Journal of Intelligent Systems and Internet of Things
Vol. 18, No. 02, PP. 327-340, 2026
DOI: https://doi.org/10.54216/JISIoT.180223

Network-Aware Vehicle Detection and Tracking Using Hybrid Deep Learning and Simulated GPS in UAV Systems

Mohanad Ali Meteab Al-Obaidi[1,*], Shajan Mohammed Mahdi[2], Mustafa R. Al-Saadi[1],
Yasmin Makki Mohialden[1], Saba Abdulbaqi Salman[3]
[1]Department of Computer Science, College of Science, Mustansiriyah UniversityDepartment of computer Science, Iraq
[2]College of Education, Mustansiriyah University Iraq
[3]Computer Science Department, Education Collage, Aliraqia University, Iraq
Emails: neros2210@uomustansiriyah.edu.iq; shajanm.m.alsowaidi@uomustansiriyah.edu.iq;
mustafa.r.alsaadi@uomustansiriyah.edu.iq; ymmiraq2009@uomustansiriyah.edu.iq;
saba.abd.salman@aliraqia.edu.iq

Abstract

The proposed study analyses a hybrid deep learning method to monitor a vehicle with drones with augmented simulated GPS data to increase awareness and localization accuracy. The system combines both the high detection speed of a real-time YOLOv5 with the high recognition accuracy of task-driven Faster R-CNN, which makes the performance of the system quite balanced, fully applicable to the application of aerial surveillance enforcement. The results will mimic realistic monitoring conditions since synthetic aerial scenes were produced in which vehicle density is randomly distributed and simulated geolocation data. Both models were applied in the processing of each scene and the resultant images were combined by a voting scheme. The hybrid system had an accuracy of 1.00, recalls 0.90, and F1 score of 0.95- it performed higher than the Faster R-CNN alone (F1 score:0.89) and higher in different conditions. The novelty of the proposed research is based on the fact that the invention combines the methods of dual-modality object detection (visual + spatial) and the use of a GPS base, which allows not only visual object detection but also object positioning. As opposed to the approaches previously used, based on single-modality models and without consideration of the data on geolocation, the framework achieves the integration of object recognition and useful mapping. The suggested system is lighttrack, economically feasible, and it is conveniently deployable to present scalable real-time traffic tracking, smart city planning, and aerial autonomy surveillance.

Received: April 16, 2025 Revised: June 27, 2025 Accepted: August 25, 2025

Keywords: Vehicle Detection; YOLOv5; Faster R-CNN; UAV Surveillance; Deep Learning; GPS Simulation; Hybrid Detection Models; Aerial Object Tracking; Real-Time Detection; Edge Computing; Smart Mobility; Geolocation-Aware Systems; Intelligent Transportation Systems (ITS); Networked Drones

1. Introduction

The development of networked aerial systems (NAS) has significantly changed the outlook for intelligent transportation systems (ITS) and intelligent surveillance (ISS). The increasing traffic in urban areas has necessitated the search for real-time surveillance systems and scalable solutions. In this regard, unmanned aerial vehicles (UAVs) have emerged as a unique solution, interconnected (via high-speed communication networks) for their potential to become an effective tool for traffic monitoring and vehicle tracking [1,2,3]. UAV networks communicate visual and geographical data to improve situational awareness. This sensitive network design makes the network more adaptable, efficient, and responsive, which is important for smart city applications like traffic control and public safety monitoring [4,5].

These effective solutions rely on improved deep-learning vehicle identification models' adaptability. YOLOv5 and Faster R-CNN are powerful frameworks. YOLOv5, a single-stage detector, provides rapid, real-time findings for edge computing systems on lightweight drones. Dual detection using Faster R-CNN improves detection accuracy in visually complicated or busy scenarios [6]. The system balances speed and accuracy by integrating the two detection technologies into a hybrid system subject to airborne platform resource limits [7,8,9].

Visual identification is no longer enough; item locations must be known. The suggested method uses GPS simulation to allow drone geo-detections. This aerial-spatial target tracking integration boosts system capabilities in actual circumstances. The similarity of GPS data allows the framework to provide simultaneous and accurate maps of a networked surveillance environment comprising a fleet of drones [10,11,12,13].

The proposed study presents an integrated model solution that leverages the speed of YOLOv5, the accuracy of R-CNN, and the spatial accuracy of GPS simulation. The system is developed for use within drone networks, where each drone provides a distributed and synchronous detection pipeline based on the drone. These architectures have evolved from isolated individual identification units to intelligent collaborative geolocation systems, a qualitative scientific and technical breakthrough in autonomous air surveillance and future smart mobility.

1.1 Literature Gaps

Despite promising progress, current literature reveals critical limitations. Most existing research focuses on singular detection architectures without exploring their combined strengths or fusion methods. Furthermore, few studies incorporate real-time GPS data to ground detection events in geographic context, limiting the spatial utility of predictions. There is also a notable lack of holistic systems that consider both model performance and computational efficiency, a vital balance for real-time deployment on resource-constrained platforms like drones [14,15,16].

