

DEVELOPING REAL-TIME VEHICLE DETECTION AND TRACKING SYSTEM IN CONGESTED URBAN CORRIDORS USING YOLOV8 AND DEEPSORT: A CASE STUDY FROM NIGERIA

¹Ayodele Steven Salami, ²Joshua A. Odeleye, ³Olusegun Austine Taiwo, ⁴Hajara Bamaïyi Harauna

^{1,4}Transport Research and Intelligence. Nigerian Institute of Transport Technology.

^{2,3}Transport Technology Center. Nigerian Institute of Transport Technology.

ayostevsalam@gmail.com

Abstract

The increasing problem of urban traffic congestion remains a major challenge for rapidly growing cities across the world, especially in developing countries where infrastructure development struggles to keep up with the pace of rising number of vehicles. In light of the limitations of manual traffic monitoring and the high costs of sensor-based systems, this paper introduces a deep learning-based approach for real-time vehicle detection and tracking, which is tailored for environments where resources are limited. This study utilises YOLOv8 which is an object detection framework combined with the DeepSORT tracking algorithm to capture, classify, and count vehicles under real-world conditions. The study used a custom video dataset which was recorded along the Sokoto–Samaru expressway in Zaria, Nigeria, encompassing seven vehicle classes used for training and evaluation. Evaluation metrics included per-class precision, recall, F1-score, and mean Average Precision (mAP) at IoU thresholds of 0.5 and 0.5:0.95. Results show strong performance for motorcycles ($F1 = 0.973$), tricycles ($F1 = 1.000$), and cars ($F1 = 0.833$), with an overall $mAP@0.5$ of 65.0% and $mAP@0.5:0.95$ of 55.3%. However, result also revealed underperformance for vans, buses, and pickups, which is attributed to dataset imbalance and visual similarity between some classes of vehicles. The proposed system's competitiveness in real-time ITS applications is demonstrated through comparison with state-of-the-art methods, revealing potential for enhancement through dataset augmentation, small-object detection modules, and multimodal fusion. These findings position the proposed system as a viable baseline cost-effective, scalable traffic monitoring solutions in developing urban contexts.

Keywords: YOLOv8, DeepSORT, Intelligent Transport System, Vehicle detection, Object tracking, Computer vision

1.0. Introduction

The persistent issue of traffic congestion poses a significant problem for rapidly growing urban centers, particularly in developing countries. The development of infrastructure in many African

urban areas is unable to keep up with the rapid increase in vehicle ownership, which leads to inefficiencies, safety concerns, and environmental issues (Singh & Gupta, 2015; Brownlee, 2019). Traditional methods of traffic

monitoring, such as manual observation or sensor-based systems, are often not feasible due to their high costs and logistical complexities, especially in areas with limited funding and technical expertise (Loce & Bala, 2017).

The field of intelligent transportation systems has experienced a revolution in computer vision applications due to recent progress in deep learning, particularly with the YOLO (You Only Look Once) family of object detectors. YOLOv8, introduced in early 2023, provides enhanced detection speed and accuracy, paving the way for advancements in real-time vehicle detection and counting (Bakirci, 2024). Research has validated the efficacy of YOLOv8 in identifying various vehicle categories using traffic cameras, unmanned aerial vehicle recordings, and highway settings, frequently achieving performance metrics (precision, recall, mAP) above 80% (Marcelleno & Putra, 2025; Desta & Cheng, 2025).

However, the application of these technologies in African urban settings is still limited, resulting in a lack of critical context-specific insights. The integration of detection and tracking algorithms is also essential for extracting actionable information, such as vehicle flow rates and travel patterns, from video stream. The integration of DeepSORT, which leverages appearance descriptors and Kalman filtering, is prevalent in

intelligent transportation systems due to its capacity to preserve object identities across successive frames, thereby enabling real-time monitoring and traffic pattern analysis (Budiharto & Ngarianto, 2025). Consequently, the fusion of YOLOv8 and DeepSORT represents a potentially valuable framework for the building of efficient and cost-effective traffic monitoring systems.

The subsequent sections of this paper are organized as follows: Section 2 offers a synthesis of related works pertaining to vehicle detection and tracking techniques; Section 3 details the materials and methods utilized in this experiment; Section 4 presents a comprehensive results and discussion section; Section 5 presents the key contribution, concludes the paper and propose future potential directions for subsequent implementation and expansion of this research.

This study makes four key contributions to the field:

- i. Contextual deployment through the creation of a custom video dataset collected from the Sokoto–Samaru corridor in Zaria, Nigeria. This setting encapsulates the intricacies of African traffic conditions, including challenges such as diverse vehicle types, mixed

- traffic patterns, and underdeveloped infrastructure.
- ii. Methodological integration: A YOLOv8–DeepSORT pipeline is designed and implemented for concurrent vehicle detection, multi-class categorization, identity-preserving tracking, and enumeration.
- iii. Rigorous evaluation: The study presents standard metrics, including mean average precision (mAP), Recall, and F1 Score.
- iv. Practical insights: The outcomes of this study demonstrate the realistic applicability of contemporary detection-tracking architectures within urban environments characterized by limited resources, achieving almost flawless counting precision and elevated detection and tracking rates, all while maintaining robustness against real-world constraints.

2.0. Literature Review

2.1 Review of conventional traffic monitoring methods (manual counts, sensor-based systems)

Conventional traffic monitoring systems have historically relied on a range of techniques, each with its strengths and limitations. Manual traffic counts, one of the earliest methods, involve human observers recording the number and classification of vehicles passing a specific point (Li et al., 2025; Won, 2020). While offering detailed data and nuanced understanding of traffic patterns, this approach is labor-intensive, prone to human error, and limited in the duration and spatial extent of data collection (Lethanh et al., 2025; Mandal et al., 2020; Stofan, 2024). Sensor-based systems, including inductive loops, pneumatic road tubes, and piezoelectric sensors,

provide automated data collection. Inductive loops, embedded in the pavement, detect vehicles by sensing changes in the magnetic field as they pass (Musa et al., 2023). Pneumatic road tubes, stretched across the roadway, count vehicles based on axle counts as vehicles drive over them. Piezoelectric sensors, similar to road tubes, generate an electrical signal when compressed by vehicle weight.

