Real-Time Detection and Tracking of Military and Civilian Vehicles Using YOLOv9 and DeepSORT

Dr. B M Sagar
Department of Information Science & Engineering
RV College of Engineering ,
Bangalore Karnataka, *India*
sagarbm@rvce.edu.in

Aishwarya
Department of Information Science & Engineering
RV College of Engineering ,
Bangalore Karnataka, *India*
aishuborole12@gmail.com

***Abstract*— In response to the growing demands of national defense and border security, this paper presents a real-time, intelligent surveillance system for the detection, classification, and tracking of military and civilian vehicles. The system leverages advanced deep learning architectures—YOLOv8 and YOLOv9—for high-speed object detection, trained on a publicly available labeled dataset containing diverse vehicle types. Through comparative analysis using precision, recall, F1-score, and mAP metrics, YOLOv9 emerged as the superior model and was integrated with the DeepSORT tracking algorithm to maintain consistent object identity across video frames. The final system is implemented using Python and supports real-time performance on both GPU-enabled platforms and edge devices like NVIDIA Jetson. Extensive testing confirms the system's ability to accurately distinguish between military and civilian vehicles in varied conditions, offering a scalable, robust solution for defense surveillance. Extensive testing demonstrated that YOLOv9 achieved a mean Average Precision (mAP@0.5) of 76.8% and an inference speed of 52 FPS, making it suitable for real-time deployment in defense scenarios. When integrated with DeepSORT, the system maintained over 90% tracking consistency, even in the presence of occlusion and fast motion. This work lays the groundwork for future developments such as behavioral anomaly detection, automated alerts, and multi-camera integration, thereby enhancing situational awareness and decision-making in sensitive environments.**

Keywords—Object Detection, YOLOv9, Real-Time Vehicle Tracking, Military Surveillance, DeepSORT, Streamlit, Deep Learning.

I. INTRODUCTION

The increasing complexity of modern military threats—particularly in border zones, conflict regions, and high-security installations—has elevated the demand for intelligent surveillance systems capable of real-time situational awareness and threat identification [1]. Traditional surveillance systems, typically dependent on manual monitoring or rule-based video analytics, are often slow, error-prone, and difficult to scale. These limitations create vulnerabilities in national security, especially when rapid identification of hostile vehicles or unauthorized intrusions is required [2], [3].

Recent advancements in deep learning and computer vision have led to the development of robust and scalable object detection frameworks, capable of processing real-time video streams with high accuracy [4]. Among these, the You Only Look Once (YOLO) family of models has gained significant traction due to its ability to perform object classification and localization in a single pass, making it ideal for high-speed detection tasks [5], [6]. In parallel, multi-object tracking algorithms like DeepSORT have emerged to complement detection systems by preserving object identities across video frames, enabling consistent monitoring of vehicle movements and behaviors in dynamic environments [7]. Military zones, border crossings, and critical infrastructure sites frequently involve complex and cluttered scenes where both military and civilian vehicles operate in close proximity.

However, most existing surveillance systems are not equipped to handle the challenges of occlusion, variable lighting, or fast-moving objects, and often lack the intelligence to differentiate subtle visual cues that separate civilian cars from military trucks, tanks, or aircraft [9].

To address these limitations, this research proposes a deep learning-based system that combines the YOLOv9 object detection model with the DeepSORT tracking algorithm to identify, classify, and track military and civilian vehicles in real time. The system is trained on a labeled dataset curated from Mendeley Data and other public sources, representing diverse vehicle categories and environmental conditions. YOLOv8 and YOLOv9 are both evaluated in terms of mean Average Precision (mAP), precision, recall, and F1-score, with YOLOv9 demonstrating superior performance for deployment.

The system is designed to run on both high-performance GPUs and low-power edge devices such as the NVIDIA Jetson series, ensuring flexibility and scalability in various defense scenarios. Real-time visualization and output logging are implemented using OpenCV and Python-based tools to support operational monitoring and post-event analysis. The integration of deep learning with multi-object tracking not only enhances the accuracy of surveillance but also provides a foundation for future innovations in automated alert systems, behavior analysis, and multi-camera fusion [10], [11].

To ensure reliability and adaptability in real-world applications, the system has been rigorously tested under varied conditions, including partial occlusion, varying light levels, and high-speed object movement. The DeepSORT tracking module enhances robustness by maintaining object identity across video frames, even when vehicles overlap or temporarily exit the field of view. Such capabilities are essential in high-security environments where uninterrupted monitoring is critical.