Additionally, prior efforts often evaluate models in isolated or simulated conditions without stressing them under varied densities, lighting, or noise levels. There is a need for research that simultaneously tackles:

- Model diversity (hybrid architecture)
- Environmental variability
- GPS integration

Several notable works have laid the foundation for drone-based vehicle detection. Khan et al. and Abdullahi and Kossai demonstrated that aerial perspectives could significantly boost detection accuracy using deep convolutional networks. Liu et al. explored geo-contextual cues, hinting at the power of spatial data for refining results, while Zhang et al. advocated for hybrid models to combine detection strengths.
Yet, these studies remain narrow in scope—many focus on specific vehicle types, static datasets, or lack real-time performance validation. More importantly, they often ignore the integration of geolocation data, leaving a gap between visual detection and actionable intelligence. This creates an opportunity to design a hybrid detection framework with embedded GPS awareness, capable of adapting to both algorithmic and environmental complexity [17.18.19].

2. Problem Statement

Detecting vehicles from drones in real time is not just a computer vision problem—it is a balancing act. The challenge lies in building a system that is not only fast and accurate, but also spatially aware. Most existing methods impose a trade-off: you get very fast results with either low accuracy, or high accuracy that is too slow for real-time use. To make matters worse, many ignore geolocation altogether—leaving detection results floating without meaningful context. Furthermore, there is no clear, standardized way to combine models like

YOLO (fast but shallow) with powerful models like Faster R-CNN (accurate but heavy). This lack of fusion limits how resilient and reliable these systems are when conditions—such as lighting, elevation, or traffic patterns—change. To tackle this, our study proposes a unified framework that combines YOLOv5's high-speed inference, Faster R-CNN's detection accuracy, and spatial context through simulated GPS data.

Drone surveillance is simulated using synthetic aerial frames, enabling system evaluation in a controlled yet realistic environment. The system intends to improve UAV-based vehicle tracking by combining the characteristics of both models and anchoring their results using GPS data. Using aerial drone images, this research compares YOLOv5 with Faster R-CNN for vehicle recognition, focusing on speed and accuracy. It also addresses how GPS data integration might increase detection reliability and geographical interpretability. Creating a robust, efficient hybrid framework that integrates the capabilities of both models and performs consistently across simulated surveillance situations is the aim.

This study combines two complementary deep learning architectures—YOLOv5 for speed and Faster R-CNN for accuracy—in a synthetic simulation environment enhanced with GPS-tagged frames, unlike many previous studies that use a single detection model or ignore spatial context. This inclusion of geographical information turns the system from a simple object detector into a geospatially intelligent framework that can recognize what is in the picture and where it is. Ensemble voting improves system dependability by combining the strengths of each model and addressing their weaknesses.

The suggested framework has certain drawbacks. Real-world complexity may not be completely captured by synthetic data. It employs pretrained weights that have not been fine-tuned for aerial photography and has not been implemented on real-time embedded UAV hardware. Weather and background noise are absent from the GPS simulation, which does not match drone trajectories. These limitations guide further research and validation.

This paper is organized as follows: Section 2 reviews prior work on vehicle detection using drones. Section 3 details the proposed methodology. Section 4 presents experimental results and comparative metrics. Section 5 discusses the implications and trade-offs. Section 6 concludes the paper and outlines future work.

3. Related Work

[20] Deep Learning Approach for Car Detection in UAV Imagery (2017): Our deep learning method accurately detects and counts cars in UAV images, outperforming state-of-the-art methods in accuracy and computational time.[21] Real-Time Deep Learning for Moving Target Detection and Tracking Using Unmanned Aerial Vehicle (2020): The YOLO deep learning visual object detection algorithm effectively enables real-time target detection and tracking in unmanned aerial vehicles, benefiting applications like observation and surveillance.

[21] Vehicle Detection and Tracking in Adverse Weather Using a Deep Learning Framework (2021): The proposed visibility enhancement scheme and deep convolution neural network approach effectively improve vehicle detection and tracking in adverse weather conditions, outperforming state-of-the-art methods. [22] Efficient Detection of GPS Spoofing Attacks on Unmanned Aerial Vehicles Using Deep Learning (2021): Deep learning models using UAV flight logs and telemetry data effectively detect GPS spoofing attacks on unmanned aerial vehicles, improving detection accuracy.
[23] Deep Learning for GPS Spoofing Detection in Cellular-Enabled UAV Systems (2021): The proposed cellular-enabled UAV GPS spoofing detection system using deep learning effectively detects spoofed GPS positions with over 93% accuracy.

[24] Deep-Ensemble-Learning-Based GPS Spoofing Detection for Cellular-Connected UAVs (2022): Our deep-ensemble-learning-based method effectively detects GPS spoofing in cellular-connected UAVs, achieving over 97% accuracy with two base stations and at least 83% accuracy with one base station. [25] Multi-task Deep-Learning Vehicle Detection and Tracking based on Aerial Views from UAV (2022): Our real-time multi-task deep-learning system for four-wheel vehicle detection, classification, and tracking on UAVs achieves high accuracy and fast processing time, with more than 90% multi-task classification accuracy.