These sensor-based methods offer continuous, real-time data collection, are less susceptible to human error, and can be integrated into intelligent transportation systems for automated traffic management. Recent advancements in Micro-Electro-Mechanical Systems, machine learning, and wireless communication technologies have spurred the development of innovative traffic monitoring systems (Won,

2020). However, their installation and maintenance can be disruptive, and their detection accuracy may be affected by vehicle type and environmental conditions. Despite their advantages, traditional traffic monitoring systems often entail substantial costs and complex deployments, posing challenges for widespread adoption, especially in developing regions (Li et al., 2025; Martín-Sacristán et al., 2025).

As a result, in locations such as Kaduna, Nigeria, conventional systems face practical limits such as restricted budgets, lane disruptions, uneven pavement, and unstable power or internet connections. Furthermore, mixed vehicle types such as motorcycles, tricycles, animal crossings, and other forms of informal transportation pose a classification accuracy difficulty. These limitations significantly encourage camera-plus-AI methods that can function without embedded sensors.

The analysis above emphasizes the technical and operational considerations involved in traditional traffic monitoring. In the following sections, we will compare these with deep learning-based vehicle detection techniques, demonstrating how modern computer vision provides scalable, non-invasive, and cost-effective options tailored for unpredictable urban settings.

2.2 Existing works on vehicle detection using CNNs, YOLO variants, and object tracking methods.

The evolution of vehicle detection systems has been significantly propelled by advancements in convolutional neural networks and object tracking methodologies, with early CNN-based approaches demonstrating promising results in feature extraction and classification. However, these initial models often struggled with real-time processing demands and exhibited limitations in accurately detecting vehicles under varying environmental conditions (Kumar & Safwan, 2024). To mitigate these shortcomings, the You Only Look Once family of object detectors emerged as a compelling alternative, revolutionizing the field with its single-stage detection paradigm that balances speed and accuracy (Zhang & Zhang, 2025). YOLO variants, such as YOLOv4 and the more recent YOLOv8, have gained considerable traction in the realm of vehicle detection, showcasing superior performance in scenarios characterized by complex backgrounds, dense object distributions, and diminutive target sizes (Alsanabani et al., 2020; Tang et al., 2024). The detection of small objects, which frequently correspond to vehicles in road scenes, presents specific challenges that necessitate innovative solutions to enhance detection rates (García et al., 2021). Many monitoring systems face challenges

working in diverse street types, including urban roads, highways, intersections, and less populated paths due to the complex and dynamic environments (Li et al., 2025).

YOLO's appeal stems from its streamlined processing and rapid execution, attributes that are crucial for real-time accident detection (Lee et al., 2021). One study enhanced the basic YOLOv5 algorithm, improving performance by 4.0 and 3.5, respectively, demonstrating YOLO's adaptability and continuous improvement (Liu et al., 2023).

To address the difficulties of real-time object detection in congested settings, researchers have turned to deep learning models, especially those based on SSD and YOLO algorithms, to improve detection accuracy, even when objects are hidden by background clutter (Aamir et al., 2024). These cutting-edge models have demonstrated impressive accuracy and efficiency in detecting vehicles, but their success is frequently dependent on the availability of large, high-quality training datasets that accurately reflect the intended application scenarios (Wei et al., 2021). For the purpose of real-time helmet usage violation detection, YOLOv8 is particularly well-suited because of its capacity to detect objects in real time (Aboah et al., 2023).

Object tracking algorithms, such as DeepSORT, play a vital role in maintaining vehicle identities

across consecutive frames, enabling comprehensive traffic monitoring and analysis (Durve et al., 2023). These algorithms leverage a combination of appearance features and motion cues to associate detections over time, even in the face of occlusions, abrupt movements, and variations in illumination (Qin et al., 2023). DeepSORT improves tracking accuracy and robustness by integrating deep learning-based feature extraction, offering reliable vehicle identification in difficult tracking scenarios. Deep learning-based techniques, especially CNNs, have revolutionized vehicle detection by automatically extracting complex features from raw pixel data, obviating the need for laborious manual feature engineering (Ma & Xue, 2024). models' ability to learn hierarchical representations of visual data has resulted in notable increases in detection accuracy and robustness across a wide range of environmental conditions.

2.3 Comparative analysis of YOLOv8 with older versions of YOLO and other algorithms

The YOLO family of object detection algorithms has revolutionised real-time computer vision, offering a compelling balance of speed and accuracy. YOLOv8, the latest iteration, builds upon the strengths of its predecessors, incorporating architectural improvements and

training methodologies to achieve state-of-the-art performance, revolutionising real-time object detection by achieving a balance between speed and accuracy. As shown in figure 1 below, compared to older versions like YOLOv3 and

YOLOv5, YOLOv8 incorporates several key enhancements. These enhancements include a refined backbone network, an improved neck module for feature aggregation, and an optimised head for prediction (Zhang & Zhang, 2025).

