frontiers | Frontiers in Neurorobotics

TYPE Original Research
PUBLISHED 13 October 2025
DOI 10.3389/fnbot.2025.1681341

OPEN ACCESS

EDITED BY
Long Jin,
Lanzhou University, China

REVIEWED BY
Eduard Manziuk,
Khmelnytskyi National University, Ukraine
Neethu Subash,
National Institute of Technology,
Tiruchirappalli, India

*CORRESPONDENCE
Ahmad Jalal
✉ ahmadjalal@mail.au.edu.pk
Hui Liu
✉ hui.liu@uni-bremen.de

†These authors have contributed equally to this work.

RECEIVED 07 August 2025
ACCEPTED 17 September 2025
PUBLISHED 13 October 2025

CITATION
Alshehri M, Wu T, Almujally NA, AlQahtani Y, Hanzla M, Jalal A and Liu H (2025) UAV-based intelligent traffic surveillance using recurrent neural networks and Swin transformer for dynamic environments.
Front. Neurorobot. 19:1681341.
doi: 10.3389/fnbot.2025.1681341

UAV-based intelligent traffic surveillance using recurrent neural networks and Swin transformer for dynamic environments

Mohammed Alshehri[1†], Ting Wu[2†], Nouf Abdullah Almujally[3†], Yahya AlQahtani[4†], Muhammad Hanzla[5†], Ahmad Jalal[5,6*†] and Hui Liu[7,8,9*†]

[1]Department of Computer Science, King Khalid University, Abha, Saudi Arabia, [2]Department of Otorhinolaryngology Head and Neck Surgery, Nanjing Tongren Hospital, School of Medicine, Southeast University, Nanjing, China, [3]Department of Information System, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia, [4]Department of Informatics and Computer Systems, King Khalid University, Abha, Saudi Arabia, [5]Department of Computer Science, Air University, Islamabad, Pakistan, [6]Department of Computer Science and Engineering, College of Informatics, Korea University, Seoul, Republic of Korea, [7]Jiangsu Key Laboratory of Intelligent Medical Image Computing, School of Artificial Intelligence (School of Future Technology), Nanjing University of Information Science and Technology, Nanjing, China, [8]Guodian Nanjing Automation Co., Ltd., Nanjing, China, [9]Cognitive Systems Lab, University of Bremen, Bremen, Germany

Introduction: Urban traffic congestion, environmental degradation, and road safety challenges necessitate intelligent aerial robotic systems capable of real-time adaptive decision-making. Unmanned Aerial Vehicles (UAVs), with their flexible deployment and high vantage point, offer a promising solution for large-scale traffic surveillance in complex urban environments. This study introduces a UAV-based neural framework that addresses challenges such as asymmetric vehicle motion, scale variations, and spatial inconsistencies in aerial imagery.

Methods: The proposed system integrates a multi-stage pipeline encompassing contrast enhancement and region-based clustering to optimize segmentation while maintaining computational efficiency for resource-constrained UAV platforms. Vehicle detection is carried out using a Recurrent Neural Network (RNN), optimized via a hybrid loss function combining cross-entropy and mean squared error to improve localization and confidence estimation. Upon detection, the system branches into two neural submodules: (i) a classification stream utilizing SURF and BRISK descriptors integrated with a Swin Transformer backbone for precise vehicle categorization, and (ii) a multi-object tracking stream employing DeepSORT, which fuses motion and appearance features within an affinity matrix for robust trajectory association.

Results: Comprehensive evaluation on three benchmark UAV datasets—AU-AIR, UAVDT, and VAID shows consistent and high performance. The model achieved detection precisions of 0.913, 0.930, and 0.920; tracking precisions of 0.901, 0.881, and 0.890; and classification accuracies of 92.14, 92.75, and 91.25%, respectively.

Discussion: These findings highlight the adaptability, robustness, and real-time viability of the proposed architecture in aerial traffic surveillance applications.

By effectively integrating detection, classification, and tracking within a unified neural framework, the system contributes significant advancements to intelligent UAV-based traffic monitoring and supports future developments in smart city mobility and decision-making systems.

KEYWORDS

neural networks, unmanned aerial vehicle, multi-object tracking, adaptive control, swin transformer, autonomous systems

1 Introduction

Recent breakthroughs in deep learning, transformer architectures, and attention-driven vision models have substantially advanced object detection, classification, and multi-object tracking in complex visual environments (Sundaresan et al., 2024; Zhang et al., 2025). These algorithmic advances underpin modern autonomous systems and intelligent transportation infrastructures by enabling improved vehicle identification and behavior analysis under varying conditions (Qureshi et al., 2023). Nevertheless, traditional ground-based traffic surveillance relies on fixed cameras and infrastructure suffers from occlusion, limited spatial coverage, and low adaptability to diverse urban layouts (Vu et al., 2025; Mei and Zhu, 2024; Hanzla and Jalal, 2025). Such constraints limit their usefulness for large-scale, flexible monitoring required by smart-city and emergency-response applications.

Unmanned Aerial Vehicles (UAVs) provide a complementary sensing modality that addresses many of these limitations by offering mobile, high-resolution observation with dynamic field-of-view control (Hanzla et al., 2024). When combined with deep learning and reinforcement-based control strategies, UAVs enable scalable video understanding and adaptive data collection for traffic monitoring. At the same time, aerial imagery imposes unique technical challenges cluttered backgrounds, irregular object scales, asymmetric spatial distributions, and dynamic motion which complicate detection and raise computational demands (Ammour et al., 2017). These factors motivate solutions that are both robust to visual artifacts and efficient enough for onboard or near-edge operation.

In response, we present a symmetry-aware UAV traffic monitoring framework that explicitly targets the principal challenges of aerial surveillance. The pipeline begins with Fast Adaptive Mean-Variance Normalization (FAMVN) to reduce illumination bias and background effects, followed by spectral–spatial segmentation (SASSC) to sharpen object boundaries and suppress clutter (Mujtaba et al., 2024a,b; Almujally et al., 2024). To exploit temporal continuity in videos, we employ an RNN-based temporal detector with attention mechanisms that improve bounding-box alignment and reduce transient false positives. Detected candidates are processed in parallel by (i) DeepSORT for identity-preserving trajectory estimation and (ii) a classification branch that fuses handcrafted descriptors (SURF + BRISK) with a Swin Transformer to handle scale variability and class imbalance. This dual-path design leverages complementary strengths preprocessing and segmentation to improve proposals, temporal modeling to enforce detection continuity, and hybrid feature fusion to boost classification reliability yielding robust detection, tracking stability, and improved classification for aerial sequences.

Key contributions of this work are:

- Traditional background subtraction and elimination techniques are ineffective for aerial imagery due to continuously changing and dynamic backgrounds. To overcome this limitation, we adopted the Self-Adaptive Spectral-Spatial Clustering (SASSC) algorithm, which clusters pixels with high spectral-spatial correlation, effectively isolating relevant regions.
- For vehicle detection, we employed a Recurrent Neural Network (RNN), a temporal deep learning architecture well-suited for modeling sequential dependencies across video frames, enabling accurate identification of small-scale vehicles in aerial views where objects appear tiny, occluded, and densely clustered.
- The system is designed to support both vehicle tracking and classification. Moving and stationary vehicles are first distinguished to ensure only dynamic vehicles are tracked. DeepSORT is used as the tracking module, leveraging both motion and appearance features to maintain consistent identities across frames.
- Simultaneously, each detected vehicle undergoes feature extraction. The extracted features are then processed through a Swin Transformer, which classifies each vehicle into its respective category with high accuracy.

The remainder of the paper is structured as follows: Section 2 presents related work; Section 3 explains the system architecture; Section 4 discusses experimental results and comparisons; Section 5 provides a detailed discussion; and Section 6 concludes the study and outlines future research directions.

2 Literature review

Effective road traffic monitoring has been a critical area of research, with numerous approaches proposed to enhance vehicle detection and tracking. This section provides a comprehensive overview of recent advancements, highlighting key methodologies, challenges, and emerging trends in the field.

2.1 Traditional methods in vehicle detection and tracking

(Yang and Qu, 2018)introduced a real-time surveillance framework leveraging a background subtraction model based on low-rank and sparse decomposition to detect moving vehicles, followed by an online Kalman filter for object tracking and counting. This classical pipeline demonstrates high robustness in complex scenes involving overlapping vehicles or visual clutter (Moutakki et al., 2017) developed a vehicle counting system using a codebook-based background model with occlusion handling. Vehicle regions are segmented via background subtraction, refined through contour filtering, and classified using HOG descriptors and an SVM, achieving comprehensive detection accuracy on highway data (Nosheen et al., 2024) proposed a lightweight approach combining blob detection with a Kernelized Correlation Filter (KCF) tracker. The system applies preprocessing (gamma correction, bilateral filtering, and mean-shift) to enhance vehicle blobs, which are then tracked using KCF, achieving around 82% detection and 86% tracking accuracy on the KITTI dataset. These studies highlight the continued relevance of classical methods such as Kalman and KCF tracking in real-time surveillance. Many other approaches utilize sequential filtering (e.g., Gaussian blur, morphological operations) or optical flow, followed by simple trackers. Traditional background subtraction models like Codebook or Mixture of Gaussians, when paired with contour filtering and trackers (e.g., Kalman, Mean-Shift, or particle filters), remain computationally efficient and interpretable, though they may degrade under severe occlusion or dynamic lighting.

2.2 Machine learning-based traffic scene analysis

Arinaldi et al. (2018) compared a classical Mixture-of-Gaussians (MoG) background model paired with HOG features and an SVM classifier against a deep learning-based Faster R-CNN. While MoG+SVM effectively segmented and classified vehicles, it under-performed in scenarios with occlusions or stationary vehicles, where Faster R-CNN proved more robust. ElSahly and Abdelfatah (2023) developed a Random Forest (RF)-based traffic incident detection system using VISSIM-generated simulations incorporating con-gestion, incident severity, and sensor data. The RF classifier accurately distinguished between incident and normal traffic states, demonstrating the value of classical ML in high-level anomaly detection. Other ML-based approaches leverage hand-crafted features such as HOG or Haar cascades with SVMs or boosting to estimate traffic flow, classify vehicle types, or detect static anomalies. Though computationally efficient, these methods often suffer from limited robustness under viewpoint or illumination changes.

Alharbi (2022) applied RF for vehicle detection in UAV imagery, achieving 87.4% recall under challenging conditions but facing performance drops with dense traffic and high-dimensional features. Meanwhile, Chandramohan et al. (2020) addressed the traffic surge in urban environments, emphasizing the importance of efficient vehicle clustering in VANETs to reduce redundant communication, maintain data integrity, and support real-time traffic management. Wang et al. (2022) address the critical challenge of maximizing fresh information collected by Unmanned Aerial Vehicles (UAVs) from fixed-point devices within complex forest environments, a task traditionally handled by human patrols but better suited for UAVs despite the limitations of existing path planning methods in such intricate settings. They propose two distinct methodologies: for two-point path planning, they employ a chaotic initialization and co-evolutionary algorithm, carefully considering key UAV performance and environmental factors and for multi-point path planning, they introduce a method based on simulated annealing. Experimental validation, which involved using benchmark functions for parameter configuration and comparing their approach against existing strategies on both simple and complex simulated maps, demonstrated that their proposed techniques effectively generate UAV patrol paths that achieve higher information freshness with fewer iterations and lower computational costs, thereby underscoring their practical value. Bianchi et al. (2024) proposed an energy-optimal reference generator combined with a hierarchical control strategy for quadrotor trajectory planning. Their approach explicitly minimizes energy expenditure while preserving system stability, showing that real-time feasible control can be achieved with substantial reductions in energy consumption. By optimizing trajectory references at the control level, the framework provides a foundation for energy-conscious UAV deployment, particularly valuable for long-duration or resource-limited aerial missions. This study illustrates how energy considerations can be embedded directly into UAV control architectures for efficient real-world operations.

2.3 Deep learning-based methods

The emergence of deep learning has significantly advanced vehicle detection, classification, and tracking. One-stage detectors like YOLO, particularly YOLOv3 and its successors, have become widely adopted for real-time applications due to their balance between speed and accuracy. However, YOLOv3 struggles with occluded or small-scale objects. Two-stage models such as Faster R-CNN, when integrated with tracking modules like Kalman filters and the Hungarian algorithm (as in the SORT framework), provide better multi-object tracking in dynamic scenes, though they lack appearance-based discrimination, which affects performance in dense traffic.

Gallo et al. (2023) employed YOLOv7 for UAV-based detection, achieving strong precision for small objects, albeit at the cost of high computational demand, limiting deployment on edge devices. Improvements such as k-means clustering for anchor box generation and multi-scale feature fusion have further enhanced earlier YOLO versions; for instance, Sang et al. (2018) proposed BIT-Vehicle dataset using an optimized YOLOv2 model. Similarly, YOLOv5 has demonstrated superior accuracy over YOLOv3/4, especially on aerial datasets, as noted by Nepal and Eslamiat (2022). Recent trends include ensemble models combining EfficientDet and YOLO variants to improve robustness under occlusion and scale variation. Transformer-based detectors and networks like YOLOv8 continue to push performance boundaries, while

two-stage models such as Mask R-CNN provide fine-grained segmentation for precise vehicle shape extraction in cluttered scenes. Overall, deep CNN-based detectors both single-stage and two-stage remain the foundation of modern vehicle detection pipelines, consistently achieving state-of-the-art performance on benchmarks like KITTI, UA-Detrac, and UAVDT.