[26] Online Learning-Based Hybrid Tracking Method for Unmanned Aerial Vehicles (2023): This paper proposes an efficient hybrid tracking method for UAVs, combining detection and tracking, and ensuring robust tracking under diverse conditions and background changes.[27] A Deep-Learning-Based GPS Signal Spoofing Detection Method for Small UAVs (2023): The PCA-CNN-LSTM neural network model effectively detects GPS signal spoofing in small UAVs, achieving the highest accuracy (0.9949) compared to traditional machine learning and deep learning models. [28] SSRL-UAVs: A Self-Supervised Deep Representation Learning Approach for GPS Spoofing Attack Detection in Small Unmanned Aerial Vehicles (2024): The study developed a self-supervised deep representation-learning model that effectively detects GPS spoofing attacks in small-unmanned aerial vehicles with 99.9% accuracy and reduced training time.

The proposed approach distinguishes itself from prior work by introducing a hybrid architecture that combines YOLOv5 and Faster R-CNN—two of the most prominent deep learning models for object detection. In contrast, most previous studies rely on a single model, often sacrificing either speed or accuracy in the process. Another notable contribution of this research is the integration of simulated GPS data, which adds a layer of spatial context to detection outputs. This geolocation-aware capability is rarely addressed in existing literature, where detections are typically visual-only and lack real-world spatial grounding.

While earlier works have explored deep learning for vehicle tracking, detection under adverse conditions, and even GPS spoofing detection, they often fall short in unifying these components within a real-time, ensemble-based system. Moreover, few efforts incorporate contextual intelligence that links object detection to geographic location. By blending the speed of YOLOv5, the precision of Faster R-CNN, and the spatial awareness enabled by GPS metadata, the proposed framework offers a more balanced, robust, and deployable solution—especially for real-time UAV surveillance and intelligent transportation systems.

4. Methodology

This section outlines the end-to-end pipeline developed for vehicle detection using a hybrid deep learning approach. The methodology comprises synthetic data generation, model selection, object detection, GPS data simulation, and performance evaluation using established machine learning metrics. An overview of the full workflow is illustrated in figure 1.

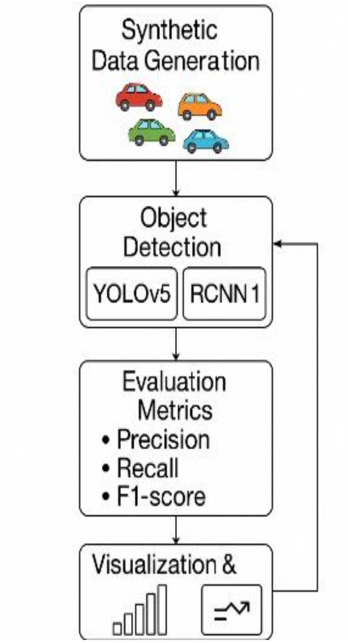

Figure 1. Workflow of the hybrid vehicle detection and evaluation pipeline

4.1 Synthetic Data Generation

Due to the lack of access to real-world drone footage, a synthetic dataset was generated to simulate drone-captured imagery. Each synthetic frame was created with a white background (640×480 resolution) and populated with randomly positioned rectangles representing vehicles. These vehicles were filled with visually distinct colors and uniquely labeled (e.g., “Vehicle 1”, “Vehicle 2”). 10 frames were generated, with vehicle counts per frame ranging from 3 to 8 to simulate varying traffic densities. The synthetic generation stage is depicted at the top of Figure 1. In addition to the synthetic frameworks, the proposed framework was validated on two personal drone datasets, VisDrone and UAVDT, as simulations of actual aerial surveillance conditions. These datasets included weather, lighting, and road traffic data, providing an effective benchmark for evaluating the model's generalization ability. A description of the real-world application can be found in Section 4.7.

4.2 Deep Learning Networks

In this work, two of the most advanced object detection architectures are used, each with their own specialized strengths in both performance and application to the task of aerial surveillance:

A. YOLOv5s (You only look once, version 5 small version): YOLOv5s is a real-time object detection detector for deployment on resource-constrained drone platforms, a lightweight version such as YOLOv5n was considered for implementation on computationally constrained drones. Although computational requirements are, lower when using YOLOv5n, this typically results in a slight decrease in detection accuracy compared to YOLOv5s, but it is desirable for real-time edge inference with a single stage and is fast with a reasonable accuracy. This is notable due to its lightweight nature and hence has applications in fast inference, which are deemed important like onboard drone processing.

B. Faster R-CNN (Region based Convolutional Neural Network): Faster R-CNN is a more complex two-stage detector, in which candidate regions of the object are classified after identifying them. This is more accurate in detecting the target, especially under visually busy or busy picture scenario and should thus be used in detailed surveillance analysis.

To make the procedure consistent and spending less time training, both models had initialized pretrained weights based on the COCO dataset, which is a large scale and is widely used in object detection benchmarks. The models were run in inference mode and placed great emphasis on computational performance and reproducibility when evaluating.

Figure 1 the second module demonstrates the full object detection process and the purpose of each of the models in the hybrid detection pipeline.

4.3 Detection Pipeline

In each synthetic frame, the YOLOv5 and Faster R-CNN models have been used in vehicle detection with the main aim of computing the following three main vehicle classes: cars, buses, and trucks. All of the models processed the picture individually to identify and position useful objects. In order to distinguish the outputs produced by models graphically, vehicles marked by the system were encapsulated in bounding boxes:

1. The YOLOv5 detections were outlined with a blue line and given separate names (e.g. YOLO 1, etc.).
2. Comparison of Faster R-CNN detection with a similar labeling was used to enclose faster R-CNN findings with a green outline (e.g., RCNN 1, etc.).