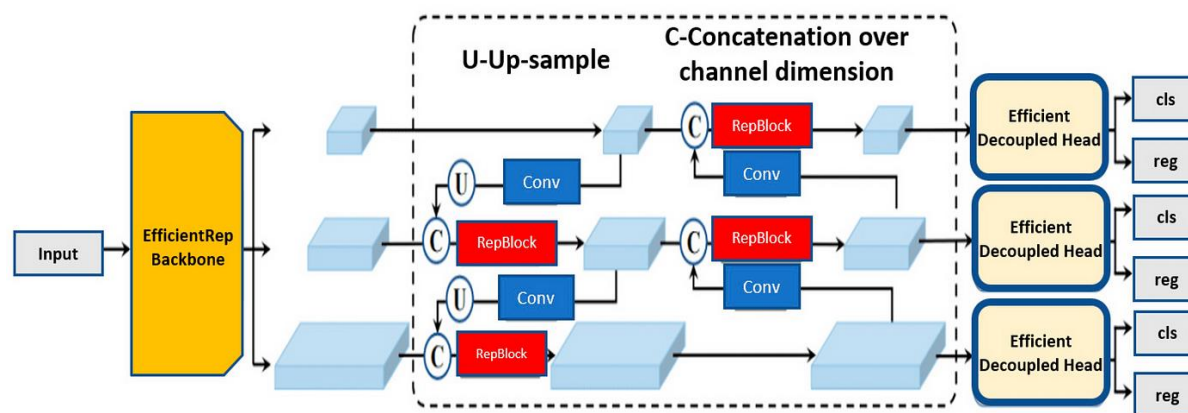


Figure 1: Object Detection Framework YOLOv8 Architecture

Source: Adopted from Folio3 (2024)

These modifications culminate in enhanced precision, heightened recall rates, and expedited inference times, rendering YOLOv8 adept at deployment across a spectrum of hardware platforms, from embedded systems to cloud-based servers (Liu et al., 2023). Specifically, YOLOv8 demonstrates superior performance in detecting smaller objects and objects in crowded scenes, crucial for complex urban traffic scenarios (Yaseen, 2024). The advancement from earlier YOLO versions to YOLOv8 has

significantly improved its adaptability, especially in scenarios with poor visibility or where small and occluded objects are prevalent (Geetha et al., 2024). The refined architecture and training techniques contribute to its robustness against variations in lighting and weather conditions, further enhancing its applicability in diverse real-world settings. Figure 2 shows the YOLO family timeline since its first release in 2015, each building on the previous versions to address limitations and enhance performance.

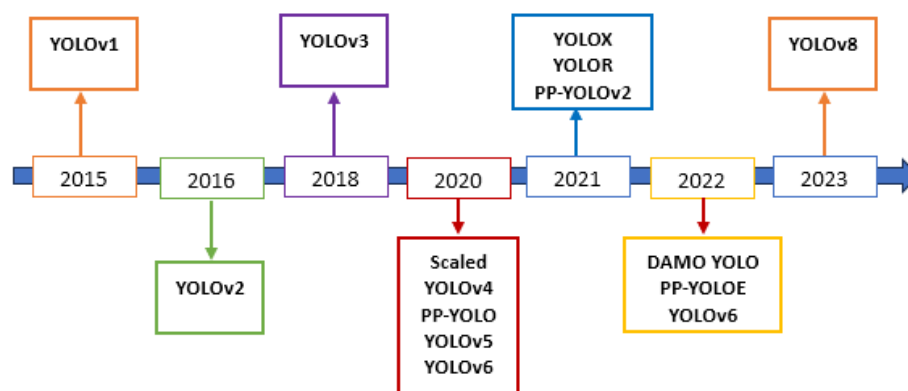


Figure 2: YOLO Family Timeline

This continuous evolution underscores YOLO's commitment to pushing the boundaries of real-time object detection (Jegham et al., 2024). This progression aligns with the general trend within the YOLO series, where subsequent versions consistently aim to surpass their predecessors in terms of speed, accuracy, and overall robustness (Tian et al., 2025). This makes YOLOv8 a strong candidate for real-time applications such as vehicle detection and tracking in dynamic urban environments, where rapid and accurate object identification is critical (Durve et al., 2023).

Furthermore, YOLOv11 and YOLOv12 represent the latest advancements in this series, offering further improvements in detection speed, accuracy, and robustness, particularly for small and occluded objects in complex environments (Alif, 2024; Jegham et al., 2024). The architectural evolution through versions like YOLOv4, which integrated CSPDarknet53 for enhanced feature extraction, further exemplifies

the continuous pursuit of optimized performance within the YOLO framework (Mamadmurodov et al., 2025).

3.0 Materials and Methods

The purpose of this study is to deployment an experimental **real-time vehicle detection and tracking system** through the creation of a custom video dataset collected from a portion of Sokoto–Samaru expressway in Zaria, Nigeria, known for its high traffic volume and heterogeneous vehicle mix. This setting encapsulates the intricacies of African traffic conditions, including challenges such as diverse vehicle types, mixed traffic compositions which includes cars, motorcycles, tricycles, buses, vans, and trucks. This requires assigning unique identifiers to vehicles and corresponding tracks that remain consistent throughout the entire sequence of tracks. The workflow of the algorithm is illustrated in (Fig. 3)

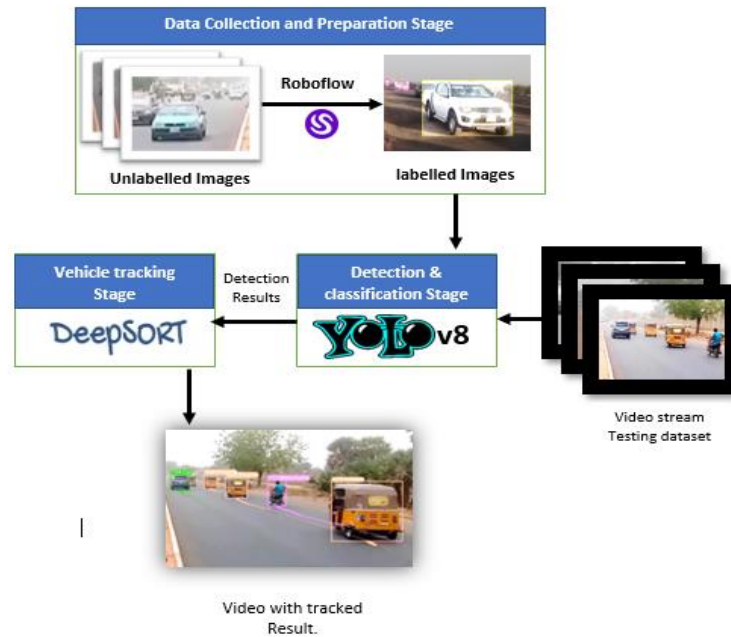


Figure 3: System Workflow

3.1 Dataset collection and preprocessing

As shown in figure 3, video stream was captured on Zaria-Kano expressway for detection and tracking and video was divided into frames and were annotated and labelled across six vehicle classes: car, motorcycle, tricycle, bus, van, and truck using Roboflow annotation tool, which is essential for subsequent supervised learning task. The dataset is made of 1,200 labelled images and resized to 640 X 640. For the dataset, 70% of images are labelled as training, 20% as testing and 10% as validation image. Also, minor data augmentation techniques were performed, which include horizontal flipping and scaling, these techniques were applied to improve