Battal et al. (2023) used YOLOv5m6 on real-world traffic video sequences to detect and classify five vehicle categories, reporting an average detection/classification accuracy of 88% across diverse conditions. Xu H. et al. (2020); Xu M. et al. (2020) introduced an improved multitask-cascaded CNN (IMC-CNN) with mixed image enhancement techniques for aerial vehicle detection; although they improved small-object recall, the detection precision plateaued at 85% and Biyik et al. (2023) trained standard YOLOv3 and YOLOv4-CSP models on orthophotos from UAV imagery; they reported mAPs of 80 and 87%, respectively, underscoring the difficulty of detection in high-noise geo-referenced contexts . Collectively, these results illustrate that while current deep architectures can approach the lowest 90s in accuracy, they often remain below 91%, especially in outdoor, aerial, or real-time video scenarios further motivating the refined approach and higher scores ($\geq$0.92) achieved in our proposed system across AU-AIR, UAVDT, and VAID

2.4 Challenges in existing work

Despite significant progress in aerial traffic monitoring, several critical challenges limit the deployment of current methods on real-world, edge-based platforms. Traditional approaches often rely on static background modeling, which proves inadequate for UAV imagery, where dynamic perspectives and shifting backgrounds are the norm. The detection of small-scale vehicles remains problematic due to resolution constraints and occlusion, especially in cluttered environments. Moreover, many state-of-the-art deep learning models are computationally intensive, making them unsuitable for real-time inference on re-source-constrained, low-power UAV systems [6]. This severely impacts the scalability and cost-effectiveness of aerial surveillance in smart cities. Additionally, existing methods struggle to distinguish between moving and stationary vehicles, particularly under dense traffic and shadowed regions, leading to errors in trajectory tracking and event recognition. Class imbalance in datasets where dominant vehicle types overshadow minority classes further introduces bias, compromising classification reliability. These limitations highlight the urgent need for lightweight, energy-efficient, and edge-deployable solutions tailored for UAV platforms to ensure real-time, accurate, and scalable traffic intelligence in complex aerial scenarios.

3 Materials and method

3.1 System methodology

The novelty of the proposed approach lies in the systematic integration of multiple complementary methods into a single pipeline. Figure 1 illustrates this integration scheme: (i) preprocessing with FAMVN and contextual smoothing reduces illumination bias and noise, providing clean inputs for segmentation; (ii) segmentation with SASSC enforces spectral–spatial coherence, improving object boundary preservation; (iii) temporal RNN-based detection leverages motion continuity to reduce false positives; (iv) DeepSORT maintains consistent object identities across frames; (v) SURF and BRISK descriptors are fused to capture both scale-robust and illumination-robust features; and (vi) the Swin Transformer performs final classification by exploiting hierarchical self-attention to combine local and global cues. The combined use of these techniques enhances robustness by compensating for the limitations of any single method—for example, DeepSORT corrects missed detections from the RNN, while feature fusion strengthens classification under challenging aerial conditions. This synergistic integration directly contributes to improved detection precision, tracking stability, and classification accuracy, thereby advancing the feasibility of UAV-based real-time surveillance.

3.2 Image preprocessing via fast adaptive mean-variance normalization (FAMVN)

To improve data quality across the AU-AIR, VAID, and UAVDT datasets, we implemented an adaptive preprocessing pipeline integrating classical and advanced enhancement techniques (Mujtaba et al., 2024a,b). Rather than relying on traditional histogram methods, we applied Fast Adaptive Mean-Variance Normalization (FAMVN), which preserves structural features under varying lighting conditions. Frames 32,823 (AU-AIR), 6,000 (VAID), and 80,000 (UAVDT) were resized to 640 $\times$ 640 and pixel values normalized to [0,1] for model stability. Unlike CLAHE, FAMVN adjusts both local mean and variance within overlapping kernels, enhancing contrast while preserving edges and minimizing over-saturation. The FAMVN transformation for a pixel $x_{i,j}$ in window $w_{i,j}$ is defined as:

$$I'_{i,j} = \frac{I_{i,j} - \mu w_{i,j}}{\sigma w_{i,j} + \in} . \sigma_T + \mu_T \quad (1)$$

Here, $\mu \boldsymbol{w_{i,j}}$ and $\sigma \boldsymbol{w_{i,j}}$ denote the local mean and standard deviation within the sliding window $\boldsymbol{w_{i,j}}$, while μ_T and σ_T represent the global dataset statistics used as normalization targets. The small constant ϵ ensures numerical stability during division. This normalization aligns local intensity variations with global distribution statistics, thereby reducing illumination bias and enhancing structural consistency across frames. To further preserve edge information while suppressing noise, we apply Multiscale Contextual Smoothing using a Gaussian-integrated bilateral filter, which maintains salient object boundaries critical for robust detection and tracking in aerial imagery, as shown in Figure 2.

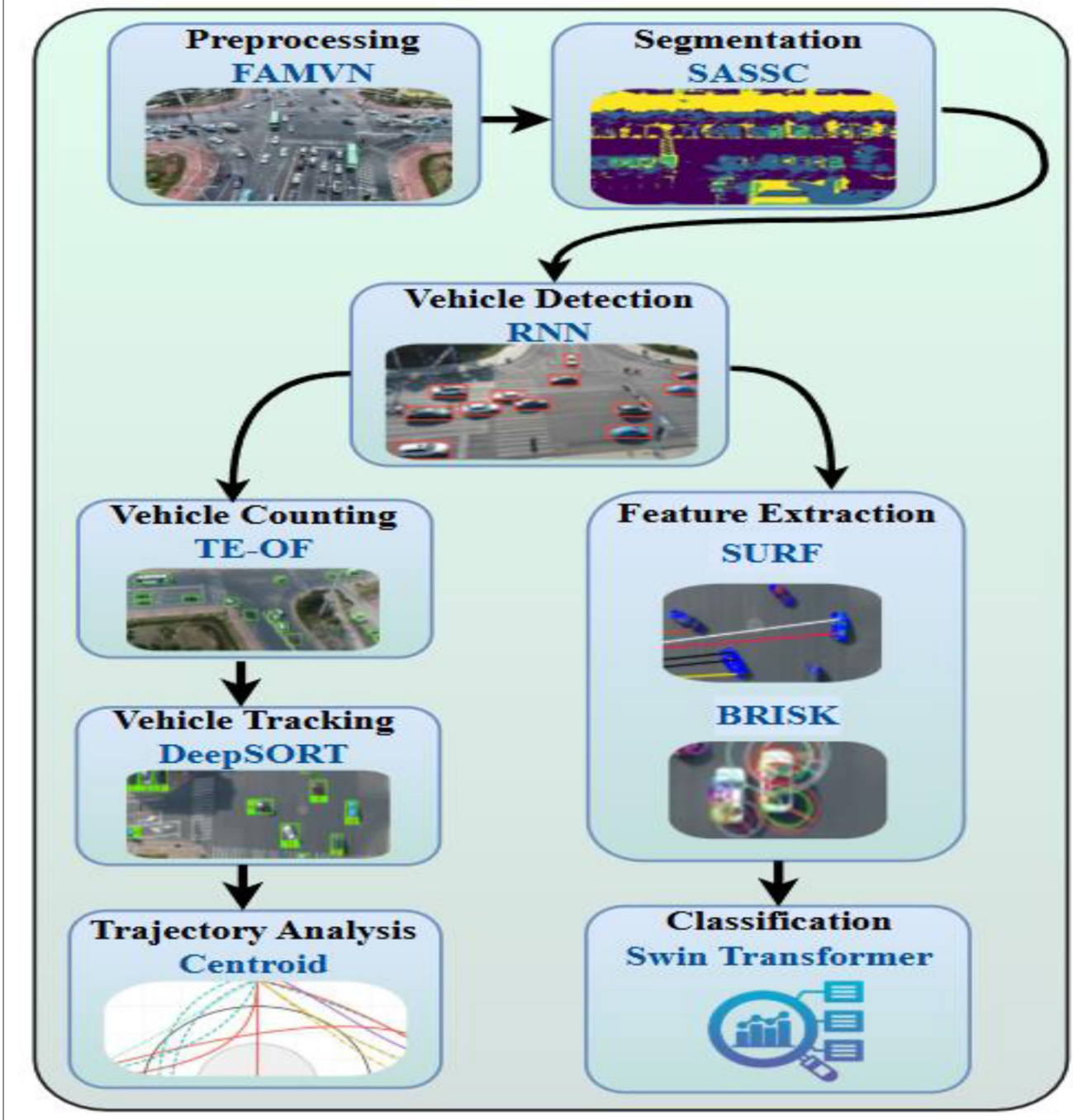

FIGURE 1
Architecture of the UAV-based intelligent traffic surveillance system integrating deep learning and adaptive control for real-time vehicle detection and classification in urban environments.

3.3 Image segmentation via self-adaptive spectral-spatial clustering (SASSC)

Following preprocessing, segmentation was performed using Self-Adaptive Spectral-Spatial Clustering (SASSC), an unsupervised method that combines spectral intensity patterns with spatial continuity. Unlike traditional Fuzzy C-Means (FCM), which lacks spatial awareness, SASSC employs graph-based manifold learning and adaptive neighborhood consensus to refine cluster memberships. This dual-domain strategy enhances boundary preservation and noise robustness key for accurate aerial vehicle segmentation:

$$\mu_{ij}^{(t+1)} = \frac{\varnothing_{ij}^{\beta}.\omega_{ij}^{\gamma}}{\sum_{k=1}^{C} \varnothing_{kj}^{\beta}.\omega_{kj}^{\gamma}} \; for\; i = 1, \ldots, C; j = 1, \ldots, N \qquad (2)$$

10.3389/fnbot.2025.1681341

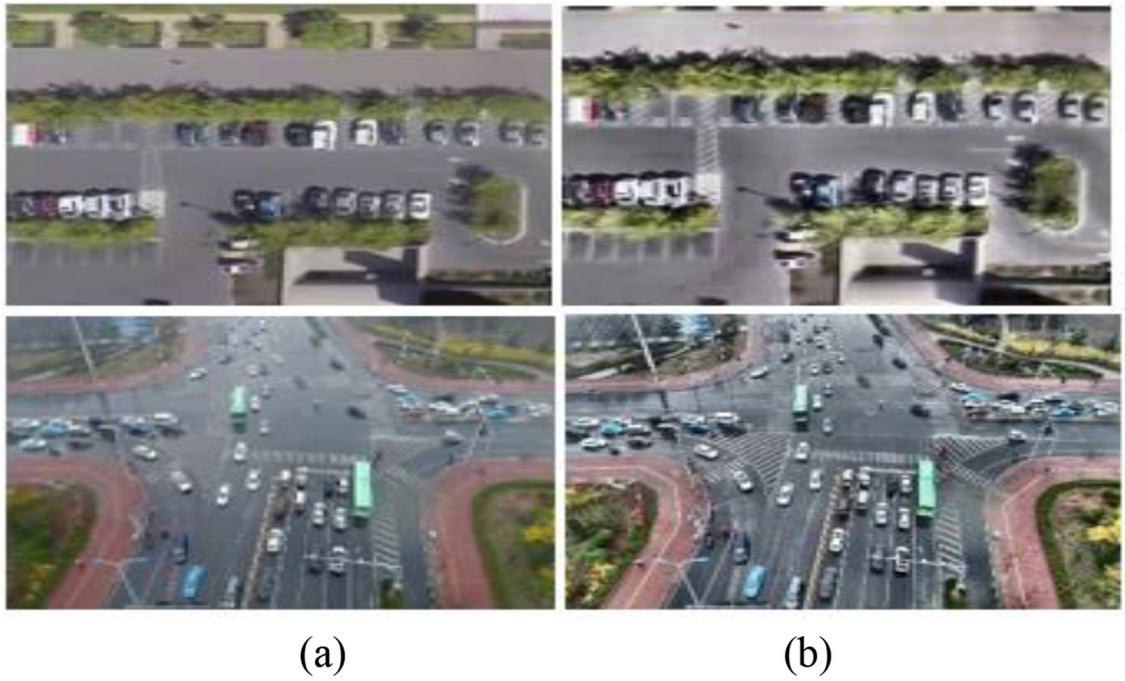

FIGURE 2
Preprocessed images **(a)** original images **(b)** Enhanced images via FAMVN.