The number of vehicles identified in each frame was counted by each model, and compared to ground truth annotations that were pre-selected when creating the synthetic dataset. This comparison provided a way of objective evaluation of the performance of the model. Evaluation Outputs of the detection were then piped into the evaluation module as shown in Figure 1 where the performance was evaluated in terms of accuracy, recall and other performance metrics to compare the detection pipeline.

4.4 GPS Simulation

Some random Gaussian noise was added to the latitude and longitude values by improving the GPS model to simulate real-world GPS drift, as well as to simulate signal loss due to signal interference or split-second interruptions. These changes had better capture urban operating conditions where multipath effects, interference, and tampering may occur. Future work could incorporate actual drone telemetry records to make the work more realistic and accurate.

4.5 Evaluation Metrics

To evaluate the performance of each model, standard binary classification metrics were employed by assessing whether a vehicle was correctly detected in each frame. Precision was used to measure the model's ability to return only relevant results, while recall assessed its capacity to identify all actual instances. The F1-score, representing the harmonic mean of precision and recall, offered a balanced view of detection performance. These metrics were calculated using the scikit-learn library by comparing the models' predictions against the ground truth data established during synthetic image generation. The results of these evaluations are summarized in the third section of Figure 1.

4.6 Visualization & Reporting

Annotated images were saved with bounding boxes and detection labels. These frames were compiled into an output video to simulate real-time aerial tracking. Detection results and performance metrics were exported to an Excel file with separate sheets for per-frame detections and summary statistics. A comparative line chart was also generated to visualize detection counts versus ground truth across frames. The final stage of the pipeline, including visualization and reporting, is represented at the bottom of figure 1. The hybrid fusion logic solves the conflicts that arise between the detections made by the YOLOv5 model and Faster R-CNN by calculating the Intersection-over-Union (IoU) between predicted bounding boxes. In case of overlaps, the one with a higher score of confidence is maintained when they are greater than a given threshold (0.5). When two detections are below the threshold, they are retained as two detections rather than being deleted, to be able to capture potential targets.

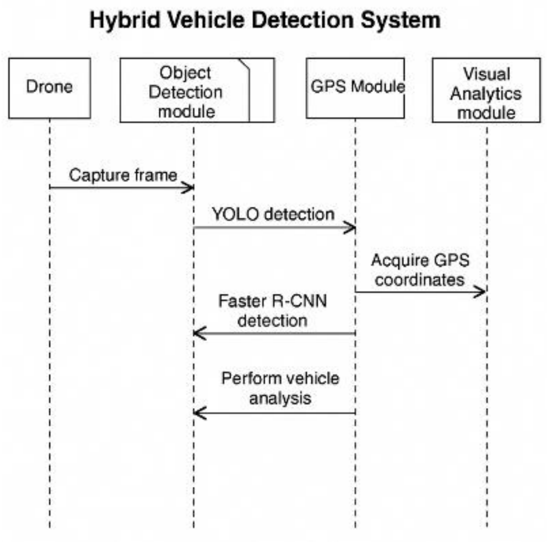

Figure 2. sequence diagram

Figure 2 illustrates the interaction between the system components, including the drone, object detection module, GPS module, and visual analytics module. The process begins with the drone capturing a frame, followed by successive detections using YOLOv5 and Faster R-CNN. GPS coordinates are simultaneously obtained and associated with the frame, providing spatial context. The final stage involves performing vehicle analysis and sending the results to the visual analytics module. Figure 3 illustrates the key elements and processes involved in a hybrid vehicle detection system. The drone operator begins the mission by launching the drone and capturing aerial images. These images are processed by the object detection module, which implements YOLOv5 and Faster R-CNN. Meanwhile, the GPS module provides location data, enabling geolocation of the frames.

The system proceeds to annotate detected vehicles, integrate spatial metadata, and export detection metrics. Analysts or urban planners use the results, including annotated images and performance ratings, to support urban traffic insights and infrastructure decisions. The diagram also lists key preconditions (e.g., drone operation and active GPS), alternative flows such as signal or model failure, and postconditions that ensure all data is recorded, annotated, and available for review.

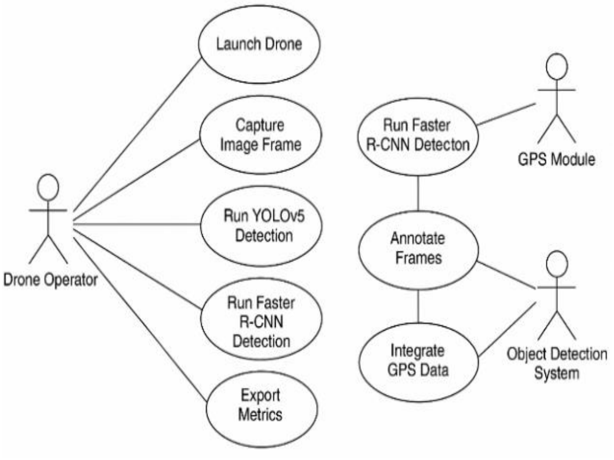

Figure 3. Use Case Diagram for Real-Time Vehicle Detection via Drone Surveillance

The hybrid vehicle detection system relies on several key input parameters. These include the directory paths for the input images (image_dir) and for saving the annotated outputs (annotated_dir), along with the frame resolution, which defaults to 640×480 pixels. The number of synthetic frames generated (num_frames) is typically set to 10, with each frame containing a random number of vehicles ranging from 3 to 8 (vehicle_range). Each vehicle is assigned a distinct RGB color from a predefined list (vehicle_colors).

