SPECTRUM OF ENGINEERING SCIENCES
ISSN (E) 3007-3138 (P) 3007-312X VOLUME 4, ISSUE 6, 2026

MULTI-CLASS VEHICLE DETECTION AND CLASSIFICATION FOR TRAFFIC SURVEILLANCE USING YOLOV8 NANO

Ummi Mursaleen[*1], Syed Muhammad Faizan Alam[2], Muhammad Hassan Jawaid[3], Dr. Shahid Khan Yusufzai[4]

[*1,2,3,4]*Department of Robotics and AI, SZABIST University, Karachi, Pakistan*

[*1]msds24101144@szabist.pk, [2]msds24101141@szabist.pk, [3]msds24101124@szabist.pk, [4]shahid.khan@szabist.edu.pk

DOI: https://doi.org/10.5281/zenodo.20621323

Keywords
YOLOv8, vehicle detection, multi-class classsification, traffic surveillance, deep learning, object detection, intelligent transportation systems, data augmentation

Article History
Received: 11 April 2026
Accepted: 23 May 2026
Published: 10 June 2026

Corresponding Author: *
Ummi Mursaleen

Abstract

This rapid urbanization has led to an increasing number of vehicles on the roads, which has created a need for automated and intelligent traffic surveillance systems that can detect and classify various vehicle types in real-time. Current computer vision techniques and manual inspection processes do not effectively deal with the complexity, scale and variability of today's traffic conditions. This paper introduces an end- to-end deep learning solution for multi-class vehicle detection focusing on road traffic images with eight vehicle classes: Car, Auto, Bus, Truck, Light Commercial Vehicle (LCV), Motorcycle, Tractor, and MultiAxle. An extensive data preprocessing pipeline was created that includes image resizing, removal of corrupt images, optimization of compression, removal of duplicate labels and verification of the data set. The YOLOv8 framework automatically applied data augmentation in training, such as horizontal flipping, HSV color adjustment, translation, scaling and mosaic augmentation. The model was trained for 25 epochs, with the AdamW optimizer and split into train/validate set at 80:20. The proposed system achieved a final precision of 0.63, recall of 0.69, mAP@50 of 0.67, and mAP@50-95 of 0.44. All three loss components were found to be decreasing uniformly in both the training and validation sets as confirmed by the convergence analysis, there was no overfitting. The results show that the proposed pre-processing methodology and training setup is capable of providing a reliable multi-class vehicle detection which is efficient to be deployed in real world traffic surveillance.

I. INTRODUCTION

TRAFFIC surveillance is a key part of ITSs and smart cities development. Efficient traffic management has become one of the critical needs in modern infrastructure planning as the urban population continues to swell and the number of vehicles being operated on the roads continues to rise. Conventional frame-differencing computer vision systems, inductive loop detectors and manual observation are becoming increasingly insufficient to deal with traffic measurement needs in modern urban areas. These legacy methods cannot deliver accurate analytics in real-time during traffic congestion, partial occlusion, low visibility, weather and other difficult scenarios, especially when traffic contains large numbers of multi-class vehicles.

The power of traffic surveillance technologies has

been revolutionized with the latest deep learning technologies. Deep learning architectures, such as Convolutional Neural Networks (CNNs) and transformer-based models can learn complex hierarchical visual features directly from traffic video data, without relying on handcrafted feature descriptors. This means that much more precise vehicle detection, classification and tracking is possible than with traditional methods, and that systems can be adapted to various environmental conditions. In the realm of deep learning paradigms for object detection, YOLO (You Only Look Once) family models have proven to be a key paradigm for their ability to combine detection and classification into a single-stage inference pipeline, achieving a good computational speed while maintaining detection ac- curacy.

YOLOv8 is the latest version of this architecture, featuring a fully anchor-free detection head with a separate classification and localization branch, thus enhancing localization accuracy without increasing inference overhead. The Nano version of YOLOv8 (yolov8n.pt) is optimized for resource-poor environments and offers comparable detection performance while using far fewer parameters than the larger variants.

This paper brings the following contributions. First, a com- prehensive pre-processing pipeline is developed for road traffic datasets following the YOLO annotation standard to tackle various data quality issues that are common in real-world deployments and affect the road traffic detection performance. Second, this YOLOv8 Nano model is trained and tested using a multi-vehicle dataset covering the entire spectrum of vehicle types present in traffic scenes in South Asia, including Tractor and Rickshaw (Auto) classes, which are not much used in benchmark datasets. Third, a detailed convergence analysis is provided along with per-class detection performance evaluation through normalized confusion matrices and metric progression curves, giving a detailed understanding for the behaviour of the models across the different vehicle classes. The rest of this paper is organized as follows. Section II provides a review of the related works on the detection of vehicles using deep learning. The data set is described in Section III. The preprocessing pipeline is presented in Section IV. The training methodology and YOLOv8 architecture are explained in Section V. Experimental results and analysis are given in Section VI. Section VII discusses findings, limitations, and future directions. The paper is concluded with Section VIII.

II. Literature Review

In the last decade, Vehicle Detection and Classification (VDC) in traffic surveillance has come a long way from hand- engineered feature-based methods to fully learned deep neural network methods. This section summarizes the literature related to the most significant previous research and clarifies the gaps in the literature that are the motivation for the current study.

El Mallahi et al. [1] propose the UA-DETRAC traffic surveillance benchmark and a vehicle detection and classification framework using Faster R-CNN. They were able to develop frame sequences from video streams and use region proposal networks for multi-class localization to improve precision and recall over previous YOLO-related baselines. The Faster R-CNN architecture achieves high localization accuracy with two-stage region proposals, but has higher inference latencies than single-stage region proposal detectors. The work showed that region-proposal networks work well in complex urban traffic scenarios but didn't consider deployment constraints on resource-limited hardware.