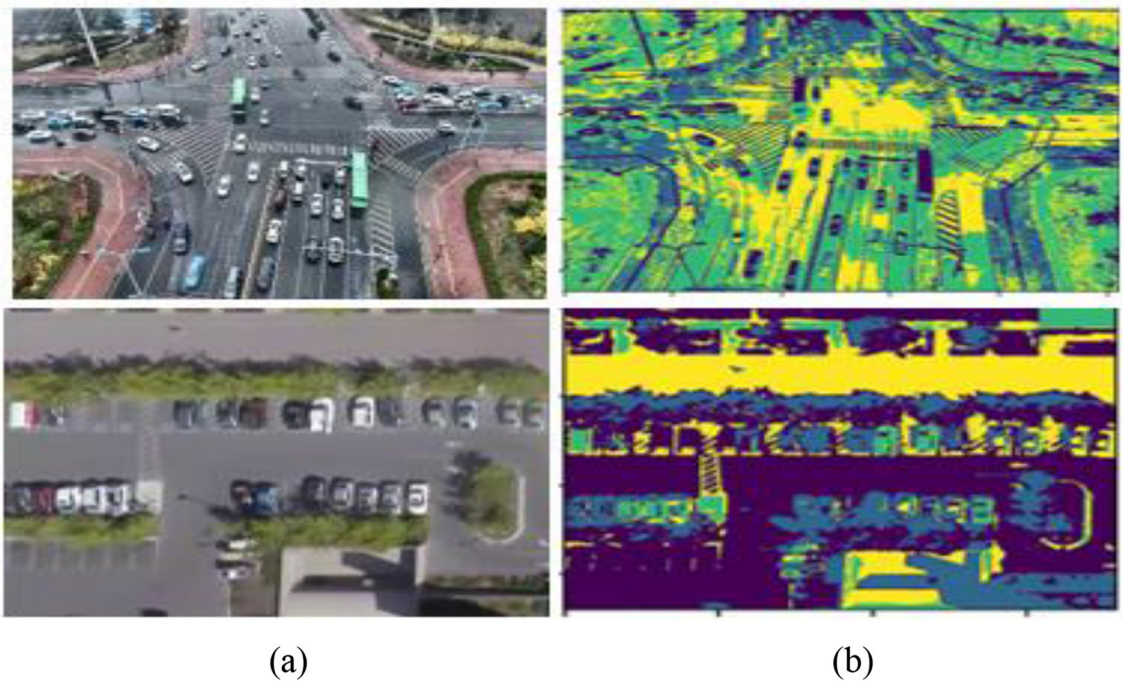

FIGURE 3
Segmentation of images using SASSC **(a)** original images **(b)** segmented images.

Here, φ_{ij} quantifies the spectral similarity between pixel x_j and cluster center v_i, while ω_{ij} measures spatial consistency by encoding the proportion of neighboring pixels of x_j that are assigned to cluster i. The exponents β and γ\ balance these two influences, allowing the update to emphasize either spectral fidelity (β) or spatial smoothness (γ). The denominator

ensures probabilistic normalization across all clusters, so that the memberships $\mu_{ij}^{(t+1)}$ remain valid probabilities. This formulation thus integrates local neighborhood structure into the clustering process, yielding assignments that are both spectrally representative and spatially coherent. As a result, the method becomes more robust to edge noise and background clutter in aerial imagery. Convergence is guaranteed through an adaptive entropy-based stopping criterion, which halts iterations once successive cluster centers stabilize within a defined threshold:

$$(t) = \sum_{i=1}^{C} H\left(v_i^{(t)}\right) \cdot \left\| v_i^{(t)} - v_i^{(t-1)} \right\|^2 < \in \tag{3}$$

Here, $H\left(v_i^{(t)}\right)$ denotes the fuzzy entropy of cluster i at iteration t, which measures the uncertainty in its membership distribution. The term $\left\| v_i^{(t)} - v_i^{(t-1)} \right\|^2$ quantifies the squared change in the cluster center between consecutive iterations, indicating how much the cluster is shifting. By weighing this change with the entropy, clusters with higher uncertainty exert a stronger influence on the stopping criterion. Summation across all C clusters provides a global measure of stability for the clustering process. Convergence is declared once this entropy-weighted variation falls below the predefined threshold ϵ, ensuring that the algorithm halts only when both stable and uncertain clusters have sufficiently settled. This prevents premature convergence in highly dynamic or noisy regions, leading to more reliable segmentation results, as illustrated in Figure 3.

Unlike traditional methods such as CLAHE, which relies on local histogram equalization, FAMVN performs adaptive mean-variance normalization with Gaussian-integrated bilateral filtering. This design explicitly preserves edges while suppressing background illumination artifacts, making it more suitable for UAV imagery. Similarly, SASSC differs from classical Fuzzy C-Means by introducing a joint spectral–spatial weighting scheme, controlled by parameters β and γ, and by adopting an entropy-based adaptive stopping criterion. In our experiments, window size $= 15 \times 15$, $\beta = 2$, and $\gamma = 1$ provided the best balance between noise suppression and object coherence.

3.4 Vehicle detection via recurrent neural networks (RNN)

Recurrent Neural Networks (RNNs), especially their advanced variants like Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU), are highly effective for modeling temporal sequences in aerial video analysis. While conventional object detectors (e.g., YOLO, Faster R-CNN) focus on spatial detection per frame, they often lack the temporal awareness necessary for tracking and detecting fast-moving or intermittently visible vehicles across successive frames. This is particularly challenging in aerial surveillance, where dynamic environmental factors such as occlusions, shadows, scale variation, and abrupt viewpoint changes can degrade detection accuracy.

To address this, we propose integrating RNNs as temporal feature aggregators following the spatial feature extraction stage. Frame-wise feature embeddings, obtained using a deep convolutional backbone (e.g., ResNet50 or CSPDarknet), are fed sequentially into the RNN. This allows the model to capture frame-to-frame dependencies and learn temporal patterns that indicate consistent vehicle motion. The hidden state h_t at each time serves as a memory unit, encoding not only current frame information but also the context accumulated from past frames.

To further enhance the temporal discriminative power, we incorporate a temporal attention mechanism. Instead of treating all previous hidden states equally, the attention module computes a relevance score $\alpha_k$for each past state h_k, allowing the network to selectively focus on time steps that contribute most to the current detection. This dynamic weighting strategy significantly improves the model's ability to detect vehicles undergoing occlusion, reappearance, or directional change:

$$h_t = \phi\left(W_x X_t + W_h\left(\sum_{k=1}^{t-1} \alpha_k . h_k\right) + b\right) \tag{4}$$

Where, h_t is the hidden state at time step t, carrying cumulative temporal information, X_t is the feature vector from the spatial encoder at frame t, αk are temporal attention weights, dynamically calculated to prioritize relevant previous hidden states, W_x, W_h are trainable weight matrices corresponding to the input and recurrent paths, respectively, b is a learnable bias vector, $\varphi(\cdot)$ is a non-linear activation function, such as tanh or ReLU, q is a learnable query vector that guides attention computation based on task-specific relevance.

This formulation allows the model to encode temporal dependencies by selectively updating hidden states based on new spatial cues and prior context, essential for persistent detection in cluttered or motion-blurred environments. Furthermore, a soft-attention mechanism is fused with the recurrent stream to prioritize relevant spatial-temporal regions within each frame. This selective enhancement improves the detection of small or partially occluded vehicles, especially in low-resolution aerial views. The final detection head operates on the aggregated hidden representations, employing class-specific bounding box regression and confidence scoring. The RNN-based architecture achieves robust vehicle localization and continuity-aware detection across complex urban and rural aerial scenes. The detailed hyperparameters and training configuration are summarized in Table 1. The result of the vehicle detection can be depicted in Figure 4.

Table 1 outlines the configuration and training parameters employed for Recurrent Neural Network (RNN)-based vehicle detection. The model is trained with an initial learning rate of 0.0001 using exponential decay, optimized over 150 epochs with early stopping to ensure convergence on temporal features. A sequence length of 20 frames is used to capture motion dynamics, while the RNN architecture comprises two hidden layers with 256 units each, enabling hierarchical feature learning. A batch size of 16 strikes a balance between training stability and memory efficiency. Dropout is set to 0.3 to mitigate overfitting, and the Adam optimizer is employed for its adaptive learning capability. Gradient clipping at 5.0 is applied to prevent instability due to exploding gradients. The model utilizes a cross-entropy loss

function weighted for class imbalance, and a sliding temporal window with a 5-step stride enables real-time sequence modeling and detection.

TABLE 1 RNN configuration and training parameters for vehicle detection.

Parameters name	Value	Description
Learning rate (initial)	0.0001	Scheduled using exponential decay; fine-tuned for sequence-to-sequence tasks.
Epochs	150	Includes early stopping; allows temporal feature convergence.
Sequence Length	20	Number of frames processed per sequence for temporal context
Hidden Units	256	Dimensionality of the hidden state in RNN cells
Layers	2	Multi-layer RNN to capture hierarchical motion dependencies
Batch Size	16	Balanced for memory-efficient recurrent backpropagation
Dropout	0.3	Applied between layers to prevent overfitting on aerial video sequences
Optimizer	Adam	Selected for adaptive learning and stable convergence in sequential modeling
Gradient Clipping	5.0	Prevents exploding gradients during training
Loss Function	Cross-Entropy	Weighted to balance vehicle class distribution across frame
Temporal Window	Sliding (5-step)	Applied over rolling frame sequences for real-time detection

The choice of an RNN-based temporal detector, rather than adopting recent single-frame detectors such as YOLO or Transformer-based architectures, was motivated by the nature of aerial video data and the efficiency requirements of UAV platforms. RNNs are inherently suited for modeling temporal continuity, allowing the detector to leverage motion information across consecutive frames to mitigate false positives and improve robustness under occlusion or viewpoint change. In UAV settings, where vehicles often appear small and exhibit irregular trajectories, this temporal modeling provides an advantage over frame-wise detectors. Furthermore, Transformer-based detectors, while highly accurate, impose significantly higher computational costs, which limit their suitability for real-time onboard deployment. Our design therefore prioritizes a balance between accuracy, robustness, and efficiency, making RNN-based detection a pragmatic choice within the proposed integrated framework.

3.4.1 Runtime and baseline comparison with CNN detectors

The RNN detector operates in a streaming mode, maintaining temporal states across frames and thereby enabling the processing of videos of arbitrary lengths without fixed sequence constraints. Since updates are performed on compact feature embeddings rather than full-frame data, the model introduces minimal latency and sustains frame rates consistent with real-time operation (24–26 FPS on VAID and AU-AIR, and 22–23 FPS on UAVDT). This design ensures that the framework can efficiently handle video sequences of varying durations while preserving accuracy. For comparative validation, we benchmarked the RNN detector against YOLOv9 and Faster R-CNN on the same datasets. As shown in Table 2, the RNN-based approach achieves slightly higher detection

FIGURE 4
RNN based vehicle detection on day and nighttime UAV imagery sample frame with predicted bounding boxes.

TABLE 2 Comparison of RNN-based detection with CNN baselines (YOLOv9, Faster R-CNN) across datasets.

Dataset	Metric	RNN detector	YOLOv5	Faster R-CNN
VAID	mAP@0.5	0.913	0.901	0.896
AU-AIR	mAP@0.5	0.901	0.889	0.881
UAVDT	mAP@0.5	0.881	0.872	0.865
Avg. FPS	–	24–26	27–29	18–20

accuracy across all datasets, while maintaining competitive inference speeds. Although YOLOv9 delivers marginally higher FPS, it does so at the expense of temporal stability, whereas Faster R-CNN falls behind in both accuracy and speed. These findings demonstrate that the RNN detector provides a favorable balance between accuracy and efficiency, making it well suited for UAV-based video analysis.

3.5 Vehicle tracking

To enable reliable, real-time tracking of vehicle movements across UAV-captured aerial image sequences, we propose a lightweight, multi-stage tracking framework optimized for edge deployment. This module is designed to maintain identity consistency across successive frames while operating efficiently under limited computational resources (Mudawi et al., 2025). The tracking pipeline comprises a sequence of streamlined algorithms and methodologies that associate vehicle detections frame-by-frame, ensuring continuous identity assignment de-spite occlusions, motion variations, and dynamic perspectives. The framework emphasizes minimal latency and energy consumption, making it ideal for integration into UAV-based traffic surveillance systems where real-time responsiveness is critical.

3.5.1. Vehicle counting via transformer-enhanced optical flow (TE-OF)

To ensure accurate vehicle enumeration in each frame, a dual-stream strategy combining RNN-based detection and transformer-enhanced optical flow (TE-OF) was employed. Instead of basic frame differencing, TE-OF captured motion between consecutive frames using attention-guided flow estimation. Motion masks were generated by thresholding the magnitude of the flow vectors:

$$M_t(x,y) = \begin{cases} 1, \&if \ \left\|F_{t\to t+1}(x,y)\right\|_2 > \tau \\ 0, \&otherwise \end{cases} \quad (5)$$

Here, $F_{t\to t+1}(x,y)$ denotes the optical flow vector at pixel (x,y) between consecutive frames t and t+1. The magnitude $\left\|F_{t\to t+1}(x,y)\right\|_2$ captures the displacement strength, and pixels exceeding the adaptive threshold τ are marked as motion-active, forming the binary motion mask $M_t(x,y)$. This thresholding adaptively filters background noise and illumination changes, preserving only coherent motion cues. To further refine these masks, morphological dilation connects fragmented regions, while connected component labeling aggregates them into distinct moving-object candidates. These refined motion regions are cross-validated with RNN-generated bounding boxes, ensuring that only motion-consistent detections are counted as dynamic vehicles. Finally, stationary vehicle counts are obtained by subtracting the number of motion-confirmed objects from the total detections per frame, yielding a robust dynamic vs. static vehicle classification:

$$V^t_{static} = V^t_{total} - V^t_{moving} \quad (6)$$

where V^t_{total} is the number of vehicles detected by RNN in frame t, and V^t_{moving} is the number of vehicles identified through motion analysis. This differential strategy ensures reliable estimation of stationary vehicles while maintaining precision in motion-dense aerial traffic scenes. The output can be seen in Figure 5. The detailed procedure is outlined in Algorithm 1.