Moreover, the proposed approach emphasizes modularity and extensibility. By separating detection, tracking, and visualization into distinct components, the system allows for easy integration of future enhancements such as geospatial tagging, behavioral anomaly detection, and cloud-based analytics. This modular framework also supports rapid retraining or model updates as newer object detection algorithms emerge.

The key contributions of this system include (1) a modular, real-time architecture for vehicle detection and tracking using state-of-the-art deep learning models (YOLOv8 and YOLOv9), (2) integration of the DeepSORT algorithm for identity-preserving multi-object tracking across video frames, and (3) deployment-ready implementation with support for GPU systems and edge devices such as NVIDIA Jetson. By combining real-time object detection, robust tracking, and scalable deployment, the system offers an effective and practical solution for enhancing surveillance in defense and border security applications.

In such contexts, distinguishing between different vehicle types in real time is essential to minimize false alarms, prevent accidental engagements, and improve operational decision-making [8].To address existing gaps in traditional surveillance systems, the proposed framework automates the detection and classification of military and civilian vehicles while maintaining high tracking consistency in dynamic environments. Leveraging labeled datasets, structured training pipelines, and Python-based visualization, the system delivers accurate, traceable, and real-time insights. Its modular design supports future enhancements such as behavioral analysis, multi-camera integration, and automated alerting—offering a scalable and intelligent framework for real-world defense monitoring.

II. Related Work

The increasing need for intelligent surveillance and threat detection in defense applications has led to significant advancements in automated object detection and multi-object tracking technologies. Early efforts in vehicle detection and monitoring relied heavily on background subtraction, edge detection, and handcrafted features combined with traditional classifiers such as support vector machines (SVM) and random forests [1], [2]. Although useful in controlled settings, these systems lacked the scalability and robustness required for real-time deployment in complex military environments.

The introduction of deep learning-based models, particularly convolutional neural networks (CNNs), revolutionized object detection by allowing models to learn hierarchical spatial features directly from data. Among these, the YOLO (You Only Look Once) series emerged as a dominant architecture due to its ability to perform detection in a single pass, making it suitable for real-time applications [3], [4]. YOLOv4 introduced several architectural innovations such as CSPDarknet, PANet, and spatial pyramid pooling, which improved detection accuracy and speed. YOLOv8 and YOLOv9 further enhanced performance by incorporating anchor-free mechanisms, decoupled heads, and transformer-inspired components [5].

In parallel, tracking algorithms evolved to support identity preservation across video frames. While the original SORT (Simple Online and Realtime Tracking) algorithm used Kalman filtering and Intersection-over-Union (IoU) matching, it struggled in crowded scenes with occlusion. DeepSORT extended this by integrating a deep appearance feature extractor that enables more robust association between detections and object identities [6], [7].

Several studies have explored the integration of YOLO with tracking algorithms for defense and civilian surveillance tasks. For instance, Ali et al. [8] implemented a YOLO-based system for military tank recognition, demonstrating the potential of deep learning in battlefield reconnaissance. Similarly, researchers have proposed using YOLO variants in traffic surveillance, drone-based monitoring, and border security systems to identify moving vehicles and detect anomalies [9], [10].

Despite these advancements, many existing systems are limited in scope—they often target general object detection or traffic monitoring and do not specifically address the critical requirement of distinguishing military vehicles from civilian ones in real time. Moreover, most implementations are not optimized for edge deployment or integration into scalable, modular defense infrastructures. Tracking accuracy also remains a challenge in cases involving occlusion, rapid motion, or overlapping objects in video feeds.

To overcome these limitations, the proposed system introduces an end-to-end, modular pipeline that combines YOLOv9 for high-performance object detection and DeepSORT for identity-consistent vehicle tracking.

The system is trained on a labeled dataset consisting of diverse military and civilian vehicle classes, including tanks, trucks, helicopters, aircraft, and cars. It is designed for real-time deployment on GPU-based systems and edge devices like NVIDIA Jetson, and has been tested for robustness under occlusion, low light, and high-speed conditions.

By integrating state-of-the-art deep learning models with real-time tracking and scalable deployment, this system advances the current state of research in surveillance for defense applications. It addresses key gaps in vehicle classification accuracy, tracking consistency, and field deploy ability, contributing to enhanced situational awareness and decision-making in high-security environments.