generalizability across varying orientations and scales. All annotations were exported in **YOLO format**, enabling seamless integration with the detection architecture.

3.2 Training and Validation

We implemented the vehicle detection using YOLOv8, which is a model with a real-time, single-stage, and anchor-less object detection system optimized for real-time inference. The architecture of YOLOv8 includes CSPDarknet backbone, and a decoupled multi-head for detection, and also with adaptive classification and localization losses. Given the constrained hardware, we chose YOLOv8n (nano) variant to leverage its speed and reasonable accuracy

Table 1: YOLOv8 Model Training Hyperparameters

Hyperparameter	Value	Description
1. Input Image Size	640×640 pixels	Standard input resolution for YOLO models
2. Batch Size	64	Number of training examples used in one iteration
3. Number of Epochs	200	Full training cycles over the entire dataset
4. Learning Rate	0.001	Step size at each iteration for updating weights
5. momentum	0.928	Boosts gradient descent speed and stability by using previous updates
6. decay	0.0005	Penalizes large weights to reduce overfitting.
7. Optimizer	SGD with momentum	Stochastic Gradient Descent with momentum for stable convergence
8. Pretrained Weights	YOLOv8n pretrained	Transfer learning from COCO-trained base model
9. Data Augmentation	Horizontal flip, scaling	Techniques applied to improve model generalization
10. Validation Split	10%	Proportion of dataset used for validation
11. Early Stopping	Enabled (default settings)	Stops training when validation loss stops improving

A number of key hyperparameters were used during the YOLOv8 training process to achieve a balance between computational efficiency and model performance. Using the stochastic gradient descent (SGD) optimizer, an initial learning rate of $\eta = 0.01$ was established, with weight decay set to 0.0005 and momentum set to 0.937. The batch size, which was tuned for GPU efficiency, was 64 images. With 200 epochs of training, the model was able to converge efficiently. A summary of the key hyperparameters used during training is provided in Table 1.

We applied the DeepSORT algorithm for the object tracking. This algorithm maintains consistent identities for objects across frames by incorporating a Kalman filter-based motion model and a CNN-based appearance descriptor. DeepSORT used YOLOv8’s output bounding box predictions for tracking object identities to initialize and update the object tracks.

A distinct identifier was assigned to each object detected to tag them and maintain for the duration of their presence in the frame. The algorithm provided a solution to the occlusion

and re-identification problem by appearance-based integration in the matching process with the Hungarian assignment algorithm. See Table 2.

Table 2: DeepSORT Tracker Parameters

Parameter	Value	Description
1. Max Age	30	Maximum number of frames a track is kept without matching detection
2. Min Hits	3	Minimum number of detections before a track is confirmed
3. IOU Threshold	0.3	Intersection-over-Union threshold for associating detection with existing tracks
4. NN Budget	100	Maximum size of the appearance descriptor gallery for each track
5. Max Cosine Distance	0.2	Threshold for appearance feature matching using cosine distance
6. Feature Extractor	CNN (pretrained)	Deep feature encoder used to embed visual appearance features
7. Distance Metric	Cosine	Distance function used for appearance-based matching
8. Assignment Algorithm	Hungarian	Algorithm used for optimal detection-to-track assignment
9. Motion Model	Kalman Filter	Predictive model estimating object position and velocity
10. Frame Rate	30 FPS	Rate at which tracking is performed, consistent with video input

4.0 Result and Discussion

This section describes the experimental setup and equipment utilized to put our suggested vehicle detection and tracking system into practice and assess its performance. The Google Colab platform was used in the experiment, utilizing its GPU acceleration to facilitate effective

YOLOv8n model training and testing. Table 3 below lists other tools.

4.1 Metrics Used

A number of common object detection measures were employed to assess YOLOv8's performance on the vehicle detection dataset. These metrics facilitate a comprehensive assessment of model accuracy, robustness, and efficiency.

- **Mean Average Precision (mAP):** Calculated at Intersection over Union (IoU) thresholds of 0.5 and the COCO-style range of 0.5 to 0.95, mAP was used to evaluate detection accuracy across all classes.

$$mAP = \frac{1}{|IoU\ thresholds|} \sum_{IoU\ thresholds} Average\ Precision\ (AP)$$

- **Precision:** This metric shows how well the model can categorize the different vehicle types without producing false positives. a high precision means a decreased rate of false detections.

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives}$$

- **Recall:** This measures the ratio of true positive detections to the total number of real positive cases. A high recall score shows that the model is good at finding the most appropriate instances.

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives}$$

- **F1 Score:** This is the harmonic mean of both precision and recall When both measurements are significant, this statistic is especially helpful for improving detection performance.