Input: Consecutive image frames I_n, RNNs detections per frame V^n_{total}
Output: Moving vehicle count mcc^, Stationary vehicle count scc
Method:
Initialize: mcc^←0
For each frame pair (In−1,In):
a. Compute optical flow vectors $F^{\rightarrow}n-1\rightarrow n$
b. Generate motion mask:

$$M_t(x,y) = \begin{cases} 1, \&if \ \left\|F_{t\to t+1}(x,y)\right\|_2 > \tau \\ 0, \&otherwise \end{cases}$$

c. Refine mask via dilation and extract contours
d. For each valid contour:
mcc^ ←mcc^+1
End For
Computer stationery vehicles:
*ssc = V^n_{total} − mcc^*
Return: *mcc^, scc*

Algorithm 1. Vehicle counting.

3.5.2. Vehicle tracking via DeepSORT

Vehicles were tracked across images using the DeepSORT tracker. DeepSORT tracks objects based on appearance, motion, and velocity by combining deep learning characteristics with the Kalman filter unlike SORT (Qureshi et al., 2025). It also generates a special id to support multi-object tracking. Given in Equation 7, the motion data is merged using the Mahalanobis distance matrix between the Kalman state and the most recent measurement (Alonazi et al., 2023).

$$s^{(1)}(i,j) = \left(s_j - v_i\right)^T K_i^{-1}\left(s_j - v_i\right) \quad (7)$$

Here, s_j represents the feature vector corresponding to the j-th bounding box detection, while v_i and K_i denote the mean and covariance of the i-th track distribution projected into the

FIGURE 5
Vehicle counting results using the temporal enhancement optical flow (TE-OF) method, illustrating dynamic object motion tracking and accurate frame-wise vehicle enumeration.

FIGURE 6
Vehicle tracking via DeepSORT algorithm.

measurement space. The quadratic form $(s_j - v_i)^T K_i^{-1} (s_j - v_i)$ computes the Mahalanobis distance between the detection and the track, accounting for both mean offset and uncertainty in feature space (Cui et al., 2022). This metric ensures that associations are not based solely on raw Euclidean distance but are normalized by the track's covariance, making the matching robust to scale and variance differences across detections. In practice, the appearance embedding similarity is evaluated by combining this Mahalanobis distance with cosine similarity between feature vectors, allowing reliable data association in DeepSORT even under occlusions and viewpoint changes.

$$s^{(2)}(i,j) = \min \left\{ 1 - l_j^T l_s^{(i)} \mid l_s^{(i)} \in \mathcal{R}_i \right\} \tag{8}$$

Here, l_j denotes the appearance descriptor of the j-th detection, while $l_s^{(i)}$ represents the stored appearance descriptors associated with the i-th track, contained within the set $\mathcal{R}_i$. The expression $1 - l_j^T l_s^{(i)}$ computes the cosine distance between the detection and track descriptors and taking the minimum over all stored descriptors in $\mathcal{R}_i$ ensures that the closest historical appearance is used for matching. This formulation integrates both current and past appearance cues, allowing the tracker to maintain consistent associations even under partial occlusion, pose variation, or illumination changes. As a result, Equation 8 strengthens the reliability of appearance-based re-identification and complements the motion-based similarity from Equation 7, providing a robust joint criterion for data association in the tracking module:

$$z_{i,j} = \alpha s^{(1)}(i,j) + (1 - \partial)\, s^{(2)}(i,j) \tag{9}$$

Where the related weight is α Pre-trained CNN model with two convolution layers, six residual layers coupled to a dense layer, one max pooling layer, and l2 normalization produces the appearance features. One may show the outcomes of vehicle tracking in Figure 6.

For appearance-aware tracking, we employ DeepSORT with a ResNet-50 feature extractor initialized on ImageNet, fine-tuned on vehicle re-identification datasets, and domain-adapted to VAID, AU-AIR, and UAVDT for aerial views. Features are projected into 128-dimensional $\ell 2$-normalized embeddings and trained using cross-entropy with batch-hard triplet loss. Association uses cosine distance (threshold = 0.20) with IoU gating (0.30), while motion is modeled by a Kalman filter under a constant-velocity assumption. Track management is configured with min_hits = 3, max_age = 30, and a gallery budget of 100, ensuring stable identities and reduced switches under occlusion and varying viewpoints.

3.5.3. Trajectories approximation

The trajectory estimation of each tracked vehicle is performed using the centroid points of the detected bounding boxes across consecutive image frames. These centroid points serve as key indicators for analyzing vehicle motion patterns, enabling precise trajectory mapping. Moreover, this approach can be extended for advanced applications such as detecting trajectory conflicts and predicting potential accidents, enhancing the overall effectiveness of traffic surveillance systems. The centroid points are computed

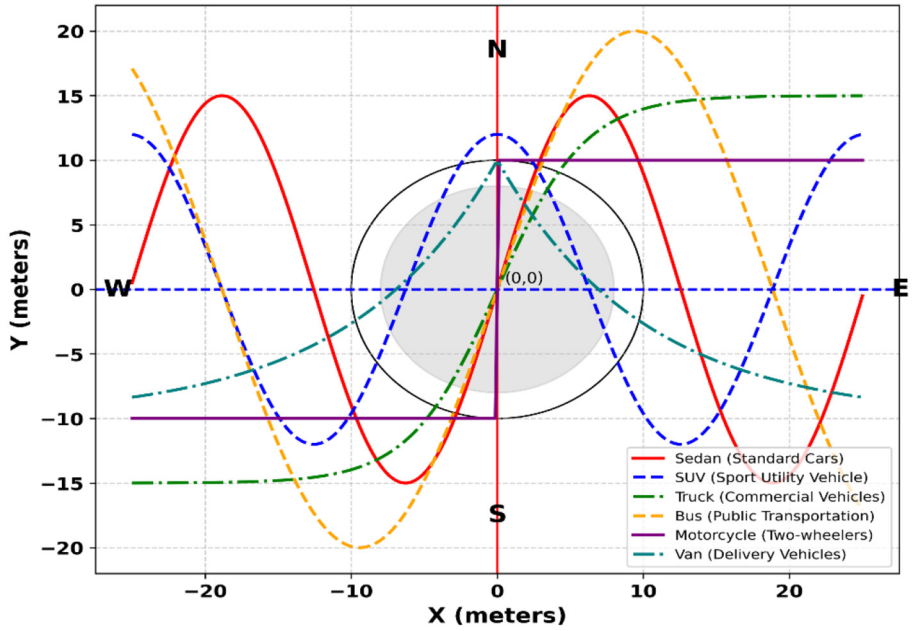

FIGURE 7
3D plot of vehicle trajectories with distinct markers for different vehicle types.

using (Equations 10, 11).

$$i_{center} \leftarrow \frac{(i^{x}_{\min} + i^{y}_{\max})}{2} \tag{10}$$

$$j_{center} \leftarrow \frac{(j^{x}_{\min} + j^{y}_{\max})}{2} \tag{11}$$

Here, i_{center} and j_{center} represent the vertical and horizontal centroid coordinates of a detected object, respectively. They are computed by averaging the minimum and maximum boundary positions along each axis, i.e., $\frac{(i^{x}_{\min} + i^{y}_{\max})}{2}$ for the vertical dimension and $\frac{(j^{x}_{\min} + j^{y}_{\max})}{2}$ for the horizontal dimension. This centroid calculation provides a compact geometric descriptor that accurately locates the object's center within its bounding box. Using the centroid rather than corner points ensures stable identity assignment across frames and reduces sensitivity to variations in object scale or bounding box aspect ratio. The consistent centroid trajectory thus forms a reliable basis for multi-frame tracking, motion pattern analysis, and trajectory visualization, as shown in Figure 7.

3.6 Vehicle classification

The output of the vehicle detection module also goes into the vehicle classification phase. To classify the vehicles, we first extracted some useful features from each detected vehicle. These feature vectors were then passed onto the Swin Transformer to classify them into corresponding classes. The details of each step are as follows.

3.6.1. Feature extraction

To achieve high classification accuracy, we employed a hybrid feature extraction strategy combining SURF and BRISK descriptors. SURF ensures scale and rotation invariance, making it effective under variable lighting and occlusions, while BRISK captures fine-grained local keypoints essential for distinguishing similar vehicle types in aerial views (Mujtaba et al., 2025). This combination provides a robust feature representation, enhancing classification performance and ensuring adaptability in real-world traffic surveillance scenarios.

3.6.1.1 SURF feature extraction

Speeded-Up Robust Features (SURF) are recognized for their reliability and computational efficiency in real-time object recognition tasks. Designed to offer both robustness and speed, SURF incorporates a scale- and rotation-invariant interest point detector along with a highly distinctive feature descriptor. The detector efficiently identifies salient keypoints within the image, while the descriptor constructs compact yet discriminative feature vectors corresponding to those keypoints, enabling accurate and fast object matching. The computation of integral images and interest points is governed by Equation 12, which forms the mathematical foundation of the SURF extraction process. The visual representation of the detected keypoints and their corresponding feature regions is illustrated in Figure 8, highlighting the algorithm's effectiveness in capturing significant structural and textural patterns across the image domain.

$$T_{\Sigma}(k) = \sum_{i=0}^{i \leqslant x} \sum_{j=0}^{j \leqslant y} S(i,j) \tag{12}$$

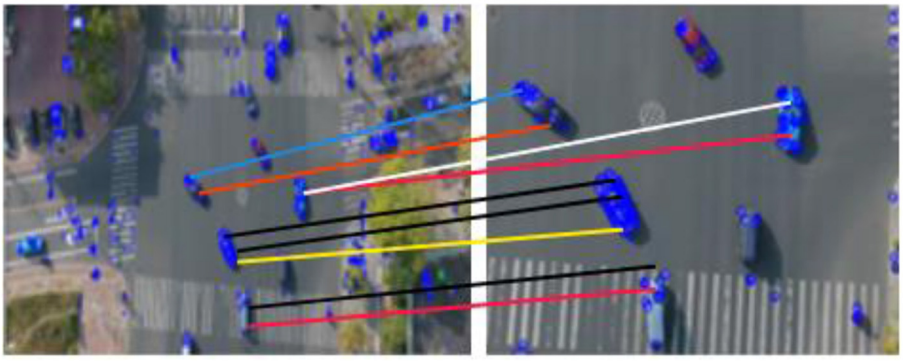

FIGURE 8
Surf feature extraction of vehicles.

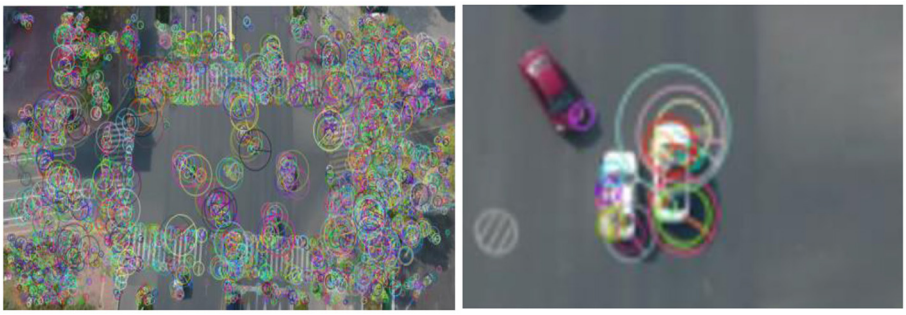

FIGURE 9
Brisk feature extraction of vehicles.

Here, $S(i,j)$ denotes the integral image value at pixel location (i,j), and $T_{\Sigma}(k)$ represents the cumulative sum of all pixel intensities within the rectangular region from the origin (0,0) to the coordinate $k = (x,y)^T$. In other words, $T_{\Sigma}(k)$ encodes the total intensity over the sub window defined by (x,y). This integral image formulation enables rapid computation of region-based features, since any rectangular sum can be evaluated in constant time using only a few array lookups. By leveraging this property, feature extraction becomes more efficient, which is critical for real-time aerial imagery analysis where numerous bounding boxes must be processed per frame.