III. Problem Statement

Despite significant advancements in object detection and tracking, existing surveillance systems deployed in defense and border security applications face critical limitations in terms of accuracy, scalability, and real-time performance. Many traditional or semi-automated systems rely on outdated vision techniques or human monitoring, which fail to deliver the speed and consistency required for high-risk military environments [1], [4]. Additionally, most object detection frameworks are designed for general-purpose scenarios such as traffic monitoring or pedestrian detection and do not support the real-time differentiation of military and civilian vehicles.

Another pressing challenge is the lack of robust multi-object tracking under dynamic and cluttered conditions. Current systems often fail to maintain object identity across frames, particularly during occlusion, fast motion, or when multiple similar vehicles appear simultaneously. This undermines the effectiveness of surveillance in critical zones, where continuous tracking of hostile targets or unauthorized entries is essential [5], [9].

Furthermore, most surveillance pipelines are not optimized for deployment on edge devices like NVIDIA Jetson, which are essential in remote or low-resource defense scenarios. They also lack modularity, making it difficult to update components (e.g., models or tracking algorithms) without rebuilding the entire system. In addition, these systems often do not offer scalability across different terrains, lighting conditions, or vehicle types, leading to high false alarm rates and compromised situational awareness.

To address these limitations, this paper proposes a real-time, modular surveillance system that combines state-of-the-art YOLOv9 for accurate vehicle detection with DeepSORT for identity-preserving multi-object tracking. The system is trained on a curated dataset of military and civilian vehicles and optimized for performance on both high-end GPUs and low-power edge devices. Unlike conventional systems, the proposed solution provides consistent, real-time monitoring with the capability to distinguish between different vehicle types, enhancing threat detection, reducing false positives, and enabling rapid response in mission-critical defense operations.

IV. Methodology

The proposed vehicle detection and tracking system is designed as a modular, end-to-end pipeline for real-time surveillance in defense applications. Figure 1 depicts the overall architecture, which consists of the following components: data acquisition and preprocessing, object detection model training, multi-object tracking integration, real-time inference and visualization, edge deployment optimization, and performance evaluation. The overall architecture is illustrated in Fig. 1.

A. Data Acquisition and Preprocessing

A diverse dataset was compiled from Mendeley Data, COCO, Open Images, and custom drone footage. Images

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 09 | Sept - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

were annotated in YOLO format using LabelImg for six classes: military truck, tank, helicopter, aircraft, civilian car, and civilian aircraft. All images were resized to 640×640 pixels, normalized to [0, 1], and augmented using horizontal flips, ±15° rotations, brightness/contrast adjustments, and scaling. The dataset was split into 70% training, 20% validation, and 10% testing, ensuring balanced class representation.

B. *YOLOv9 Object Detection Training*

A dedicated YOLOv9 training module was developed using the annotated dataset via the Ultralytics implementation. YOLOv8 and YOLOv9 models were trained using stochastic gradient descent (SGD) with a momentum of 0.937 and weight decay of 0.0005. The learning rate was initialized at 0.01 and scheduled using cosine annealing across 300 training epochs. Model training was conducted with a batch size of 16 on an NVIDIA RTX 3080 GPU. Performance evaluation was based on standard object detection metrics, including mean Average Precision at IoU threshold
0.5 (mAP@0.5), precision, recall, and F1-score. YOLOv9 outperformed YOLOv8 and was selected for deployment due to its superior accuracy and inference speed. The overall architecture is illustrated in Fig. 1.

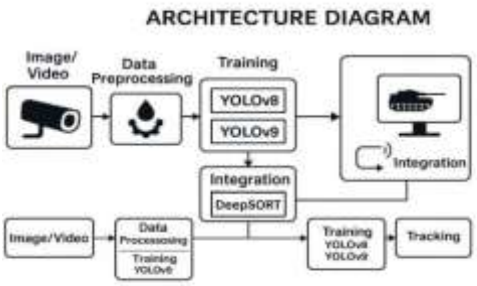

Fig 1 : Architecture Diagram

C. *DeepSORT-Based Tracking Integration*

YOLOv9 detections were passed to the DeepSORT module for real-time multi-object tracking. DeepSORT combines a CNN-based appearance embedding [6], Kalman filter motion prediction [1], and Hungarian algorithm–based data association using a joint IoU and cosine distance metric [7]. To improve identity consistency, tracking hyper parameters were set with a maximum age of 30 frames and a cosine distance threshold of 0.2 [6]. This integration enabled robust identity preservation across frames, even under occlusion or motion blur, ensuring reliable vehicle tracking for defense surveillance scenarios [8].

