sun, C.; Li, R. Multi-object vehicle detection and tracking algorithm based on improved YOLOv8 and ByteTrack. *Electronics* **2024**, *13*, 3033. [[CrossRef](#)]
38. Wang, C.Y.; Yeh, I.H.; Mark Liao, H.Y. Yolov9: Learning what you want to learn using programmable gradient information. In *Proceedings of the European Conference on Computer Vision*; Springer: Cham, Switzerland, 2024; pp. 1–21.
39. Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
40. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal loss for dense object detection. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2980–2988.
41. Kang, L.; Lu, Z.; Meng, L.; Gao, Z. YOLO-FA: Type-1 fuzzy attention based YOLO detector for vehicle detection. *Expert Syst. Appl.* **2024**, *237*, 121209. [[CrossRef](#)]
42. Tan, L.; Liu, Z.; Liu, H.; Li, D.; Zhang, C. A real-time unmanned aerial vehicle (UAV) aerial image object detection model. In Proceedings of the 2024 International Joint Conference on Neural Networks (IJCNN), Yokohama, Japan, 30 June–5 July 2024; pp. 1–7.
43. Zhang, C.; Yang, J. Emsd-detr: Efficient small object detection for UAV aerial images based on enhanced RT-DETR model. *J. Supercomput.* **2025**, *81*, 1–33. [[CrossRef](#)]

44. Zhang, Y.; Ye, M.; Zhu, G.; Liu, Y.; Guo, P.; Yan, J. FFCA-YOLO for Small Object Detection in Remote Sensing Images. *IEEE Trans. Geosci. Remote Sens.* **2024**, *62*, 1–15. [[CrossRef](#)]
45. Wang, C.; He, W.; Nie, Y.; Guo, J.; Liu, C.; Wang, Y.; Han, K. Gold-YOLO: Efficient object detector via gather-and-distribute mechanism. *Adv. Neural Inf. Process. Syst.* **2023**, *36*, 51094–51112.
46. Zhang, Y.; Chen, X.; Sun, S.; You, H.; Wang, Y.; Lin, J.; Wang, J. Vehicle detection in drone aerial views based on lightweight OSD-YOLOv10. *Sci. Rep.* **2025**, *15*, 25155. [[CrossRef](#)]
47. Shokravi, H.; Shokravi, H.; Bakhary, N.; Heidarrezaei, M.; Rahimian Koloor, S.S.; Petru, M. A review on vehicle classification and potential use of smart vehicle-assisted techniques. *Sensors* **2020**, *20*, 3274. [[CrossRef](#)]
48. Zheng, Z.; Chen, Y.; Hua, B.S.; Wu, Y.; Yeung, S.K. Cross-domain autonomous driving perception using contrastive appearance adaptation. In Proceedings of the 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Detroit, MI, USA, 1–5 October 2023; pp. 3240–3247.
49. Li, J.; Xu, R.; Liu, X.; Ma, J.; Li, B.; Zou, Q.; Ma, J.; Yu, H. Domain adaptation based object detection for autonomous driving in foggy and rainy weather. *IEEE Trans. Intell. Veh.* **2024**, *10*, 900–911. [[CrossRef](#)]

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.