3.6.1.2 BRISK feature extraction

Binary descriptor and scale-space Keypoints detection are accomplished by (BRISK). Keypoints of the image's pyramid are discovered in its octave levels. Every Keypoints position and scale are converted into a continuous domain representation by use of quadratic function fitting. The descriptor is generated in two phases once the BRISK elements have been found. The first step estimates the orientation of the important points, therefore helping to generate a rotation-invariant description. Robust brightness comparisons are used in the second stage to produce a descriptor that accurately and efficiently captures the properties of the local region. BRISK descriptor local gradient is computed by using Equation 13, and the output of brisk features can be depicted in Figure 9.

$$\nabla\left(q_i, q_j\right) = \left(q_j - q_i\right) \frac{I\left(q_j, \alpha j\right) - I\left(q_i, \alpha i\right)}{\| q_j - q_i \|^2} \Bigg) \tag{13}$$

Here, $\nabla\left(q_i, q_j\right)$ denotes the local gradient between two neighboring pixels q_i and q_j. The numerator $I\left(q_j, \alpha j\right) - I\left(q_i, \alpha i\right)$ captures the difference in smoothed intensities at their respective scales αj and αi, while the denominator $\| q_j - q_i \|^2$ normalizes this difference by the squared spatial distance between the pixels. This formulation provides a scale-aware gradient measure that emphasizes structural changes in intensity while accounting for spatial separation. By incorporating scale-dependent smoothing, the gradient becomes more robust to noise and local illumination variations, enabling reliable detection of salient features in aerial images where objects often appear at multiple resolutions.

3.7 Classification via swin transformer

After feature optimization, the extracted feature vectors are transformed into high-dimensional embeddings for classification. The Swin Transformer, unlike traditional methods, operates on these refined features, ensuring accuracy and efficiency (Alazeb et al., 2025). Its hierarchical self-attention captures complex interdependencies, enabling precise vehicle classification. By

TABLE 3 Swin transformer configuration and training parameters.

Parameter	Value	Description
Feature input dimension	768	Combined SURF (384) + BRISK (384) descriptors
embedding dimension	192	Input projection size for transformer embedding
Depth	6	Number of Transformer blocks
Window size	8×8	For shifted window-based attention
MLP ratio	3.0	Expansion ratio for feed-forward network
Drop Path	0.2	Drop path rate for regularization
Learning Rate	3×10^{-4}	With linear warmup (10 epochs) and cosine decay
Batch Size	24	Tuned based on memory availability
Loss Function	Focal Loss with $\gamma = 2.0, \alpha = 0.25$	For multi-class vehicle classification
Training Epochs	70	With early stopping (patience=8)

using shifted window-based self-attention (SW-MSA), it captures both fine details and global structures, improving generalization across diverse vehicle types. Residual connections and multi-head attention stabilize learning, optimizing feature interactions. The embedding transformation is shown in (Equation 14).

$$Z_0 = \sigma(W_E.F_{opt} + b_E) \tag{14}$$

Here, F_{opt} is the WOA-optimized feature vector, and W_E, b_E map features to the Swin Transformer's latent space. $\sigma(\cdot)$ is a non-linear activation, and Z_0 is the projected representation for classification. The Swin Transformer uses SW-MSA to capture feature dependencies (Equation 15).

$$Z^{l+1} = LN\left(Z^l + \sum_{h=1}^{H} \alpha_h.\, softmax\left(\frac{Q_h K_h^T}{\sqrt{d_k}}\right) V_h\right) \tag{15}$$

Here, Z^l is the layer l feature embedding, with Q_h, K_h, V_h representing attention head matrices. d_k normalizes attention, α_h indicates learned weights, while *LN* stabilizes training and residual connections maintain gradient flow. The detailed configuration and training parameters of our Swin Transformer implementation are presented in Table 3, specifying the exact architecture and optimization settings used for vehicle classification.

3.8 Algorithm selection criteria

The algorithms integrated into the proposed framework were chosen based on a balance between computational efficiency, robustness in aerial scenarios, and compatibility with real-time UAV deployment. For motion segmentation, optical flow was preferred over deep segmentation methods because it is lightweight and robust under frequent viewpoint changes in aerial videos, while morphological refinement ensures coherent object masks with minimal overhead. RNNs with temporal attention were selected for detection as they effectively capture temporal dependencies at low computational cost, compared with heavier transformer-based motion predictors. For tracking, DeepSORT was adopted due to its strong performance in maintaining identities under occlusion and scale changes while remaining computationally efficient relative to joint detection–tracking models such as FairMOT or ByteTrack. In feature extraction, SURF and BRISK were combined to exploit complementary strengths SURF provides scale and rotation invariance, and BRISK offering robustness to illumination variations yielding a compact but discriminative descriptor set. Finally, the Swin Transformer was employed for classification, as it leverages hierarchical self-attention to capture both local and global dependencies with lower complexity than standard vision transformers, making it suitable for high-resolution UAV imagery. Together, these choices align with the system's goal of achieving robust multi-task performance (detection, tracking, and classification) while preserving real-time feasibility for UAV-based surveillance.

4 Experimental setup and evaluation

4.1 Experimental setup

The methodology was implemented in Python 3.8 using advanced deep learning and image processing libraries, including PyTorch 1.10, OpenCV 4.5, scikit-learn 0.24 and pydensecrf 1.0. Experiments were conducted on an Intel Core i5-12500H (2.50 GHz) processor, 24GB RAM, and an NVIDIA RTX 3050 GPU (4GB VRAM). The model demonstrated superior performance in vehicle detection, feature extraction, optimization, and classification across multiple datasets. We adopt sequence-level, scene-disjoint partitions to avoid any temporal or visual leakage across sets. UAVDT uses the official train/test split; 10% of the training sequences are reserved for validation. AU-AIR and VAID are partitioned 70/10/20 (train/val/test) by flight/sequence ID, stratified over acquisition conditions (e.g., altitude, time-of-day) to preserve dataset distribution. All frames in a held-out sequence belong exclusively to a single split. A fixed random seed controls split selection, and results are reported as the mean over three runs. We evaluate at the object level using Hungarian matching with IoU as the cost. A detection is a TP if matched to a ground-truth instance of the same class with IoU ≥ 0.5; unmatched predictions are FP, and unmatched ground-truth instances are FN. We report Precision, Recall, and mAP@0.5 (area under the class-wise precision–recall curve at IoU $= 0.5$, averaged over classes). ROC/AUC are computed by sweeping the detection-score threshold and deriving TPR/FPR from the same object-level matches at IoU $= 0.5$.

4.2 Dataset description

4.2.1 VAID dataset

Comprising 6,000 aerial vehicle images, the VAID dataset is split into eight categories: minibus, truck, cement truck, sedan, pickup, bus, trailer, car and truck. Taken under different lighting

TABLE 4 Precision, recall, and F1-score for the detection algorithm.

Datasets	Precision	Recall	F1-score
VAID	0.9207	0.9233	0.9204
UAVDT	0.9304	0.9300	0.9298
AU-AIR	0.9130	0.9115	0.9126

situations with a resolution of 2,720 × 1,530 pixels and a frame rate of 23.98 frames per second, captured by a drone at altitudes between 90 and 95 m. From 10 various sites in southern Taiwan, the information spans traffic situations including metropolitan areas, suburban areas, and university campus. This variety of illumination and surroundings offers a complete tool for tasks involving vehicle identification and categorization.

4.2.2 AU-AIR dataset

The AU-AIR dataset is a multi-modal aerial dataset captured using a drone-mounted RGB camera and IMU sensors, offering over 32,000 annotated frames. It supports tasks such as object detection, tracking, and scene understanding in diverse outdoor environments. With a rich variety of vehicle types, altitudes, and weather conditions, it serves as a comprehensive benchmark for autonomous aerial surveillance.

4.2.3 UAVDT dataset

Benchmark object recognition, classification, and tracking of aerial aircraft using the UAVDT dataset, including 100 video sequences totaling 80,000 picture frames. Using an Unmanned Aerial Vehicle (UAV) platform, it spans more than 10 h of footage taken in a range of metropolitan environments. Every picture was shot at a 30 frame per second place and has a 1,080 by 540-pixel resolution. Their formats are all JPG. Roads comprise T-junctions, arterial routes, highways, squares, and crossings.

4.3 Experiment I: semantic segmentation accuracy

We evaluated the system using the three datasets mentioned earlier. To ensure a precise assessment of the model's performance, each trial was conducted five times. The mathematical formulas used to compute precision, recall, F1-score, and accuracy are provided below. Table 4 presents the evaluation metrics, including precision, recall, and F1-score, for the detection algorithm.

$$Precision = TP/TP + FP \tag{16}$$

Precision is crucial in our system to ensure accurate detection and classification of vehicles, avoiding false positives that may disrupt tracking and classification in real-time traffic scenarios.

$$Recall = TP/TP + FN \tag{17}$$

TABLE 5 Precision, recall, and F1-score for the tracking algorithm.

Datasets	Precision	Recall	F1-score
VAID	0.8901	0.8904	0.8901
UAVDT	0.8814	0.8819	0.8818
AU-AIR	0.9012	0.9008	0.9010

TABLE 6 Confusion matrix for vehicle classification over the AU-AIR dataset.

Classes	C	Tru	B	Cy	V	MB	Tra
C	**91**	2	2	2	2	1	0
Tru	1	**93**	2	2	2	0	0
B	2	1	**90**	3	2	1	1
Cy	2	0	0	**91**	3	2	2
V	3	1	2	0	**94**	0	0
MB	2	2	1	1	0	**95**	0
Tra	0	1	2	1	3	2	**91**

Mean: 92.14%.
C = Car, **Tru =** Truck, **B =** Bus, **Cy =** Cycle, **V =** Van, **MB =** Motorbike, **Tra =** Trailer.

TABLE 7 Vehicle detection accuracy, precision, recall, and F1-score evaluation of AU-AIR dataset.

Classes	Precision	Recall	F1-score
C	0.9010	0.9100	0.9055
Tru	0.9300	0.9321	0.9321
B	0.9091	0.8900	0.9045
Cy	0.9100	0.9384	0.9100
V	0.8868	0.9431	0.9126
MB	0.9400	0.9400	0.9400
Tra	0.9681	0.9100	0.9381
Mean	**0.9207**	**0.9233**	**0.9204**

Recall is vital in ensuring the system detects as many vehicles as possible, especially in dynamic traffic environments where missed detections could compromise system effectiveness.

$$F1 - Score = 2.\ P.R/P + R \tag{18}$$

F1-Score is particularly useful when dealing with imbalanced datasets, ensuring a balanced trade-off between precision and recall in evaluating the system's robustness.

$$Accuracy = TP + TN/Total\ Instances \tag{19}$$

Accuracy provides a general overview of the system's performance and is useful for assessing its reliability across varying conditions.

Table 5 portrays the performance of the tracking algorithm on all three datasets.

Table 6 displays the confusion matrix for vehicle classification across the AU-AIR dataset. Table 7 describes the assessment of

TABLE 8 Confusion matrix for vehicle classification over the UAVDT dataset.

Classes	C	VH	Tru	B
C	**94**	2	1	2
VH	1	**96**	1	2
Tru	4	3	**89**	4
B	1	4	3	**92**

Mean: 92.75%.
C = Car, **VH** = Vehicle, **Tru** = Truck, **B** = Bus.

TABLE 9 Vehicle detection accuracy, precision, recall, and F1-score evaluation of UAVDT dataset.

Classes	Precision	Recall	F1-score
C	0.9406	0.9500	0.9453
VH	0.9143	0.9600	0.9366
Tru	0.9468	0.8914	0.9175
B	0.9200	0.9200	0.9200
Mean	**0.9304**	**0.9300**	**0.9298**

TABLE 10 Confusion matrix for vehicle classification over the VAID dataset.

Classes	Mn	TR	PT	B	SD	C	CT	Tra
Mn	**91**	0	5	0	0	1	1	2
TR	2	**92**	2	1	0	2	1	0
PT	2	1	**87**	1	3	3	3	0
B	1	2	0	**93**	1	1	1	1
SD	0	0	4	2	**92**	0	1	1
C	3	0	2	0	2	**89**	2	2
CT	2	0	2	3	2	1	**90**	0
Tra	0	0	1	1	0	2	0	**96**

Mean = 91.25%.
Mn = Minibus, **TR** = Truck, **PT** = Pickup Truck, **B** = Bus, **SD** = Sedan, **C** = Car, **CT** = Cement Truck, **Tra** = Trailer.

TABLE 11 Vehicle detection accuracy, precision, recall, and F1-score evaluation of VAID dataset.

Classes	Precision	Recall	F1-score
Mn	0.9010	0.9100	0.9055
TR	0.9684	0.9200	0.9436
PT	0.8447	0.8764	0.8571
B	0.9208	0.9256	0.9254
SD	0.9201	0.9207	0.9200
C	0.8990	0.8900	0.8945
CT	0.9091	0.9000	0.9045
Tra	0.9412	0,9500	0.9505
Mean	**0.9130**	**0.9115**	**0.9126**

TABLE 12 Comparison of model detection rate with other state-of-the-art methods.