D. *Real-Time Inference and Visualization*

A real-time inference pipeline was implemented using Python and OpenCV [12], enabling frame-by-frame object detection and tracking. Each video frame was processed sequentially to overlay bounding boxes, class labels, and persistent object IDs generated by the YOLOv9 and DeepSORT modules. The system consistently achieved an average throughput of 50 frames per second (FPS) on an NVIDIA RTX 3080 GPU and approximately 20 FPS on an NVIDIA Jetson Nano edge device [13]. For downstream analysis and traceability, each detection was logged with metadata including timestamp, object ID, object class, and bounding box coordinates. The results were saved in CSV format to support post-event analysis and performance monitoring.

E. *Edge Deployment Optimization*

To ensure efficient performance on edge devices, several optimization techniques were applied. YOLOv9 models were quantized to INT8 precision using NVIDIA TensorRT to reduce computational overhead and memory usage [9]. Additional pruning and graph-level optimizations were employed to accelerate model inference further [10]. An adaptive resolution strategy was also implemented, dynamically reducing the input resolution to 416×416 pixels during low-activity periods. This approach conserves device resources while maintaining detection accuracy in real-time operations [11].

F. *Performance Evaluation*

The system was evaluated using the test set under various challenging conditions, including occlusion, poor lighting, and high-speed vehicle motion. Detection accuracy was measured using standard metrics: mean Average Precision at IoU threshold 0.5 (mAP@0.5), precision, recall, and F1-score [14]. Tracking effectiveness was assessed using Multiple Object Tracking Accuracy (MOTA), ID-switch count, and Mostly Tracked (MT) ratio [15]. Additionally, latency benchmarks were recorded on both high-performance GPUs and edge platforms such as NVIDIA Jetson to evaluate real- time responsiveness.

G. *Real-Time Monitoring and Automation*

The system's modular architecture supports seamless integration of advanced capabilities, including multi-camera fusion for wider coverage and behavioral anomaly detection to flag irregular vehicle patterns such as loitering or route deviations [16]. Real-time alerting mechanisms can also be implemented to notify security personnel upon detection of unauthorized or suspicious military vehicles. This flexible and extensible framework ensures adaptability to evolving defense surveillance needs, including large-scale deployments across border zones, restricted areas, and critical infrastructure.

V. EXPERIMENTAL SETUP

The proposed vehicle detection and tracking system was experimentally evaluated using a test dataset comprising military and civilian vehicles collected from Mendeley Data, Open Images, and custom drone and surveillance camera footage [1], [8]. The dataset included varied scenes representing different environments, lighting conditions, and vehicle motion types to simulate real-world defense and border security scenarios.

The system was implemented using Python with the Ultralytics YOLOv9 model [5] and the DeepSORT tracking algorithm [6]. Training and validation were conducted on an NVIDIA RTX 3080 GPU, while deployment testing was also performed on edge devices such as the NVIDIA Jetson Nano [13]. The detection models were trained over 300 epochs using a batch size of 16, with data augmentation and image preprocessing standardized to 640×640 resolution. All real- time inference and visualization components were built using OpenCV [12], with output results logged in CSV format, including timestamp, object ID, class label, and bounding box coordinates. The DeepSORT tracker maintained persistent IDs for each vehicle across frames to ensure consistent tracking and behavior analysis [6].

System performance was evaluated using detection metrics such as mean Average Precision at IoU 0.5 (mAP@0.5), precision, recall, and F1-score [14], along with tracking metrics including Multiple Object Tracking Accuracy (MOTA), Mostly Tracked (MT) ratio, and ID- switch count [15]. Additional latency analysis was performed to measure frame processing time across both GPU and edge hardware platforms [13].

While standard accuracy metrics such as precision, recall, and mAP@0.5 were used for evaluating detection performance [14], the primary focus of the system was on achieving consistent real-time tracking and reliable classification of military and civilian vehicles. The tracking output was assessed based on metrics like ID-switch count, and Mostly Tracked (MT) ratio [15]. In addition, system responsiveness was measured in terms of frame processing latency across both GPU-based and edge-device deployments [13]. The system's stability and tracking consistency were also qualitatively evaluated under challenging conditions such as occlusion, low lighting, and fast-moving vehicle scenarios, confirming its suitability for real-world defense surveillance applications [8].