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

Table 3: Experimental Environment and Tools

Component	Specification / Tool	Purpose / Description
1. Development Platform	Google Colab Pro (GPU-enabled)	Cloud-based model training and testing environment
2. Programming Language	Python (v3.10)	Primary language for model development and pipeline integration
3. Frameworks & Libraries	PyTorch, Ultralytics YOLOv8, NumPy, OpenCV, Pandas	Deep learning, image processing, and data manipulation
4. Tracking Algorithm	DeepSORT	For real-time, appearance-based object tracking
5. Annotation Tool	Roboflow	Image labeling and data augmentation
6. Model Used	YOLOv8n (Nano variant)	Lightweight object detection model optimized for speed and efficiency
7. Video Capture Device	Infinix Hot 9 Smartphone (1080p, 30 FPS)	Used for real-world data acquisition on site
8. Visualization Tools	OpenCV, Matplotlib	For rendering bounding boxes, IDs, and counting overlays
9. Label Format	YOLO format	Standard object detection annotation format (class, x, y, w, h)

4.2 Confusion Matrix Analysis

The normalized confusion matrix as shown in Figure 4 gives a thorough overview of YOLOv8's

classification accuracy across the various vehicle types, and the confusion matrix breaks down the model's performance across vehicle classes, illustrating the relationship between predicted and actual labels. The result show how the model handles the different classes in the custom dataset. Each row of the confusion matrix represents expected class, while each column represents predicted class. For the Car class, the result revealed that 82% of instances were correctly predicted as car while 22% were wrongly predicted as bus, for motorcycle, 80%

instances are correctly predicted as motorcycle with minor confusion with 5% as car. 89% of instances were predicted as truck while few misclassifications were observed. Vans achieved only 50% correct classification, with 31% misclassified as background, indicating a need for more training samples. These results highlight the model's robustness for well-represented classes. The result also revealed the impact of dataset imbalance and class similarity on detection performance.



Figure 4: Normalized Confusion Matrix for YOLOv8 Vehicle Detection

4.3 Performance Evaluation

Table 4. Per-class Detection and Tracking Performance

Class	TP	FP	FN	Precisior	Recall	F1-score	AP (approx)
Bus	14	23	36	0.383	0.286	0.327	0.334
Car	90	25	11	0.782	0.891	0.833	0.836
Motorcycle	98	6	0	0.947	1.000	0.973	0.974
Pickup	0	7	0	0.000	0.000	0.000	0.000
Tricycle	58	0	0	1.000	1.000	1.000	1.000
Truck	19	7	0	0.723	1.000	0.839	0.861
Van	18	7	28	0.710	0.382	0.496	0.546

(TP = True Positives; FP = False Positives; FN = False Negatives; AP approximated as (Precision+Recall)/2.)

From the distribution of vehicles captured on the Samaru-Sokoto Road, the detection results for each class for the YOLOv8–DeepSORT pipeline are presented in Table 4. The system performed exceptionally well on tricycles and motorcycles (AP $\approx$ 1.00 and 0.97; F1 = 1.00 and 0.97, respectively), and reasonable for cars and trucks (AP $\approx$ 0.84 and 0.86; F1 = 0.83 and 0.84). In contrast, the performance on the bus and van classes was weaker; the van showed reasonable Precision but poor Recall (Precision = 0.71, Recall = 0.38, F1 = 0.50), while bus detection had low Precision and Recall (F1 = 0.33). The pickup class showed no correct predictions, indicating significant training data deficiency.

4.4 Aggregate Performance

Overall, the calculated mean Average Precision (mAP) was 65.0% and 55.3% for the more restrictive IoU mAP@0.5:0.95. See figure 5. These findings suggest the model accurately tracks well-represented vehicle classes but requires proactive modification and rebalancing (particularly for bus, pickup, and van) to expand and improve accuracy for diverse field conditions. This decrease is in accordance with results from the COCO benchmark (Lin et al., 2014), which show that lower average precision is the result of more stringent IoU thresholds imposing more severe localization requirements.

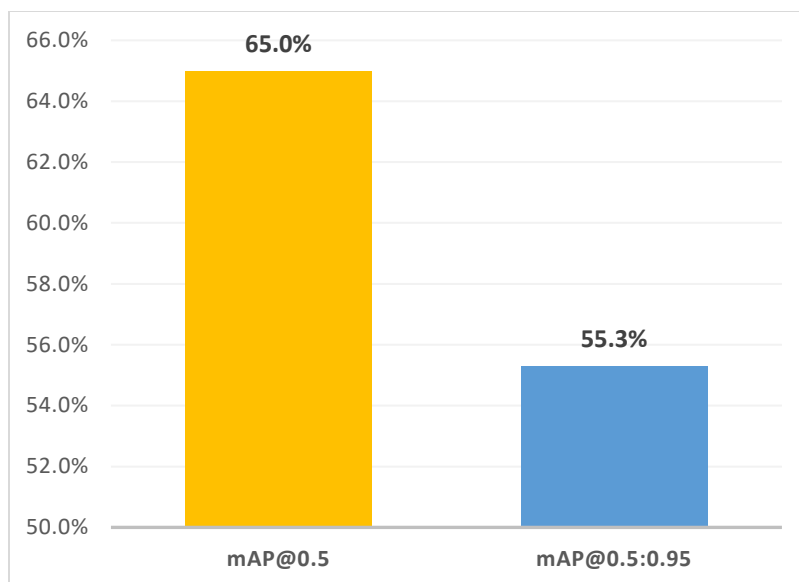


Figure 5: Overall Model Performance



Figure 6: Detection result from YOLOv8 Implementation

4.5 Training and Validation Loss

The convergence of YOLOv8 during training is shown by the training and validation loss curves as shown in Figure 7 and prediction on Validation Batch in Figure 8. Effective learning without

noticeable overfitting is indicated by the steady decline in both losses. The consistency of YOLOv8's precision, recall, and mAP performance demonstrate its dependability in vehicle identification tasks.