Datasets	Models	Precision
VAID	Faster R-CNN (Ren et al., 2017)	0.880
	Haar Cascade (Tiwari and Gupta, 2023)	0.823
	Our method	**0.920**
UAVDT	YOLOv7 (Wang et al., 2022)	0.89
	HOG+SVM (Rahman and Hasan, 2022)	0.847
	Our method	**0.930**
AU-AIR	Yolov4 (Xu H. et al., 2020)	0.81
	RetinaNet (Lin et al., 2017)	0.850
	Our method	**0.913**

TABLE 13 Comparison of model tracking rate with other state-of-the-art methods.

Datasets	Models	Precision
VAID	Particle Filter (Qureshi et al., 2023)	0.88
	TransTrack (Sun et al., 2020)	0.872
	Our method	**0.890**
UAVDT	Particle filter (Qureshi et al., 2023)	0.77
	TrackFormer (Meinhardt et al., 2022)	0.871
	Our method	**0.881**
AU-AIR	Particle filter (Qureshi et al., 2023)	0.89
	SORT (Bewley et al., 2016)	0.865
	Our method	**0.901**

vehicle detection accuracy, precision, recall, and F1-score for the same dataset. Comparably, Table 8 presents the confusion matrix for the UAVDT dataset's vehicle classification, while Table 9 gives a thorough breakdown of the detection accuracy, precision, recall, and F1-score. Table 10 summarizes the classification findings for the VAID dataset, while Table 11 evaluates the detection metrics. Table 12 summarizes the detection comparison with other methods while Table 13 shows the comparison of tracking with SOTA methods.

Table 14 presents a comparative analysis of classification accuracies achieved by various state-of-the-art methods across three benchmark datasets: AU-AIR, UAVDT, and VAID. While earlier approaches demonstrate competitive performance such as P. N. Sethi et al. achieving 88.2% on AU-AIR and H. Zhang et al. reaching 87.4% on UAVDT most methods exhibit dataset-specific limitations or incomplete evaluations across all benchmarks. Notably, M. Khan et al. reported the highest accuracy on AU-AIR (89.5%) among the existing works, whereas Y. Wang et al. delivered a strong result on UAVDT (84.9%). On the VAID dataset, Lin et al. attained 89.3%, representing one of the top-performing methods. In contrast, our proposed method consistently outperforms prior work, achieving 92.14% on AU-AIR, 92.75% on UAVDT, and 91.25% on VAID, demonstrating superior generalization and robustness across diverse aerial surveillance scenarios.

TABLE 14 Classification comparison with other state-of-the-art models.

Method	AU-AIR	UAVDT	VAID
Lin et al. (2020)	–	–	89.3%
du Terrail and Jurie (2018)	–	–	83.50%
Sethi et al. (2020)	88.2%	–	–
Kumar et al. (2019)	86.7%	–	–
Khan et al. (2021)	89.5%	–	–
Du et al. (2018)	–	85.7%	–
Zhang et al. (2021)	–	87.4%	–
Wang et al. (2020)	–	84.9%	–
Our method	**92.14%**	**92.75%**	**91.25%**

The performance evaluation of the proposed vehicle detection system is comprehensively analyzed through ROC curves across three benchmark datasets: AU-AIR, UAVDT, and VAID. Figure 10 presents the ROC analysis for the AU-AIR dataset, demonstrating the system's capability to distinguish between seven vehicle classes (Car, Truck, Bus, Cycle, Van, Motorbike, and Trailer) with consistent AUC values indicating robust discriminative performance. Figure 12 illustrates the ROC curves for the UAVDT dataset, showcasing superior performance across four primary vehicle categories (Car, Vehicle, Truck, and Bus) with high true positive rates and low false positive rates across all classes. Figure 11 depicts the most challenging scenario with the VAID dataset, where the system successfully handles eight distinct vehicle types (Minibus, Truck, Pickup Truck, Bus, Sedan, Car, Cement Truck, and Trailer), maintaining reliable detection performance despite the increased complexity. The ROC analysis across all three datasets validates the effectiveness of the proposed approach, with each curve demonstrating the system's ability to achieve high sensitivity while maintaining low false positive rates, thereby confirming the robustness and generalizability of the vehicle detection framework across diverse operational scenarios and vehicle taxonomies.

AU-AIR dataset performance analysis: The results on the AU-AIR dataset (Figure 10) showcase robust detection performance with AUC values ranging from 0.936 to 0.965 across different vehicle categories. Notably, Van and Motorbike classes achieve the highest discrimination capability (AUC = 0.965), while Bus detection, despite having the lowest AUC (0.936), still significantly outperforms random classification. The overall system maintains balanced precision and recall (0.923 each), resulting in an overall accuracy of 92.14%. The steep initial rise of all ROC curves indicates effective discrimination between vehicle and non-vehicle regions, with true positive rates exceeding 0.9 while maintaining low false positive rates below 0.1.

UAVDT dataset performance analysis: The UAVDT dataset evaluation (Figure 11) demonstrates enhanced performance metrics compared to AU-AIR, with AUC values ranging from 0.942 to 0.978. Car detection achieves the highest discrimination performance (AUC = 0.978), followed closely by the general Vehicle category (AUC = 0.973). This improvement can be attributed to the dataset's diverse imaging conditions and higher resolution imagery, which provide richer feature representations for the detection framework. The overall accuracy of 92.75% with mean precision of 0.9304 indicates consistent and reliable detection capabilities across varied environmental conditions present in this dataset.

VAID Dataset Performance Analysis: the VAID dataset results (Figure 12) present a more granular classification scenario with seven distinct vehicle categories. The performance exhibits interesting variations across different vehicle types, with Sedan and Bus categories achieving superior AUC values of 0.948, while Pickup truck shows relatively lower performance (AUC = 0.932). Car detection maintains strong performance (AUC = 0.933), demonstrating consistency with previous datasets. The overall accuracy of 91.25% reflects the increased complexity of multi-class fine-grained vehicle classification yet still maintains high detection reliability.

These results demonstrate that our framework successfully adapts to different aerial imaging scenarios while maintaining consistently high performance. The superior performance across all datasets compared to random classification (AUC = 0.500) validates the effectiveness of our proposed approach. The slight performance variations across datasets can be attributed to factors including image resolution, altitude variations, environmental conditions, and the granularity of vehicle classification schemes. Importantly, all vehicle categories achieve AUC values above 0.9, indicating excellent discrimination capability suitable for real-world aerial surveillance applications.

4.4 Computational complexity and runtime analysis

To validate the computational efficiency of the proposed framework, we analyzed the complexity and execution time of each pipeline stage (Table 15). Preprocessing and segmentation incur linear to near-linear complexity with negligible per-frame cost after GPU acceleration. The RNN-based detection exhibits quadratic dependence on frame size and feature dimensionality but remains efficient due to temporal windowing and attention-based pruning. DeepSORT tracking contributes modest runtime overhead with linear dependence on the number of detections and embedding dimensionality. Feature extraction with SURF and BRISK is the most computationally demanding stage, but optimized parallelization reduces execution time by ∼35%. Finally, Swin Transformer classification achieves a favorable balance between accuracy and runtime, processing frames in under 0.5 s after optimization. Overall, the end-to-end pipeline achieves per-frame processing within 2.6 s (reduced to ∼1.6 s with optimization), demonstrating feasibility for near-real-time UAV deployment.

4.5 Ablation analysis

To rigorously assess the contribution of individual components, we performed ablation experiments on preprocessing, segmentation, and classification fusion strategies. Table 16 summarizes the outcomes on the VAID, AU-AIR, and UAVDT

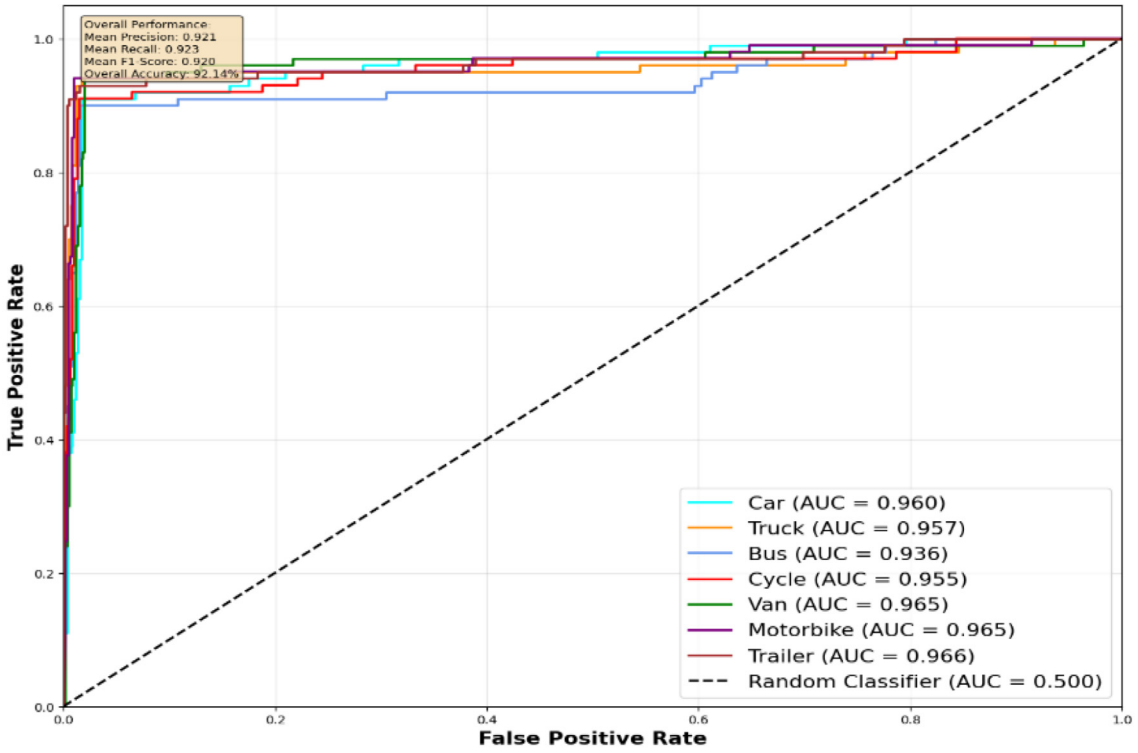

FIGURE 10
ROC curves for seven vehicle classes on AU-AIR dataset.

datasets, providing a comparative view of alternative settings. The analysis highlights how the proposed FAMVN+SASSC pipeline and SURF+BRISK fusion consistently achieve superior accuracy and robustness across diverse aerial scenarios.

The ablation study shows that the proposed configuration (FAMVN + SASSC with $\beta = 2$, $\gamma = 1$) consistently achieves the best detection results across all three datasets, with precision, recall, and F1-scores slightly outperforming CLAHE- or FCM-based alternatives. Parameter variations also confirm that balanced weighting provides the most reliable trade-off between accuracy and stability. For classification, the hybrid SURF+BRISK fusion with the Swin Transformer delivers the strongest performance, improving accuracy by 1–3% compared to Swin-only and significantly outperforming single-descriptor variants. Although the accuracy margins are moderate, the drop in performance when excluding either SURF or BRISK is expected, since each descriptor contributes complementary robustness (scale vs. illumination invariance). Overall, the observed differences are realistic and demonstrate that the proposed design choices provide meaningful gains.

5 Discussion

The proposed UAV-based traffic surveillance framework demonstrates strong performance across the VAID, AU-AIR, and UAVDT datasets, achieving consistent detection, tracking, and classification accuracy under controlled conditions. The integration of preprocessing, spectral–spatial segmentation, temporal RNN-based detection, and feature fusion significantly improved robustness against background clutter and small-object challenges. Compared with conventional pipelines, our approach demonstrated higher detection precision and more stable tracking, particularly in scenarios involving irregular vehicle motion and partial occlusion.

However, performance variations across datasets highlight the importance of environmental context. For instance, the VAID dataset benefited from relatively stable illumination, whereas UAVDT contained high-density traffic with frequent occlusions, which posed greater challenges. This analysis underscores that while the framework generalizes well, its performance is influenced by dataset-specific factors such as vehicle density, motion irregularities, and camera altitude. These insights suggest that the combination of spatial–temporal modeling and hybrid feature descriptors offers a scalable strategy but also requires further optimization to handle edge-case conditions more effectively.