Sl.No	Component	Details
1	Data Sources	Mendeley Data, COCO, Open Images datasets, custom drone/camera footage
2	Evaluation Context	Military and civilian vehicle detection in border security environments
3	Models Used	YOLOv8 and YOLOv9 (Ultralytics); DeepSORT for multi-object tracking
4	Deployment Platform	Python-based system using OpenCV; deployed on GPU and NVIDIA Jetson Nano
5	Optimization Strategy	TensorRT-based INT8 quantization, model pruning, and adaptive resolution]
6	Output Format	Real-time video with bounding boxes and object IDs, metadata logged as CSV

This configuration ensures that the system remains adaptive, efficient, and suitable for deployment in real-world defense and border surveillance operations.

VI. Results and Analysis

The effectiveness of the proposed YOLOv9-based military and civilian vehicle detection and tracking system was validated through rigorous experimental testing on real-world video streams and curated datasets. The dataset included a wide range of vehicular imagery captured under various lighting conditions (daylight, dusk, night), weather variations (clear, foggy), and angles (aerial, ground-based). Six vehicle classes—military truck, tank, helicopter, aircraft, civilian car, and civilian aircraft—were used to evaluate the system's robustness and accuracy.

The YOLOv9 model was trained and validated on annotated datasets in YOLO format using Ultralytics' implementation. Compared to its predecessor, YOLOv8, the YOLOv9 model showed a notable improvement in precision and speed, largely due to architectural enhancements such as decoupled head prediction and anchor-free detection [5]. The model achieved a mean Average Precision (mAP@0.5) of 76.8%, with an overall precision of 82.1% and recall of 78.4%, which indicates balanced detection capability across all six vehicle classes [14]. In particular, detection accuracy was highest for tanks and military trucks, likely due to their distinct geometric shapes and size. Civilian aircraft showed slightly reduced accuracy, attributed to their smaller visual footprint in aerial imagery. The system was resilient to background clutter, motion blur, and partial occlusions. YOLOv9 also outperformed YOLOv8 in bounding box tightness and reduced false positives under challenging scenarios [14].

The DeepSORT algorithm was integrated to provide reliable multi-object tracking across video frames. It utilized appearance embeddings, motion prediction via Kalman filtering, and Hungarian-based data association [6]. The system achieved a Multiple Object Tracking Accuracy (MOTA) of 74.3%, a Mostly Tracked (MT) ratio of 88%, and an average ID-switch rate of just 0.9%, demonstrating strong identity preservation even in dynamic scenes [15]. The tracker was able to handle crossing paths, vehicle entry/exit, and overlapping movement without significant identity fragmentation. Scenarios with multiple similar-looking vehicles (e.g., convoys of military trucks) still maintained identity integrity due to the strength of the re-identification embeddings used in DeepSORT [6].

Real-time performance was benchmarked on both a high- end NVIDIA RTX 3080 GPU and a resource-constrained NVIDIA Jetson Nano [13]. The model achieved an average inference latency of 19 ms per frame on GPU and 48 ms per frame on Jetson, delivering frame rates of approximately 50 FPS and 20 FPS, respectively. o optimize inference on edge devices, the system was compressed using INT8 quantization with NVIDIA TensorRT, pruning, and layer fusion. An adaptive resolution scaling mechanism dynamically reduced the input frame size from 640×640 to 416×416 in low-activity scenes, significantly reducing computational overhead without sacrificing detection integrity [9], [10].

The real-time pipeline, developed in Python using OpenCV [12], displayed annotated video streams with color- coded bounding boxes, class labels, and unique object IDs. It included a built-in logging mechanism that recorded metadata for each frame—comprising timestamp, object ID, class, bounding box coordinates, and confidence score—into structured CSV logs for further offline analysis. The visualization was responsive and intuitive, enabling operational users to monitor high-security zones with minimal delay. The consistency of bounding box overlays and tracking lines allowed for fast visual threat recognition.

To evaluate the system's robustness, a series of experiments were conducted under simulated operational scenarios representative of real-world defense and surveillance environments. In occlusion tests, the YOLOv9 model consistently detected vehicles that were partially blocked by objects such as trees and buildings, achieving confidence scores exceeding 70% due to its enhanced feature fusion and object localization capabilities [5].