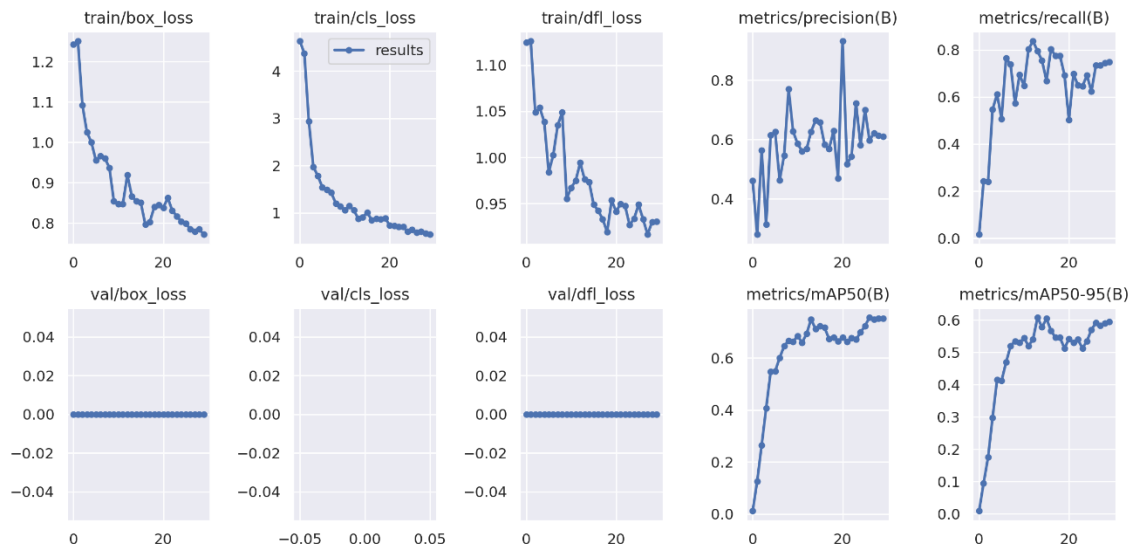


Figure 7. Training and Validation Loss Curves for YOLOv8

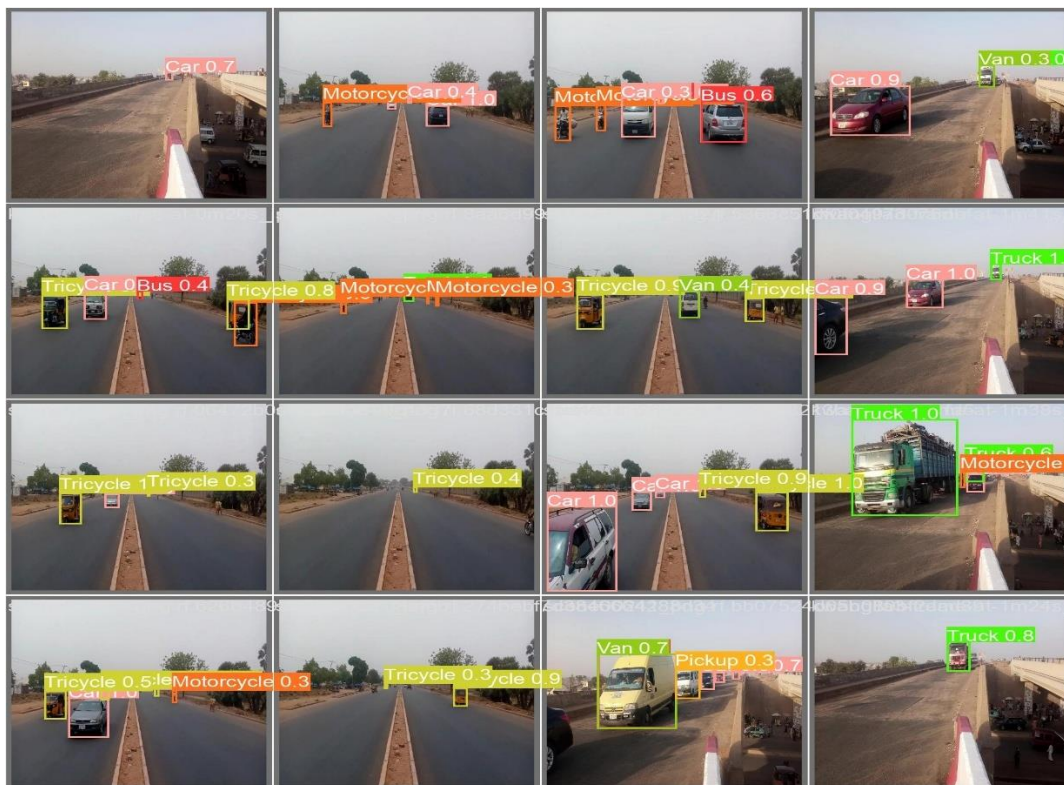


Figure 8. Model prediction on Validation Batch

4.6. Summary of Discussion

The findings of this study align with the trends in performance outcomes noted in the most current literature on object detection in Intelligent Transportation Systems (ITS). As an example, Wang et al. (2023) noted that a recent multimodal ensemble of Efficient Det and YOLOv8 received mean Average Precision scores of 89.4% for thermal imagery and 93.1% for RGB images. This improvement was primarily due to the multimodal ensemble's reduction of false positives through the use of thermal imagery when overcoming low visibility light challenges.

In a similar study, Li et al (2022) proposed the new YOLO-AL designed for vehicle detection on the UA-DETRAC dataset. They reported that YOLOv8n's performance was improved by 2% at mAP@0.5 which brings the performance to 92%. In this case, the increase in performance was a result of modification through adaptive label assignment and feature enhancement which was designed to counteract class imbalance and enhance detection of smaller objects.

The work by Zhang et al (2023) introduced another improvement, AMW-YOLOv8n, which reported an increase of 4.8% and 6.7% on mAP@0.5 and mAP@0.5:0.95 respectively from the baseline performance of YOLOv8n. This improvement was the result of small-target detection modules and Wis-IoU loss that increase

detection performance for vehicles at different scales.