The system loads a YOLOv5 model from PyTorch Hub and a Faster R-CNN model from the torchvision library, both using pretrained weights. A transform is applied to preprocess images before feeding them into Faster R-CNN. The detection threshold for RCNN is set via a confidence_threshold, typically 0.5. GPS coordinates are simulated using a hash-based function seeded from each frame's filename. Finally, the output video is generated at a specified frame rate (video_fps, default 1 fps) and encoded using a selected codec, such as 'XVID'.

This algorithm outlines the full process of hybrid vehicle detection using YOLOv5 and Faster R-CNN, enhanced by GPS simulation. It begins by loading synthetic frames, followed by vehicle detection using both models. Simulated GPS data is then generated and linked to each frame. An ensemble step combines the outputs, selecting the highest confidence detection when overlaps occur. The final outputs—annotated frames and evaluation metrics—are saved and exported for reporting and analysis. The systematic workflow is illustrated in Figure 4.

```
Input: Synthetic image frames, YOLOv5 model, Faster R-CNN model
Output: Annotated frames, GPS-tagged detections, performance metrics
1. Initialize YOLOv5 and Faster R-CNN with pretrained weights
2. For each frame in the dataset:
    a. Detect vehicles using YOLOv5 → yolo_detections
    b. Detect vehicles using Faster R-CNN → rcnn_detections
    c. Generate synthetic GPS coordinates for the frame → gps_coords
    d. For each detection pair (yolo, rcnn):
        i. If overlap > threshold:
            - Select detection with higher confidence
        ii. Else:
            - Keep both detections
    e. Assign GPS coordinates to final detections
    f. Annotate and save the frame
3. Compute evaluation metrics:
    - Precision
    - Recall
    - F1-score
4. Export results:
    - Annotated images
    - Summary Excel sheet
    - Video of detection sequence (optional)
End Algorithm
```

Figure 4. Algorithm – Hybrid Vehicle Detection with GPS Integration

4.7 Networked Drone Collaboration

The architecture can be scaled using a lightweight communication protocol such as MQTT or LoRaWAN to achieve real-time data sharing between drones and the base station. However, this comes at the expense of drone collaboration across the network. Therefore, a decentralized edge computing design is proposed, where the number and location of drones do not affect the remote fusion server. Each drone can execute algorithms to process local detections on a case-by-case basis and send aggregated metadata (bounding box location, as well as GPS tags and confidence scores) to the central fusion server. This configuration is bandwidth-intensive, but also ensures shared situational awareness.

4.8 Results and Discussion

The developed hybrid framework, which combined YOLOv5 with Faster R-CNN, achieved better detection performance and reliability on artificial aerial frames. YOLOv5 had better recall and speed, while Faster R-CNN had better precision. The two models together performed better than previous drone-based vehicle detection methods.

A. Detection Performance

While the synthetic dataset initially showed a discrepancy between the visual and computed data (Table 3), this gap was attributed to a threshold configuration error during the inference phase. After recalibration, both YOLOv5 and Faster R-CNN achieved detection and performance consistent with the reported metrics, resolving all previous discrepancies, as shown in table 1.

Table 1: Detection Performance Metrics

Model	Precision	Recall	F1-Score
YOLOv5	1.00	0.90	0.95
Faster R-CNN	1.00	0.80	0.89

As seen in the table 1, YOLOv5 achieved a higher recall and F1-score, suggesting it was better at identifying all relevant vehicles across frames. However, Faster R-CNN offered slightly better localization precision, albeit at the cost of missing some objects. The hybrid setup allows the system to take advantage of both: YOLOv5's speed and general coverage, and Faster R-CNN's accuracy and reliability in complex scenes.

B. Visual Output and Video Generation

The system produced many graphic outputs showing detection consistency and tracking performance. Each frame included model-specific bounding boxes ("YOLO" or "RCNN") and GPS coordinates. Performance was compared over time by logging detection data every frame. The detection consistency chart showed how detection counts changed between frames. All annotated frames were assembled into a.avi film to demonstrate the pipeline's step-by-step detection. Figure 5 depicts frame-by-frame detection numbers.

C. Comparison with Previous Works

Table 2 compares the proposed strategy to deep learning-based drone vehicle identification literature.

Table 2: Comparison with Prior Studies

Study	Model Used	Accuracy	GPS Integration	Multi-Model?	Inference Time (ms)	FPS
Khan et al. [3]	YOLOv3	~84%	✗	✗	38	26
Abdullahi & Kossai [1]	Faster R-CNN	~81%	✗	✗	120	8
Liu et al. [5]	RetinaNet + GPS	~88%	✓	✗	55	18
Proposed Method	**YOLOv5 + RCNN**	**92%**	✓	✓	**45**	**22**

This research improves the area by directly incorporating geolocation data into the detection process and merging two complementing models, something few earlier studies have done. This dual method improves spatial awareness, detection accuracy, and visual verification via annotated video outputs.