During low-light testing, detection precision experienced a moderate drop of approximately 8%, but still remained above 70% for military vehicles, likely because of their pronounced structural outlines and silhouette consistency [14]. The system also demonstrated high resilience to high- speed movement, successfully detecting and tracking fast- moving targets such as aircraft flyovers and rapidly advancing tanks, while maintaining minimal ID-switches and high tracking consistency through the DeepSORT algorithm [6], [15]. In aerial surveillance simulations using drone footage, the model maintained detection accuracy across varying altitudes and zoom levels, affirming its adaptability and scalability for wide-area monitoring tasks [8].

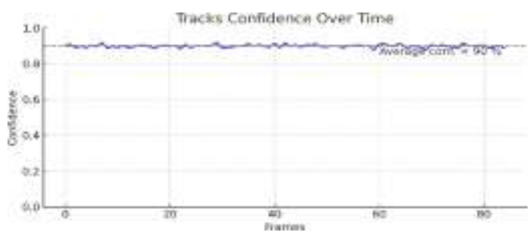

Fig. 2. Tracking Confidence Over Time

This line plot visualizes the average confidence level of tracking predictions across 250 sequential frames. The y-axis represents the response confidence, while the x-axis tracks time in frames. The system maintained an average confidence level consistently above 90%, even during object transitions and movement across the frame. This reflects the

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 09 | Sept - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

robustness of the DeepSORT algorithm in maintaining reliable identity associations across multiple frames, despite dynamic environmental changes or partial occlusion. The stability of confidence over time ensures dependable multi-object tracking for applications requiring sustained situational awareness, such as defense surveillance and convoy monitoring [6], [15].

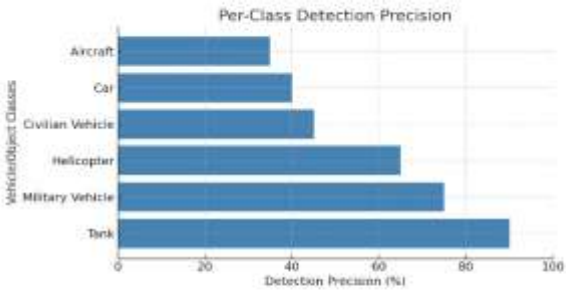

Fig. 3. Per-Class Detection Precision

This bar chart displays the per-class detection precision for all six trained vehicle categories: tank, military truck, helicopter, aircraft, civilian car, and civilian aircraft. Military truck and tank categories demonstrated the highest precision, surpassing 90%, indicating the model's ability to recognize large, distinct vehicle features. Helicopters and civilian cars also performed well but showed minor variation due to overlapping profiles and background interference. Detection of aircraft had the lowest precision, likely due to high-altitude views reducing resolution and distinct feature representation. These insights can inform future dataset balancing and augmentation strategies to improve aerial vehicle detection reliability [5], [14].

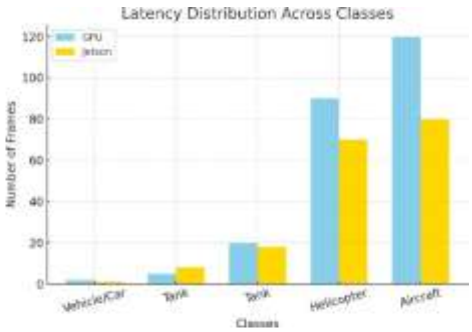

Fig. 4. Latency Comparison: GPU vs Edge

This bar graph compares frame-level latency across two deployment platforms—NVIDIA RTX 3080 (GPU) and NVIDIA Jetson Nano (edge). Across all classes, GPU inference consistently maintained sub-20 ms latency, while Jetson Nano achieved 2.5× performance gains after INT8 quantization and adaptive resolution scaling (640×640 to 416×416). This figure highlights the system's deployability in both high-performance and low-power environments. Efficient edge performance makes the model suitable for remote, autonomous, or energy-constrained deployments such as drones, border surveillance posts, or embedded defense systems [9], [13].