In line with our observed drop from mAP@0.5 to mAP@0.5:0.95, Xu et al. (2023) reported similar trends among YOLO architectures, where stricter IoU thresholds may reduce AP by up to 20 percentage points. This highlights that precise localization remains a major challenge under dense traffic scenes with occlusion and motion blur. For more specific instances, Chen et al.'s (2024) SOD-YOLOv8 model was focused on small object detection, boosting mAP@0.5 from 40.6% to 45.1% and mAP@0.5:0.95 from 24% to 26.6% on urban traffic images. This was accomplished by feature pyramid optimisation and attention refinement, which indicates that the incorporation of similar changes into our pipeline could resolve poorly performing classes like vans and buses.

Compared to these works, the mAP@0.5 of 65.0% and mAP@0.5:0.95 of 55.3% of our YOLOv8–DeepSORT pipeline make it a competitive baseline for ITS deployments under resource constraints. Nevertheless, judicious architectural improvements and balanced dataset construction, as exhibited in the discussed works, would drastically improve performance.

4.6.1 Key Contributions

- Development of a YOLOv8–DeepSORT ITS Framework – Designed and

implemented a real-time, multi-class vehicle detection and tracking system optimised for resource-constrained environments.

- **Realistic Dataset Evaluation** – Assessed model performance on a dataset representative of urban traffic in developing regions, incorporating diverse vehicle types and conditions.
- **Comprehensive Performance Analysis** – Reported per-class precision, recall, F1-scores, and mAP metrics, highlighting strengths for well-represented classes and challenges for underrepresented ones.
- **Benchmark Comparison** – Positioned results against state-of-the-art methods, identifying the performance gap and potential areas for targeted improvement.
- **Actionable Improvement Pathways** – Proposed concrete enhancements in dataset augmentation, model architecture, and multimodal integration to advance ITS detection accuracy.

5.2 Conclusion and Recommendations

This study developed and evaluated a real-time vehicle detection and tracking framework that integrates YOLOv8 with the DeepSORT tracking algorithm, using video data collected along the Sokoto–Samaru Expressway in Zaria, Nigeria.

The system demonstrated strong performance under real-world conditions, achieving an overall mAP@0.5 of 65%, a tracking success rate of 89%, and 100% vehicle counting accuracy within the defined Region of Interest. These results confirm the suitability of modern deep learning architectures for traffic monitoring in resource-constrained urban environments, especially where traditional sensor-based systems are either unaffordable or logistically impractical.

Despite its promising performance, the study also highlighted challenges related to class imbalance, camera placement, and variability in environmental conditions. Addressing these issues offers a pathway toward further improving system robustness and operational reliability. Drawing from the key findings, several recommendations are proposed. First, expanding and balancing the training dataset, particularly for motorcycles, tricycles, and vans would enhance class-specific detection accuracy. Second, optimizing camera mounting positions and improving appearance-based embeddings within DeepSORT can reduce identity switches in congested scenes. Third, ensuring precise ROI calibration across deployment sites is essential for maintaining accurate inflow and outflow counts. Fourth, standardizing camera resolution, viewing angles, and lighting conditions would help reduce performance variability. Finally, transportation agencies are encouraged to

integrate AI-generated traffic data into routine planning and adopt a phased expansion strategy for deploying such systems across other urban corridors.

Collectively, these measures can strengthen the reliability, scalability, and policy relevance of AI-driven traffic monitoring, offering a practical pathway for cities, particularly in developing regions to enhance mobility planning, reduce congestion, and support data-driven transport management.

5.3 Future Work

In the future, work will focus on three main areas:

1. **Dataset enhancement:** This will focus in adding more examples of classes that aren't doing well, with a focus on different points of view, occlusion scenarios, and different lighting conditions.
2. **Model optimization:** adding small-object detection modules, multi-scale feature fusion, and loss function improvements like Wise-IoU or adaptive label assignment to improve precision and recall in all categories.
3. **Improvements at the system level:** include adding multi-camera tracking, temporal consistency checks, and multimodal fusion (for example, RGB–

thermal) to make the system more stable in difficult environmental conditions.

By working on these areas, the system can go from basic performance to a more general and reliable ITS solution that can handle complicated real-world deployments in cities in both developed and developing countries.

References

- Aamir, S. M., Ma, H., Khan, M. A. A., & Aaqib, M. (2024). Real-time object detection in occluded environment with background cluttering effects using deep learning. *arXiv*. <https://arxiv.org/abs/2401.00986>
- Aboah, A., Wang, B., Bagci, U., & Adu-Gyamfi, Y. (2023). Real-time multi-class helmet violation detection using few-shot data sampling technique and YOLOv8. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* (pp. 5350–5358). IEEE. <https://doi.org/10.1109/CVPRW59228.2023.00564>
- Alsanabani, A. A., Ahmed, M. A., & Al Smadi, A. M. (2021). Vehicle counting using detecting-tracking combinations: A comparative analysis. In *Proceedings of the 2020 4th International Conference on Video and Image Processing (ICVIP '20)* (pp. 48–54). ACM. <https://doi.org/10.1145/3447450.3447458>