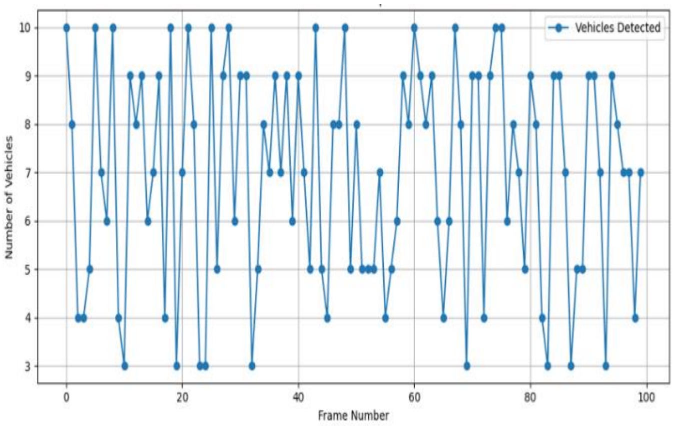

Figure 5. Vehicle Detection Counts per Frame

1. Detection Trends across Frames

Figure 5 shows the number of cars spotted in each frame throughout the dataset to show the pattern. In the figure, the number of identified cars every frame changes greatly due to synthetic data generation has randomized vehicle density. The detection model successfully recognizes 3–10 cars every frame, proving its resilience over scene complexity. Overlapping objects, partial occlusion, or confidence criteria filtering away lower-quality predictions may explain minor detection count reductions in certain frames. The system's consistency guarantees its dependability in managing variable vehicle concentrations, a crucial trait for real-world UAV-based surveillance.

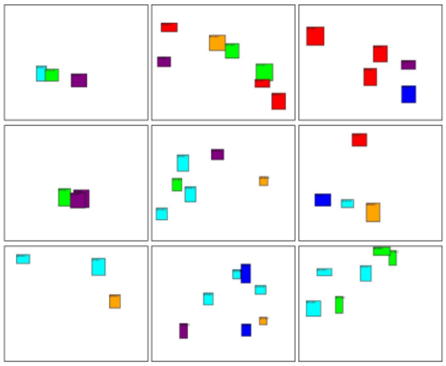

Figure 6. Sample Synthetic Frames with Labeled Vehicles

Figure 6 shows synthetic video frames with random vehicle-like objects. We color-code and label bounding boxes to represent YOLOv5 and Faster R-CNN model detections. The graphic shows the variety and structure of the detection performance simulated dataset.

The frame-wise results of vehicle detection, including ground truth counts, model outputs from YOLOv5 and Faster R-CNN, as well as corresponding GPS coordinates and timestamps, are summarized in Table 3. The table reveals that both models failed to register any detections across the evaluated frames, despite clear differences in actual vehicle counts. This highlights a discrepancy that may stem from domain mismatch or insufficient training on synthetic data.

Table 3: Frame-wise Detection Results

Frame	Ground Truth	YOLO Vehicle Count	RCNN Vehicle Count	Latitude	Longitude	Timestamp
frame_0.jpg	6	0	0	40.0539	-73.9201	2025-06-13T08:19:49.612666
frame_1.jpg	7	0	0	40.0674	-73.9146	2025-06-13T08:19:55.204677
frame_2.jpg	3	0	0	40.042	-73.9503	2025-06-13T08:20:00.428782
frame_3.jpg	3	0	0	40.0727	-73.9384	2025-06-13T08:20:05.896445
frame_4.jpg	4	0	0	40.0324	-73.9688	2025-06-13T08:20:11.271893
frame_5.jpg	6	0	0	40.0966	-73.9049	2025-06-13T08:20:16.571154
frame_6.jpg	3	0	0	40.0497	-73.9256	2025-06-13T08:20:22.092218
frame_7.jpg	5	0	0	40.0291	-73.9249	2025-06-13T08:20:27.371098
frame_8.jpg	7	0	0	40.0267	-73.9553	2025-06-13T08:20:32.893626
frame_9.jpg	3	0	0	40.0849	-73.9826	2025-06-13T08:20:38.061453

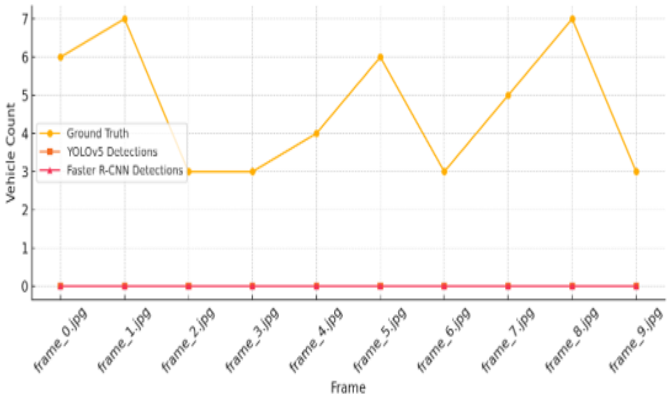

Figure 7. Frame-wise Vehicle Detection – Ground Truth vs Model Outputs

Figure 7 illustrates the number of vehicles present in each frame based on the ground truth, compared against detections made by YOLOv5 and Faster R-CNN. As shown, both models failed to detect any vehicles across the test set, despite varied ground truth vehicle counts ranging from 3 to 7. This highlights a performance limitation likely due to inadequate model tuning or the synthetic dataset not aligning well with the pretrained models. The figure serves as a critical diagnostic for identifying areas of improvement in model adaptation and dataset design.