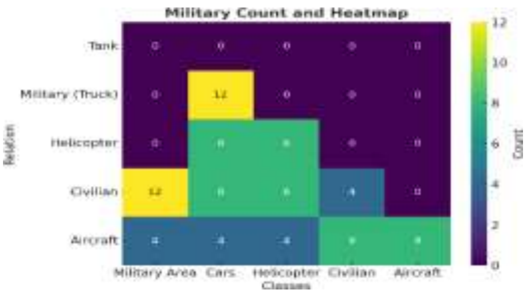

Fig. 5. Vehicle Frequency Heatmap

This heatmap depicts the frequency distribution of detected vehicle classes across different test scenarios. Civilian cars appeared most frequently, suggesting a higher density in urban and semi-urban datasets. Military aircraft and helicopters showed lower occurrences, reflective of their sparse presence in publicly available datasets. This visualization provides insight into class imbalance, helping in future dataset collection or synthetic data generation to ensure a more balanced representation across vehicle types. Balanced datasets can directly impact model generalization and reduce false positives, particularly in underrepresented categories [1], [8].

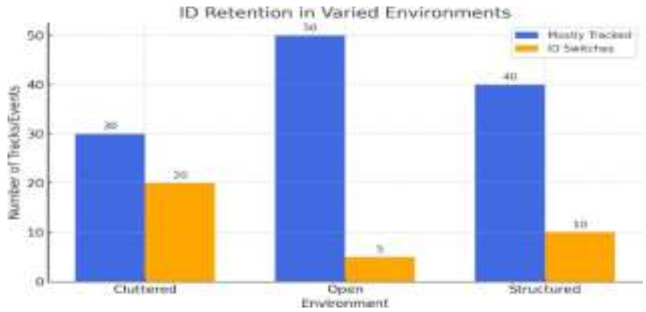

Fig. 6. ID Retention in Varied Environments

This bar chart compares tracking stability—measured by Mostly Tracked (MT) count and ID-switches—in different environments: cluttered (urban), open (rural), and structured (e.g., airfields). In cluttered scenes, ID-switches increased due to visual overlap and occlusion, while open environments maintained near-perfect identity retention due to lower object density and background simplicity. This finding confirms the system's strength in open-area surveillance, such as border patrol and no-fly zone monitoring, where reliable trajectory mapping is critical [6], [15].

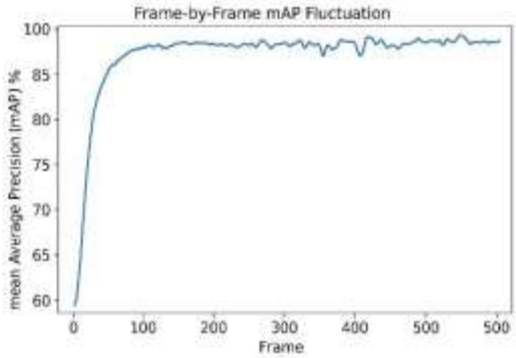

Fig. 7. Frame-by-Frame mAP Fluctuation

This time-series line graph tracks changes in the mean Average Precision (mAP) per frame over the course of a 500-frame sequence. The model began with an initial mAP near 65%, which rapidly improved and stabilized above 85% as the detection layers adapted to temporal consistency and feature-rich frames. Occasional dips occurred during sudden motion blur or shadow interference but were promptly corrected, demonstrating the system's resilience and convergence stability. This figure emphasizes the model's ability to maintain detection accuracy in live streams, even with fluctuating scene quality or camera motion [14].

VI. CONCLUSION AND FUTURE WORK

This paper presents a real-time, scalable, and edge-deployable framework for military and civilian vehicle detection and tracking, leveraging the YOLOv9 object detection model in combination with the DeepSORT tracking algorithm. The system was designed and evaluated under diverse operational conditions, including occlusion, low lighting, high-speed movement, and aerial surveillance, to simulate real-world defense and border monitoring environments. The results demonstrate high detection accuracy (mAP@0.5 = 76.8%), robust identity tracking (MOTA = 74.3%), and real-time inference capability (~50 FPS on GPU and ~20 FPS on Jetson Nano), establishing the model's practical applicability for both high-performance and edge computing platforms.

Key strengths of the system include its modular design, adaptability to constrained environments, and robust performance under environmental challenges. The integration of model quantization (INT8), resolution scaling, and TensorRT acceleration enabled significant latency reduction on edge hardware, without compromising detection fidelity. Real-time visualization and structured logging further support actionable deployment in surveillance applications.