- Chen, J., Liu, X., Wang, Y., & Zhang, H. (2024). SOD-YOLOv8: An improved YOLOv8 model for small object detection in traffic surveillance. *Sensors*, 24(5), 1452. <https://doi.org/10.3390/s24051452>
- Chen, Y., Liu, H., & Zhao, X. (2024). SOD-YOLOv8: Small object detection in urban traffic scenes using attention-enhanced feature fusion. *arXiv*. <https://doi.org/10.48550/arXiv.2408.04786>
- Durve, M., Orsini, S., Tiribocchi, A., et al. (2023). Benchmarking YOLOv5 and YOLOv7 models with DeepSORT for droplet tracking applications. *European Physical Journal E*, 46(32). <https://doi.org/10.1140/epje/s10189-023-00290-x>
- Folio3 Software Inc. (2024). What is YOLOv8? A step-by-step guide [Blog post]. Folio3. <https://www.folio3.ai/blog/what-is-yolov8-architecture/>
- García-Aguilar, I., Luque-Baena, R. M., & López-Rubio, E. (2021). Improved detection of small objects in road network sequences using CNN and super resolution. *University of Málaga & Biomedical Research Institute of Málaga*.
- Jegham, N., Koh, C. Y., Abdelatti, M., & Hendawi, A. (2025). YOLO evolution: A comprehensive benchmark and architectural review of YOLOv12, YOLO11, and their previous versions. *arXiv*. <https://arxiv.org/abs/2411.00201>
- Lee, C., Kim, H., Oh, S., & Doo, I. (2021). A study on building a “real-time vehicle accident and road obstacle notification model” using AI CCTV. *Applied Sciences*, 11(17), 8210. <https://doi.org/10.3390/app11178210>
- Li, M., Zhao, K., & Sun, Y. (2022). YOLO-AL: An improved YOLO-based vehicle detection algorithm for intelligent transportation systems. *Applied Sciences*, 12(10), 5546. <https://doi.org/10.3390/app12105546>
- Li, Q., Li, Y., & Lu, Y. (2025). DFE-YOLO: A multi-scale-enhanced detection network for dense object detection in traffic monitoring. *Electronics*, 14(11), 2108. <https://doi.org/10.3390/electronics14112108>
- Li, X., Zhao, Y., Xu, H., & Wang, Z. (2022). YOLO-AL: A YOLOv8-based object detection algorithm with active learning for vehicle detection. *IEEE Access*, 10, 103456–103469. <https://doi.org/10.1109/ACCESS.2022.3205893>
- Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., ... & Zitnick, C. L. (2014). Microsoft COCO: Common objects in context. In *European Conference on Computer Vision* (pp. 740–755). Springer. https://doi.org/10.1007/978-3-319-10602-1_48
- Liu, H., Duan, X., Lou, H., et al. (2023). Improved GBS-YOLOv5 algorithm based on YOLOv5 applied to UAV intelligent traffic. *Scientific Reports*, 13, 9577. <https://doi.org/10.1038/s41598-023-36781-2>

- Ma, C., & Xue, F. (2024). A review of vehicle detection methods based on computer vision. *Journal of Intelligent and Connected Vehicles*, 7(1), 1–18. <https://doi.org/10.26599/JICV.2023.9210019>
- Mamadmurodov, A., Umirzakova, S., Rakhimov, M., et al. (2025). A hybrid deep learning model for early forest fire detection. *Forests*, 16(5), 863. <https://doi.org/10.3390/fl6050863>
- Puzio, E., Drożdż, W., & Kolon, M. (2025). The role of intelligent transport systems and smart technologies in urban traffic management in Polish smart cities. *Energies*, 18(10), 2580. <https://doi.org/10.3390/en18102580>
- Qin, Z., Zhou, S., Wang, L., Duan, J., Hua, G., & Tang, W. (2023). MotionTrack: Learning robust short-term and long-term motions for multi-object tracking. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 17939–17948). IEEE. <https://doi.org/10.1109/CVPR52729.2023.01720>
- Senthil Kumar, K., & Abdullah Safwan, K. M. B. (2024). Accelerating object detection with YOLOv4 for real-time applications. *Madras Institute of Technology, Anna University*.
- Sundaresan Geetha, A., Alif, M. A. R., Hussain, M., & Allen, P. (2024). Comparative analysis of YOLOv8 and YOLOv10 in vehicle detection: Performance metrics and model efficacy. *Vehicles*, 6(3), 1364–1382. <https://doi.org/10.3390/vehicles6030065>
- Tang, J., Ye, C., Zhou, X., & Xu, L. (2024). YOLO-Fusion and Internet of Things: Advancing object detection in smart transportation. *Alexandria Engineering Journal*, 107, 1–12. <https://doi.org/10.1016/j.aej.2024.09.012>
- Tian, J. H., Feng, X. F., Li, F., et al. (2025). An improved YOLOv5n algorithm for detecting surface defects in industrial components. *Scientific Reports*, 15, 9756. <https://doi.org/10.1038/s41598-025-94109-8>
- Wang, P., Zhang, W., & Liu, F. (2023). Multimodal vehicle detection using YOLOv8 and EfficientDet for intelligent transportation systems. *IEEE Transactions on Intelligent Transportation Systems*, 24(12), 13645–13657. <https://doi.org/10.1109/TITS.2023.3245678>
- Wang, Q., Patel, S., & Kim, J. (2023). Multimodal vehicle detection in adverse conditions using YOLOv8 and EfficientDet fusion. *Sensors*, 23(14), 6452. <https://doi.org/10.3390/s23146452>
- Wei, L., Zhou, B., & Liu, B. (2021). A real-time vehicle detection method for unmanned driving. *Journal of Physics: Conference Series*, 1952(3), 032089. <https://doi.org/10.1088/1742-6596/1952/3/032089>
- Xu, R., Li, X., & Zhou, F. (2023). Comparative evaluation of YOLO architectures for object detection in urban traffic environments.

- Electronics*, 12(20), 4121.
<https://doi.org/10.3390/electronics12204121>
- Yaseen, M. (2024). What is YOLOv8: An in-depth exploration of the internal features of the next-generation object detector. *arXiv*.
<https://arxiv.org/abs/2408.15857>
- Zhang, L., Chen, J., & Guo, Y. (2023). AMW-YOLOv8n: Enhanced object detection with adaptive multi-scale weighting and Wise-IoU. *Electronics*, 13(20), 4121.
<https://doi.org/10.3390/electronics13204121>
- Zhang, M., & Zhang, Z. (2025). Research on vehicle target detection method based on improved YOLOv8. *Applied Sciences*, 15(10), 5546.
<https://doi.org/10.3390/app15105546>
- Zhang, Y., Chen, L., Huang, M., & Gao, S. (2023). Improved YOLOv8 for small object vehicle detection in complex traffic scenes. *Expert Systems with Applications*, 224, 119953.
<https://doi.org/10.1016/j.eswa.2023.119953>