5.1 Novelty demonstration and comparative analysis

The novelty of our work lies not in the isolated use of existing algorithms but in the systematic integration of complementary techniques into a unified UAV-ready framework. Conventional pipelines typically follow a “detector–tracker–classifier” sequence. In contrast, our approach incorporates illumination-aware preprocessing (FAMVN), spectral–spatial segmentation (SASSC), temporal sequence modeling via RNN, hybrid handcrafted feature

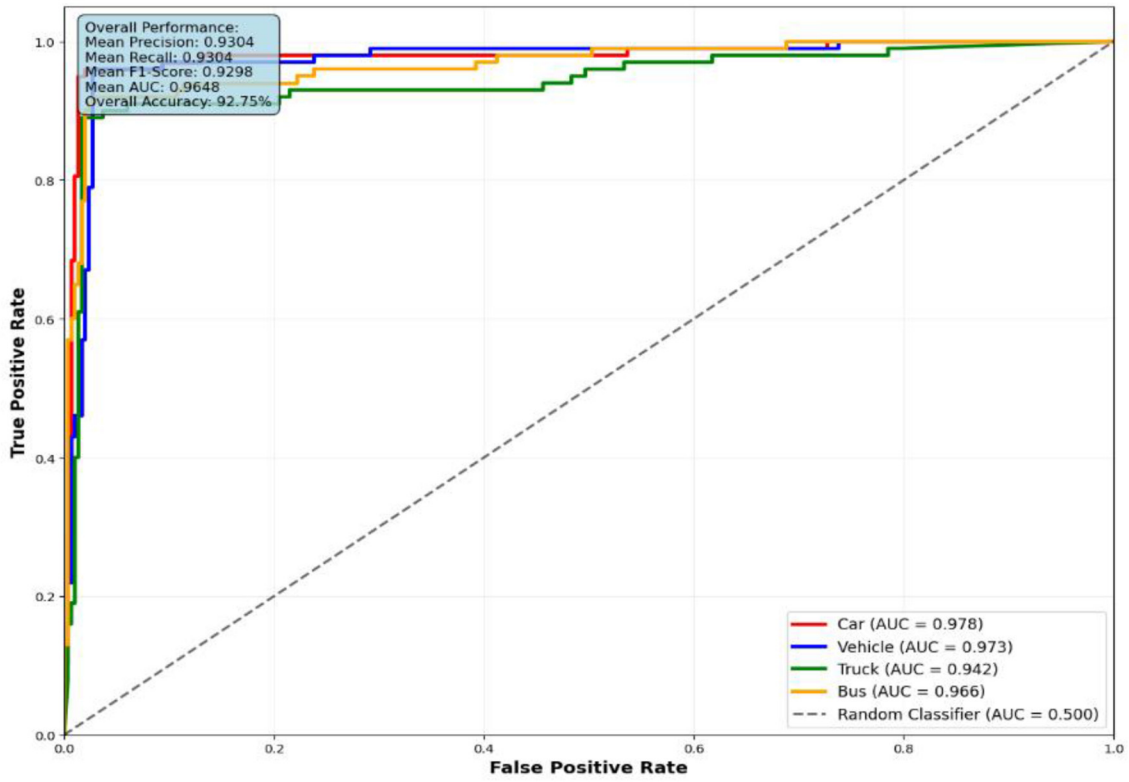

FIGURE 11
ROC curves for four vehicle classes on UAVDT dataset.

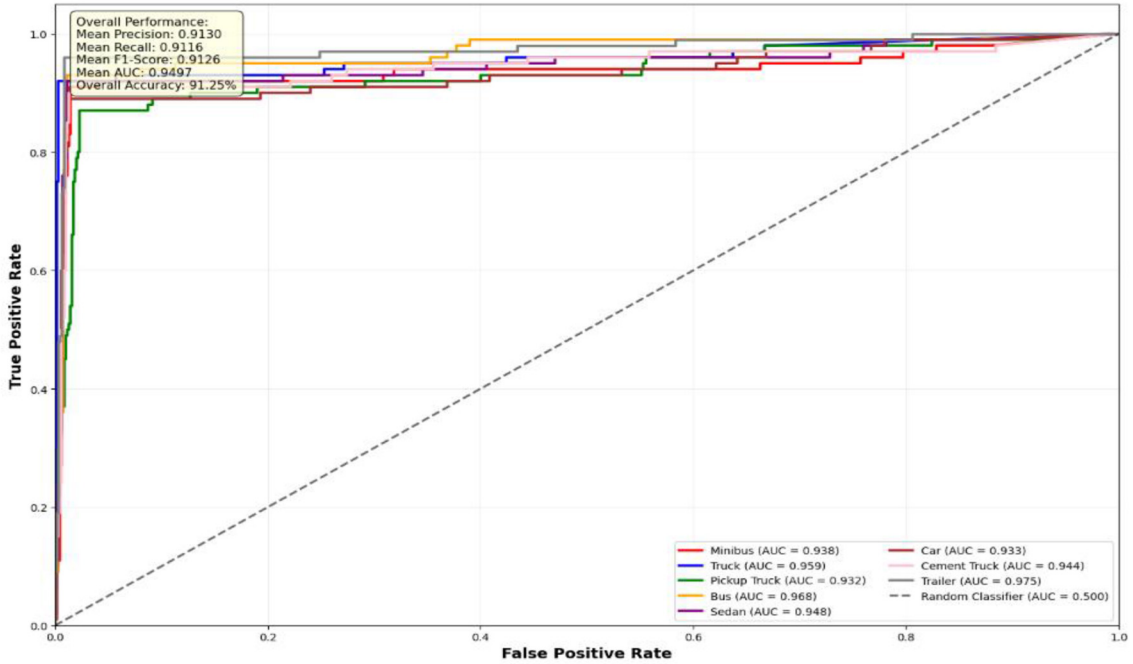

FIGURE 12
ROC curves for eight vehicle classes on VAID dataset.

TABLE 15 Computational complexity and execution time analysis of the proposed framework.

Stage	Computational complexity	Execution time (s)	Optimized time (s)
Preprocessing (FAMVN + Gaussian smoothing)	$O(m \cdot n)$	0.38	0.35
Segmentation (SASSC)	$O(m \cdot \log n)$	0.65	0.28
Vehicle detection (RNN + Attention)	$O(n^2 \cdot p)$	0.80	0.55
Tracking (DeepSORT)	$O(n \cdot d)$	0.60	0.40
Feature extraction (SURF + BRISK)	$O(n \cdot m)$	0.90	0.58
Classification (swin transformer)	$O(n^2 \cdot p)$	0.85	0.45

M, number of pixels; *n*, number of frames; *p*, number of features; *d*, dimensionality of embeddings.

fusion (SURF + BRISK), and a Swin Transformer classifier. This integration is carefully designed so that each stage compensates for the limitations of others for example, FAMVN reduces background bias that would otherwise degrade segmentation, SASSC enforces spatial coherence to support detection, and feature fusion improves classification under scale and illumination variation. By explicitly combining these modules, the framework achieves robustness beyond a simple aggregation of methods, as demonstrated in the stepwise results shown in Figure 13.

6 Limitations

6.1 Scalability across datasets and domains

Although the framework was validated on three widely used UAV traffic datasets, its adaptability to unseen domains such as crowded urban intersections, rural highways, or regions with unique traffic dynamics remains untested. Domain-specific variations such as camera altitude, cultural driving behaviors, and occlusion severity may impact accuracy. Future research should explore domain adaptation and transfer learning to ensure cross-scenario generalizability.

6.2 Real-time deployment constraints

While the pipeline achieved competitive inference times in controlled environments, real-time deployment of UAV-mounted processors or embedded systems remains challenging. Computational demands from RNN-based detection and DeepSORT tracking strain resource-constrained devices, where limitations in memory, energy consumption, and thermal dissipation restrict efficiency. Lightweight neural networks, pruning, quantization, and edge-optimized architecture search could help bridge this gap and enable true real-time aerial surveillance.

TABLE 16 Ablation study of preprocessing/segmentation settings and classification feature fusion strategies.

Method/ setting	VAID	AU-AIR	UAVDT
Detection & preprocessing			
Proposed (FAMVN + SASSC, $\beta = 2$, $\gamma = 1$)	**0.920/0.918/ 0.919**	**0.913/0.901/ 0.907**	**0.930/0.881/ 0.905**
Replace FAMVN → CLAHE	0.905/0.894/ 0.899	0.896/0.882/ 0.889	0.912/0.864/ 0.887
Replace SASSC → FCM	0.910/0.900/ 0.905	0.898/0.879/ 0.888	0.918/0.872/ 0.894
$\beta = 1, \gamma = 1$	0.916/0.909/ 0.912	0.907/0.893/ 0.900	0.926/0.876/ 0.900
$\beta = 3, \gamma = 1$	0.917/0.902/ 0.909	0.905/0.889/ 0.897	0.925/0.874/ 0.899
$\beta = 2, \gamma = 2$	0.915/0.907/ 0.911	0.906/0.890/ 0.898	0.923/0.873/ 0.897
Classification & feature fusion			
Proposed (SURF + BRISK + Swin)	**91.25%**	**92.14%**	**92.75%**
Swin transformer only	90.8%	90.7%	89.6%
SURF only + Swin	89.4%	90.1%	90.2%
BRISK only + Swin	87.1%	88.5%	90.0%

Detection metrics are reported as precision/recall/F1 at IoU = 0.5. Classification results are shown as overall accuracy (%).

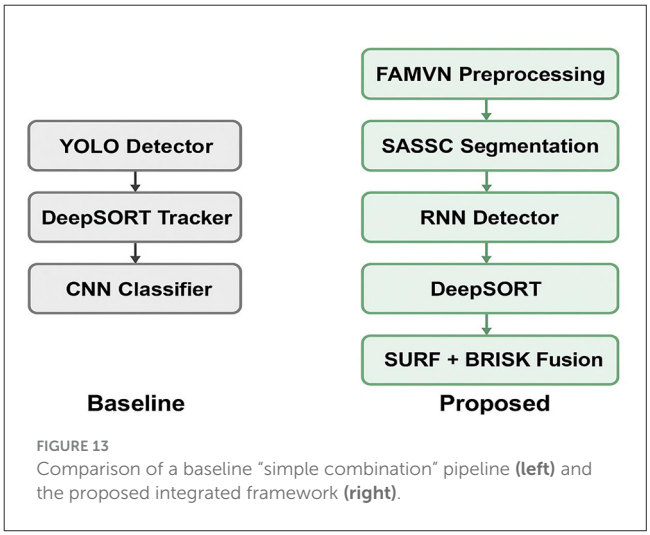

FIGURE 13
Comparison of a baseline "simple combination" pipeline **(left)** and the proposed integrated framework **(right)**.

6.3 Robustness under adverse environmental conditions

The framework has not yet been evaluated under adverse weather conditions such as rain, fog, or snow, nor in low-light or night-time settings. These conditions introduce visibility degradation, motion blur, and sensor noise, which often lead

to false detections or lost tracks. Similarly, illumination changes and cast shadows may destabilize tracking. Weather-aware augmentation, synthetic data simulation, and sensor fusion with modalities such as infrared or LiDAR can improve robustness in these contexts.

6.4 Error analysis and mitigation strategies

A closer inspection of errors revealed that most misclassifications occurred with visually similar vehicle classes (e.g., vans vs. minibuses), while missed detections often arose under heavy occlusion or extreme scale variation. These errors highlight the limitations of both handcrafted feature descriptors and temporal modeling in isolation. Incorporating multi-scale attention mechanisms, finer-grained class descriptors, and uncertainty modeling could help reduce these shortcomings. Furthermore, feedback-based learning with active error correction may improve adaptability in dynamic environments.

7 Conclusion

This work presents a comprehensive deep learning-based framework tailored for intelligent traffic surveillance using UAV-acquired imagery. The proposed system integrates Self-Adaptive Spectral-Spatial Clustering (SASSC) for image segmentation, RNN for precise vehicle detection, DeepSORT for consistent multi-object tracking, and a Swin Transformer-based classifier leveraging SURF and BRISK features for accurate vehicle categorization. Evaluated on the VAID, AU-AIR, and UAVDT datasets, the system achieved detection precisions of 0.913, 0.930, and 0.920; tracking precisions of 0.901, 0.881, and 0.890; and classification accuracies of 92.14% for AU-AIR, 92.75% for the UAVDT, and 91.25 for VAID%, respectively. These results underscore the framework's effectiveness, robustness, and adaptability across diverse aerial traffic scenarios. Designed with a focus on computational efficiency, the architecture supports deployment on low-cost and energy-constrained UAV platforms. Future work will target real-time implementation under challenging environmental conditions, enhance scalability to larger urban areas, and integrate predictive analytics to support proactive traffic flow management in smart cities.

Data availability statement

Publicly available datasets were analyzed in this study. This data can be found at: https://github.com/paperswithcode/paperswithcode-data https://www.kaggle.com/datasets/foryolotrain1/uavdt-2024.

Author contributions

MA: Methodology, Writing – review & editing. TW: Formal analysis, Validation, Writing – review & editing. NA: Conceptualization, Methodology, Writing – review & editing. YA: Methodology, Conceptualization, Writing – review & editing. MH: Methodology, Validation, Writing – original draft. AJ: Supervision, Writing – review & editing. HL: Formal analysis, Methodology, Writing – review & editing.

Funding

The author(s) declare that financial support was received for the research and/or publication of this article. The APC was funded by the Open Access Initiative of the University of Bremen and the DFG via SuUB Bremen. This work was supported through Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2025R410), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia. The authors extend their appreciation to the Deanship of Research and Graduate Studies at King Khalid University for funding this work through Large Group Project under grant number (RGP2/367/46).

Acknowledgments

The authors acknowledge Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2025R410), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia. The authors extend their appreciation to the Deanship of Research and Graduate Studies at King Khalid University for funding this work through Large Group Project under grant number (RGP2/367/46).

Conflict of interest

HL was employed by Guodian Nanjing Automation Co., Ltd.