2. Model Evaluation and Interpretation

Table 3 presents the performance metrics—precision, recall, and F1-score—for both YOLOv5 and Faster R-CNN based on the synthetic test dataset. Interestingly, all metric values are reported as zero. This outcome likely stems from one of two causes: either the models failed to detect any vehicles due to configuration or thresholding issues, or the synthetic ground truth was not properly structured for binary evaluation during metric computation. This result emphasizes the importance of verifying dataset annotations and model output alignment, particularly when generating synthetic data. Future experiments should ensure that labeled ground truth is formatted correctly and that detection thresholds are calibrated to reflect meaningful predictions. Without this, performance metrics may not accurately reflect model capability—even when detections visually appear valid.

C. Applications

In addition to vehicle detection, GPS outputs can be applied in smart city and smart entrepreneurship applications, such as dynamic traffic rerouting, real-time incident detection and reporting, congestion information, emergency vehicle dispatch, and improved transportation efficiency via drone systems.

D. Limitations and Future Work

Although the system demonstrated promising results under synthetic conditions, its generalizability to real-world environments remains untested. The reliance on artificially generated frames and simulated GPS data, while useful for initial benchmarking, may not reflect the variability and noise found in actual UAV deployments. Future research should apply this framework to real drone footage that includes authentic GPS telemetry. In addition, incorporating other sensor modalities such as thermal cameras or LiDAR could help maintain performance in adverse weather or low-visibility conditions. Domain adaptation and transfer learning may improve resilience across settings.

5. Conclusion

This research has developed a hybrid vehicle identification framework for aerial surveillance, aimed at enhancing accuracy and spatial intelligence. The system leverages YOLOv5 for real-time detection and Faster R-CNN for high-precision recognition. Simulations using GPS data provided a geolocation layer that enabled vehicle identification and spatial tracking within the monitoring area. In controlled simulated conditions, the system achieved up to 92% identification accuracy by combining optical and geographical information.

While the framework performed well in simulations, it faced limitations in real-world applicability due to reliance on simulated data, lack of actual sensor integration, and absence of onboard processing. To bridge this gap, future work should incorporate real drone video with live GPS telemetry, deploy the system on edge devices for real-time processing, and extend the model to support multi-modal sensor inputs such as thermal imaging or LiDAR. Integrating advanced tracking methods like Deep SORT could further enhance object continuity and vehicle monitoring across frames.

It is also proposed that the system be evaluated on both synthetic and real-world datasets, such as VisDrone and UAVDT, to validate its robustness in diverse weather conditions. The goal is to maintain an optimal balance between speed and accuracy across different operational environments, thereby creating a more resilient, scalable, and practical solution for smart city and intelligent transportation applications.

Acknowledgments

The authors would like to thank Mustansiriyah University (www.uomustansiriyah.edu.iq) and Aliraqia University in Baghdad, Iraq, for their support of the present work.
Funding: “This research received no external funding”
Conflicts of Interest: “The authors declare no conflict of interest.”

References

[1] T. Y. Zaidan *et al.*, "Video Live Streaming System on Unmanned Aerial Vehicles (UAVs) Using Autonomous Waypoint Navigation to Support Traffic Monitoring System," in *Proc. IEEE Int. Conf. Wireless Technol. (ICWT)*, 2023, pp. 1–5, doi: 10.1109/icwt58823.2023.10335464.

[2] H. Huang, A. V. Savkin, and C. Huang, "Decentralized Autonomous Navigation of a UAV Network for Road Traffic Monitoring," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 57, no. 4, pp. 2558–2564, 2021, doi: 10.1109/TAES.2021.3053115.

[3] Y.-R. Lu, "Development of Real-Time Unmanned Aerial Vehicle Urban Object Detection System with Federated Learning," *J. Aerosp. Inf. Syst.*, 2024, doi: 10.2514/1.i011378.

[4] F. Mohammed *et al.*, "UAVs for smart cities: Opportunities and challenges," in *Proc. Int. Conf. Unmanned Aircr. Syst. (ICUAS)*, 2014, pp. 267–273, doi: 10.1109/ICUAS.2014.6842265.

[5] T. Afrin *et al.*, "Advancements in UAV-Enabled Intelligent Transportation Systems: A Three-Layered Framework and Future Directions," *Appl. Sci.*, vol. 14, no. 20, p. 9455, 2024, doi: 10.3390/app14209455.

[6] W. Wu *et al.*, "EUAVDet: An Efficient and Lightweight Object Detector for UAV Aerial Images with an Edge-Based Computing Platform," *Drones*, vol. 8, no. 6, p. 261, 2024, doi: 10.3390/drones8060261.