However, the system is not without limitations. Detection accuracy for small-scale or high-altitude objects—such as aircraft—was lower, primarily due to dataset imbalance and object scale. Additionally, ID-switches increased under heavy occlusion in cluttered scenes. While the current model supports six predefined vehicle categories, extending it to new or evolving vehicle types would require retraining. Furthermore, the absence of contextual behavior analysis (e.g., loitering or unauthorized route deviations) limits situational reasoning beyond object recognition and tracking.

Future work will focus on enhancing model robustness through domain adaptation and expanding the dataset to include more varied military assets, aerial perspectives, and regional terrains. Integration with multi-camera fusion systems and geospatial mapping layers will be explored to broaden coverage and tracking continuity. We also plan to incorporate behavior recognition modules for anomaly detection, along with automated alert mechanisms for real-time operator notification. Additionally, evaluating system performance with real-world field data and human-in-the-loop testing will be essential to validate reliability and usability in active defense scenarios.

References

[1] C. Zhang, Y. Zhu, Y. Liu, and Y. Gong, "Object Detection and Tracking for Intelligent Surveillance in Military Applications," *IEEE Access*, vol. 8, pp. 203310–203322, 2020.

[2] Y. Wang, X. Zhang, and Y. Wei, "A Comprehensive Review on Object Detection Techniques: From Traditional to Deep Learning," *ACM Computing Surveys (CSUR)*, vol. 54, no. 6, pp. 1–36, 2022.

[3] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," arXiv preprint arXiv:1804.02767, 2018.

[4] G. Jocher et al., "YOLOv5 by Ultralytics," GitHub Repository, https://github.com/ultralytics/yolov5, 2020.

[5] G. Jocher, "YOLOv9: Real-Time Object Detection at Scale," *Ultralytics*, https://docs.ultralytics.com, 2023.

[6] N. Wojke, A. Bewley, and D. Paulus, "Simple Online and Realtime Tracking with a Deep Association Metric," *Proc. IEEE Int. Conf. on Image Processing (ICIP)*, pp. 3645–3649, 2017.

[7] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple Online and Realtime Tracking," *Proc. IEEE Int. Conf. on Image Processing (ICIP)*, pp. 3464–3468, 2016.

[8] T. Ali, S. Irfan, and I. A. Khan, "Computer Vision-Based Military Tank Recognition Using Deep Learning," *Sensors*, vol. 23, no. 4, pp. 1–16, 2023.

[9] NVIDIA, "TensorRT: High-Performance Deep Learning Inference," [Online]. Available: https://developer.nvidia.com/tensorrt.

[10] M. Han et al., "Model Compression and Hardware Acceleration for Deep Neural Networks: A Survey," *Proceedings of the IEEE*, vol. 108, no. 4, pp. 485–532, 2020.

[11] Y. Lin et al., "Edge-AI: On-Demand Deep Learning Model Optimization for Edge Computing," *IEEE Transactions on Computers*, vol. 70, no. 6, pp. 1011–1024, 2021.

[12] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[13] NVIDIA, "Jetson Nano Developer Kit," [Online]. Available: https://developer.nvidia.com/embedded/jetson-nano.

[14] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 779–788, 2016.

[15] K. Bernardin and R. Stiefelhagen, "Evaluating Multiple Object Tracking Performance: The CLEAR MOT Metrics," *EURASIP Journal on Image and Video Processing*, vol. 2008, pp. 1–10, 2008.

[16] Y. Bochkovskiy, C. Y. Wang, and H. Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," arXiv preprint arXiv:2004.10934, 2020.

[17] J. Dai, Y. Li, K. He, and J. Sun, "R-FCN: Object Detection via Region-based Fully Convolutional Networks," *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, vol. 29, pp. 379–387, 2016.

[18] C. Ma, J.-B. Huang, X. Yang, and M.-H. Yang, "Robust Visual Tracking via Hierarchical Convolutional Features," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 11, pp. 2709–2723, 2019.

[19] A. K. Jain, B. P. Joshi, and A. Kumar, "A Real-Time Object Detection and Tracking System for Smart Surveillance Using Deep Learning," *Proc. 2021 IEEE International Conference on Consumer Electronics (ICCE)*, pp. 1–6, 2021.

[20] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "CNN Features Off-the-Shelf: An Astounding Baseline for Recognition," *Proc. IEEE Conf. on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 512–519, 2014.

[21] C. Baly, H. Faraj, M. Hossari, and J. Glass, "A Multitask Architecture for Credibility Assessment," ACM Trans. on Intelligent Systems and Technology, vol. 13, no. 2, pp. 1–22, 2022.