The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Generative AI statement

The author(s) declare that no Gen AI was used in the creation of this manuscript.

Any alternative text (alt text) provided alongside figures in this article has been generated by Frontiers with the support of artificial intelligence and reasonable efforts have been made to ensure accuracy, including review by the authors wherever possible. If you identify any issues, please contact us.

Publisher's note

References

Alazeb, A., Hanzla, M., Mudawi, N. A., Alshehri, M., Alhasson, H. F., et al. (2025). Nighttime intelligent UAV-based vehicle detection and classification using YOLOv10 and swin transformer. *Comp. Mater. Continua* 84, 4677–4697. doi: 10.32604/cmc.2025.065899

Alharbi, M. Alshammari, F., Almujally, N., Alqahtani, R., and Alhasson, H. (2022). Vehicle detection using random forests in UAV aerial imagery. *Remote Sensing Lett.* 13, 312–318. doi: 10.1080/2150704X.2022.2021234

Almujally, N. A., Qureshi, A. M., Alazeb, A., Rahman, H., Sadiq, T., Alonazi, M., et al. (2024). A novel framework for vehicle detection and tracking in night ware surveillance systems. *IEEE Access* 12, 88075–88085. doi: 10.1109/ACCESS.2024.3417267

Alonazi, M., Qureshi, A. M., Alotaibi, S. S., Almujally, N. A., Al Mudawi, N., Alazeb, A., et al. (2023). A smart traffic control system based on pixel-labeling and SORT tracker. *IEEE Access* 11, 80973–80985. doi: 10.1109/ACCESS.2023.3299488

Ammour, N., Alhichri, H., Bazi, Y., Benjdira, B., Alajlan, N., Zuair, M., et al. (2017). Deep learning approach for car detection in UAV imagery. *Remote Sens.* 9:312. doi: 10.3390/rs9040312

Arinaldi, A., Pradana, J. A., and Gurusinga, A. A. (2018). Detection and classification of vehicles for traffic video analytics. *Procedia Comp. Sci.* 144, 259–268. doi: 10.1016/j.procs.10, 527.

Battal, A., Avci, Y. E., and Tuncer, A. (2023). "*Vehicle detection and counting in traffic videos using deep learning*," in *Proceedings of the International Conference on Engineering Technologies (ICENTE '23), Konya, Turkey, 23–25 Nov 2023* (Turkey: IEEE Xplore), 272–275.

Bewley, A., Ge, Z., Ott, L., Ramos, F., and Upcroft, B. (2016). "Simple online and real-time tracking," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)* (Phoenix, AZ: IEEE Xplore), 3464–3468. doi: 10.1109/ICIP.2016.7533003

Bianchi, D., Borri, A., Cappuzzo, F., and Di Gennaro, S. (2024). Quadrotor trajectory control based on energy-optimal reference generator. *Drones* 8:29. doi: 10.3390/drones8010029

Biyik, M. Y., Atik, M. E., and Duran, Z. (2023). Deep learning-based vehicle detection from orthophoto and spatial accuracy analysis. *Int. J. Eng. Geosci.* 8, 138–145. doi: 10.26833/ijeg.1080624

Chandramohan, D., Dumka, A., and Jayakumar, L. (2020). 2M2C R2ED: multi-metric cooperative clustering based routing for energy-efficient data dissemination in green VANETs. *Technol. Econ. Smart Grids Sustainable Energy* 5:15. doi: 10.1007/s40866-020-00086-4

Cui, X., Wang, Y., Yang, S., Liu, H., and Mou, C. (2022). UAV, fixed-point path planning, multi-point path planning, two-point path planning, co-evolutionary algorithm. *Front. Neurorob.* 16, 1662–5218. doi: 10.3389/fnbot.2022.1105177

du Terrail, J. O., and Jurie, F. (2018). Faster RER-CNN: application to the detection of vehicles in aerial images. *arXiv [Preprint]*. doi: 10.48550/arXiv.1809.07628

Du, L., He, L., Ye, X., and Xu, Z. (2018). "UAVDT: unmanned aerial vehicle benchmark for detection and tracking," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (Salt Lake City, UT: CVF), 3133–3141.

ElSahly, O., and Abdelfatah, A. (2023). An incident detection model using random forest classifier. *Smart Cities* 6, 1786–1813. doi: 10.3390/smartcities6040083

Gallo, I., Rehman, A. U., Dehkordi, R. H., Landro, N., La Grassa, R., and Boschetti, M. (2023). Deep object detection of crop weeds: performance of YOLOv7 on a real case dataset from UAV images. *Remote Sens.* 15:2. doi: 10.3390/rs15020539

Hanzla, M., Ali, S., and Jalal, A. (2024). "Smart traffic monitoring through drone images via YOLOv5 and Kalman filter," in *Proceedings of the 5th International Conference on Advancements in Computational Sciences (ICACS), Lahore, Pakistan* (Lahore: IEEE Xplore), 1–8. doi: 10.1109/ICACS60934.2024.10473259

Hanzla, M., and Jalal, A. (2025). "Intelligent transportation surveillance via YOLOv9 and NASNet over aerial imagery," in *6th International Conference on Advancements in Computational Sciences (ICACS), Lahore, Pakistan* (Lahore: IEEE Xplore), 1–8, doi: 10.1109/ICACS64902.2025.10937840

Khan, M., Akram, S., and Ali, H. (2021). Efficient deep learning model for object classification on AU-AIR dataset. *IEEE Access*, 9, 5804–5812.

Kumar, R., Singh, A., and Sharma, P. (2019). Object detection and tracking in UAV videos using deep learning: a case study on AU-AIR dataset. *Int. J. Comp. Appl.* 178, 34–39.

Lin, H-. Y., Tu, K-. C., and Li, C-. Y. (2020). VAID: an aerial image dataset for vehicle detection and classification. *IEEE Access* 8, 212209–212219. doi: 10.1109/ACCESS.2020.3040290

Lin, T-. Y., Goyal, P., Girshick, R., He, K., and Dollar, P. (2017). "Focal loss for dense object detection," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)* (Venice: IEEE Xplore), 2999–3007. doi: 10.1109/ICCV.2017.324

Mei, J., and Zhu, W. (2024). BGF YOLOv10: small object detection algorithm from unmanned aerial vehicle perspective based on improved YOLOv10. *Sensors* 24:6911. doi: 10.3390/s24216911

Meinhardt, T., Kirillov, A., Leal Taixé, L., and Feichtenhofer, C. (2022). "TrackFormer: multi-object tracking with transformers," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* (New Orleans, LA: CVPRW), 1–10. doi: 10.1109/CVPR52022.00864

Moutakki, Z., Ouloul, I. M., Afdel, K., and Amghar, A. (2017). Real-time video surveillance system for traffic management with background subtraction using codebook model and occlusion handling. *Trans. Telecommun.* 18, 297–307. doi: 10.1515/ttj-2017-0027

Mudawi, N. A., Hanzla, M., Alazeb, A., Alshehri, M., Alhasson, H. F., et al. (2025). Remote sensing imagery for multi-stage vehicle detection and classification via YOLOv9 and deep learner. *Comp. Mater. Continua* 84, 4491–4509. doi: 10.32604/cmc.2025.065490

Mujtaba, G., Hanzla, M., and A. Jalal (2024b). "Drone-based road traffic surveillance: multi-vehicle tracking and classification," *5th International Conference on Innovative Computing (ICIC), Lahore, Pakistan* (Lahore: IEEE Xplore), 1–7, doi: 10.1109/ICIC63915.2024, 11116597.

Mujtaba, G., Hanzla, M., and Jalal, A. (2024a). "Smart traffic monitoring with efficientnet and neuro-fuzzy classifier via aerial surveillance," in *2024 26th International Multi-Topic Conference (INMIC), Karachi, Pakistan* (Lahore: IEEE Xplore), 1–6, doi: 10.1109/INMIC64792.2024.11004350

Mujtaba, G., Hanzla, M., and Jalal, A. (2025). "Drone surveillance for intelligent multi-vehicles monitoring and classification," in *6th International Conference on Advancements in Computational Sciences (ICACS), Lahore, Pakistan* (Lahore: IEEE Xplore), 1–6, doi: 10.1109/ICACS64902.2025.10937829

Nepal, U., and Eslamiat, H. (2022). Comparing YOLOv3, YOLOv4 and YOLOv5 for autonomous landing spot detection in faulty UAVs. *Sensors* 22:464. doi: 10.3390/s22020464

Nosheen, I., Naseer, A., and Jalal, A. (2024). "Efficient vehicle detection and tracking using blob detection and Kernelized filter," in *Proceedings of the 5th International Conference on Advancements in Computational Sciences (ICACS), Lahore, Pakistan* (Lahore: IEEE Xplore), 1–8. doi: 10.1109/ICACS60934.2024.10473292

Qureshi, A. M., Almujally, N. A., Alotaibi, S. S., Alatiyyah, M. H., and Park, J. (2023). Intelligent traffic surveillance through multi-label semantic segmentation and filter-based tracking. *Comput. Mater. Continua* 76, 3707–3725. doi: 10.32604/cmc.2023.040738

Qureshi, A. M., Alotaibi, M., and Alotaibi, S. R. AlHammadi, D. A., Jamal, M. A., Jalal, A., Lee, B. (2025). Autonomous vehicle surveillance through fuzzy C-means segmentation and DeepSORT on aerial images. *PeerJ Comp. Sci.* 11:e2835. doi: 10.7717/peerj-cs.2835

Rahman, M. S., and Hasan, M. (2022). Vehicle detection and classification in UAVDT dataset using HOG + SVM. *Int. J. Intell. Syst. Appl.* 14, 47–56.

Ren, S., He, K., Girshick, R., and Sun, J. (2017). Faster R-CNN: towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Machine Intell.* 39, 1137–1149. doi: 10.1109/TPAMI.2016.2577031

Sang, Z., Wu, Z., Guo, P., Hu, H., Xiang, H., Zhang, Q., et al. (2018). An improved YOLOv2 for vehicle detection. *Sensors* 18:4272. doi: 10.3390/s18124272

Sethi, P. N., Ullah, A., and Akhtar, N. (2020). "AU-AIR: a multi-modal UAV dataset for object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* (Seattle, WA: CVPRW), 1735–1744.

Sun, P., Cao, J., Jiang, Y., Zhang, R., Liu, W., Wang, Z., et al. (2020). "TransTrack: transformer-based object tracking," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* (Seattle, WA: CVPRW), 1–10.

Sundaresan, S. G. A., Alif, M. A. R., Hussain, M., and Allen, P. (2024). Comparative analysis of YOLOv8 and YOLOv10 in vehicle detection: performance metrics and model efficacy. *Vehicles* 6, 1364–1382. doi: 10.3390/vehicles6030065

Tiwari, P., and Gupta, P. (2023). Vehicle detection using Haar cascade classifier and VAID dataset. *Int. J. Comp. Vision Image Process.* 13, 115–130.

Vu, T. C., Tran, T. D., Nguyen, T. V., Nguyen, D. T., Dinh, L. Q., Nguyen, M. D., et al. (2025). Vehicle detection, tracking and counting in traffic video streams based on the combination of YOLOv9 and DeepSORT algorithms. *J. Future Artif. Intell. Technol.* 2, 255–268. doi: 10.62411/faith.3048-3719-115

Wang, C-. Y., Liao, H-. Y. M., and Lin, Y-. H. (2022). YOLOv7: trainable bag of freebies for real-time object detectors. *IEEE Access* 10, 1–15. doi: 10.1109/ACCESS.2021.3137641

Wang, Y., Wu, J., and Fang, Q. (2020). Vehicle detection in aerial imagery using deep convolutional networks: experiments on the UAVDT dataset. *Remote Sens.* 12:468.

Xu, H., Cao, Y., Lu, Q., and Yang, Q. (2020). "Performance comparison of small object detection algorithms of UAV-based aerial images," in *Proceedings of the 9th International Symposium on Distributed Computing and Applications for Business Engineering and Science (DCABES), Hong Kong* (Xuzhou: IEEE Xplore), 16–19. doi: 10.1109/DCABES50732.2020.00014

Xu, M., Li, Z., Wang, Y., Pan, H., and Li, Z. (2020). Vehicle detection based on improved multitask cascaded convolutional neural network and mixed image enhancement. *IET Image Processing*, 14, 1283–1290. doi: 10.1049/iet-ipr.2020.1005

Yang, H., and Qu, S. (2018). Real-time vehicle detection and counting in complex traffic scenes using background subtraction model with low-rank decomposition. *IET Intelligent Transport Systems*, 12, 75–85. doi: 10.1049/iet-its.2017.0047

Zhang, H., Zheng, F., Yang, Y., and Xiao, J. (2021). Aerial object recognition with UAVDT dataset: challenges and approaches. *IEEE Trans. Geosci. Remote Sens.* 59, 3657–3669.

Zhang, Q., Wang, X., Shi, H., Wang, K., Tian, Y., Xu, Z., et al. (2025). BRA YOLOv10: UAV small target detection based on YOLOv10. *Drones* 9:159. doi: 10.3390/drones9030159