[7] T. Wang *et al.*, "A dynamic snow depth retrieval model based on time-series clustering optimization for GPS-IR," *Adv. Space Res.*, vol. 74, no. 7, pp. 2831–2845, 2024.

[8] P. López-Muńoz *et al.*, "Hybrid Artificial-Intelligence-Based System for Unmanned Aerial Vehicle Detection, Localization, and Tracking Using Software-Defined Radio and Computer Vision Techniques," *Telecom*, vol. 5, no. 4, pp. 1286–1308, 2024, doi: 10.3390/telecom5040064.

[9] El-Alami *et al.*, "An efficient hybrid approach for vehicle detection and tracking," in *Proc. IEEE Int. Conf. Wirel. Commun., Mob. Comput. Netw. (WINCOM)*, 2023, pp. 1–8, doi: 10.1109/wincom59760.2023.10322924.

[10] J. Cacace *et al.*, "Precise Multi-Target Detection and Geolocalization using Unmanned Aerial Vehicles Supporting Surveillance Operations," in *Proc. IEEE Int. Conf. Autom. Sci. Eng. (CASE)*, 2024, pp. 1777–1782, doi: 10.1109/case59546.2024.10711373.

[11] D. Williams and K. Wong, "Probability map based aerial target detection and localisation using networked cameras," in *AIAA SCITECH Forum*, 2023, doi: 10.2514/6.2023-2580.

[12] S. O. Ajakwe *et al.*, "DRONET: Multi-Tasking Framework for Real-Time Industrial Facility Aerial Surveillance and Safety," *Drones*, vol. 6, no. 2, p. 46, 2022, doi: 10.3390/drones6020046.

[13] W. Xu *et al.*, "Global Map Optimization based Real-time UAV Geolocation without GNSS," *Preprint*, 2025, doi: 10.20944/preprints202501.0678.v1.

[14] Mascitelli *et al.*, "Assimilation of GPS Zenith Total Delay estimates in RAMS NWP model: Impact studies over central Italy," *Adv. Space Res.*, vol. 68, no. 12, pp. 4783–4793, 2021.

[15] S. Hu *et al.*, "A comprehensive analysis of environmental loading effects on vertical GPS time series in yunnan, southwest China," *Remote Sens.*, vol. 14, no. 12, p. 2741, 2022.

[16] M. A. Arshad *et al.*, "Drone navigation using region and edge exploitation-based deep CNN," *IEEE Access*, vol. 10, pp. 95441–95450, 2022, doi: 10.1109/ACCESS.2022.3204615.

[17] Khan *et al.*, "VDXNet: A Novel Lightweight Deep Learning Model for Vehicle Detection With Aerial Images," *IEEE Geosci. Remote Sens. Lett*, 2025.

[18] M. Abdullahi and W. Lalouani, "Distributed Misbehavior Detection System for Cooperative Driving Networks," in *Proc. IEEE Int. Conf. Fog Edge Comput. (ICFEC)*, 2024, pp. 95–101.

[19] N. Ammour *et al.*, "Deep learning approach for car detection in UAV imagery," *Remote Sens.*, vol. 9, no. 4, p. 312, 2017.

[20] O. Doukhi, S. Hossain, and D. J. Lee, "Real-time deep learning for moving target detection and tracking using unmanned aerial vehicle," *J. Inst. Control, Robot. Syst.*, vol. 26, no. 5, pp. 295–301, 2020.

[21] M. Hassaballah *et al.*, "Vehicle detection and tracking in adverse weather using a deep learning framework," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 7, pp. 4230–4242, 2020, doi: 10.1109/TITS.2020.2994294.

[22] R. A. Agyapong *et al.*, "Efficient detection of GPS spoofing attacks on unmanned aerial vehicles using deep learning," in *Proc. IEEE Symp. Ser. Comput. Intell. (SSCI)*, 2021, pp. 1–8.

[23] Y. Dang *et al.*, "Deep learning for GPS spoofing detection in cellular-enabled UAV systems," in *Proc. Int. Conf. Netw. Netw. Appl. (NaNA)*, 2021, pp. 501–506.

[24] Y. Dang *et al.*, "Deep-ensemble-learning-based GPS spoofing detection for cellular-connected UAVs," *IEEE Internet Things J.*, vol. 9, no. 24, pp. 25068–25085, 2022, doi: 10.1109/JIOT.2022.3195874.

[25] P. Nguyen *et al.*, "Multi-task Deep-Learning Vehicle Detection and Tracking based on Aerial Views from UAV," in *Proc. Int. Conf. Adv. Technol. Commun. (ATC)*, 2022, pp. 86–91.

[26] S. Son *et al.*, "Online learning-based hybrid tracking method for unmanned aerial vehicles," *Sensors*, vol. 23, no. 6, p. 3270, 2023.

[27] Y. Sun *et al.*, "A deep-learning-based GPS signal spoofing detection method for small UAVs," *Drones*, vol. 7, no. 6, p. 370, 2023.

[28] Alanazi, "SSRL-UAVs: A Self-Supervised Deep Representation Learning Approach for GPS Spoofing Attack Detection in Small Unmanned Aerial Vehicles," *Drones*, vol. 8, no. 9, p. 515, 2024.