Pillai [2] developed a real-world roadside camera deployment traffic surveillance framework based on YOLO. The system was able to combine vehicle detection, vehicle tracking and vehicle classification from continuous video streams and used data augmentation to enhance the accuracy for different lighting and occlusion scenarios. The paper showed that the system is practically applicable to smart transportation systems and proved that a light-weight detection pipeline can be used to get actionable traffic information in dense environments. The work, however, was mainly limited to a small number of vehicle classes and the evaluation of the performance was

not conducted on a mix of vehicle classes which are common in traffic from developing countries.

Alotaibi et al. [3] introduced the concept of detecting and identifying vehicles in remote sensing images, which includes satellite and aerial data sources, thereby expanding traffic surveillance beyond road-side cameras to remote sensing data. The authors used hyperparameter tuning using the chaotic equilibrium optimization method with high detection accuracy of small and dense vehicles in overhead images. This work is important because it showed that it is possible to implement such a system with aerial traffic surveillance, although the field of the domain of imagery features of overhead images is significantly different from those of the ground-based camera systems that were used in this work.

Luo et al. [4] proposed an improved YOLOv5s and Deep- SORT system that is specially designed for the application of monitoring highway traffic, which is the ability to detect vehicles with high accuracy, track the multi-objects, and estimate the vehicle's speed with spatial calibration. DeepSORT integration allows continuous tracking of vehicle identity over time, a key requirement in traffic monitoring and camera speeding. This method works best in highway traffic with rather sparse and regular flows; however, dense and mixed traffic flow is not considered in the adopted method in the context of urban South Asian traffic.

In multi-class vehicle detection, the model proposed by Ramakalyani and Umamaheswaran [5] utilizes a Vision Trans- former architecture for a traffic surveillance system or smart city application. They showed that their transformer architecture offered higher feature representation and context understanding than simply CNN based architecture, and thus better detection performance in complex urban scenes. While the work showcases an emerging trend in detection research, the former type of models (transformer-based) usually require more computational resources than small CNN models and are more difficult to deploy on the edge.

In a real-time vehicle detection and classification system, Kenza et al. [6] proposed using deep transfer learning to fine-tune pre-trained convolutional networks for video data. They solved practical problem of occlusion and low resolution footage and showed that it achieved high accuracy and real- time inference. The hybrid ML–DL system proposed by Mehta [7] outperforms single deep learning models in challenging scenarios, incorporating YOLO for detection and CNN for classification refinement and multi-object tracking.

Khanpour et al. [8] designed an intelligent traffic surveil- lance system using an unmanned aerial vehicle (UAV) that processes aerial video to detect, classify, track, and analyze vehicle behaviors in real time. While UAVs can cover a wide area that cannot be covered by fixed cameras, there are also some challenges in use, such as camera instability, change of altitude and motion blur, which needs to be handled separately. Al Rabbani Alif [9] investigated the YOLOv11 model for vehicle detection in ITSs, which achieved important efficiency gains from the previous version of YOLO, surpassing it in terms of small and occluded vehicle detection and speeding up the inference time.

A. Research Gap

Although a lot of research has been conducted in the field of deep learning-based vehicle detection, there are still several significant issues lacking. Most existing works test their detection performance on standard test sets like UA-DETRAC, COCO, or Pascal VOC, that are not representative of the diversity of vehicles used in South Asian or developing world road traffic systems. Some categories are completely missing from these benchmarks, but are a large share of real-world traffic including Tractor, Rickshaw (Auto), and MultiAxle heavy vehicles. Additionally, many studies that claim to achieve high detection accuracy do not specify how accurate or accurate they are for each individual class, and this is what would show the difficulties that arise in the detection of categories of vehicles that are not well represented. To tackle these weaknesses, the present work directly reduces the number of model parameters to train and evaluate a YOLOv8 Nano model using a locally

assembled eight-class dataset, which contains all major vehicle types in the target traffic environment, and performs detailed analysis of the per-class confusion matrix.

TABLE I
DATASET COMPOSITION SUMMARY

Property	Value
Total Training Images	≈6,500
Total Validation Images	>2,959
Number of Classes	8
Dataset Split	80% train / 20% validation
Annotation Format	YOLO TXT (normalized)

Vehicle Classes	
1	Car
2	Auto (Rickshaw)
3	Bus
4	Truck
5	LCV (Light Commercial Vehicle)
6	Motorcycle
7	Tractor
8	MultiAxle

III. DATASET DESCRIPTION

The dataset used in this research consists of road traffic images collected from real-world surveillance scenarios containing multiple vehicle categories representative of South Asian urban and highway traffic. All images are annotated in the standard YOLO format, where each image is accompanied by a corresponding annotation file in plain text (.txt) format. Each annotation entry specifies the object class index followed by four normalized coordinates representing the bounding box center position, width, and height relative to image dimensions. This normalized representation ensures that annotations remain valid across images of different resolutions.

The complete dataset comprises approximately 6,500 training images and more than 2,959 validation images distributed across eight vehicle classes: Car, Auto, Bus, Truck, LCV (Light Commercial Vehicle), Motorcycle, Tractor, and Multi Axle. The dataset is organized in four directories: training images, training labels, validation images, and validation la- bels. Dataset configuration is managed through a data.yaml file specifying the dataset paths, class names, and the total number of detection categories. Table I summarizes the dataset composition.

The dataset exhibits significant class imbalance, with Car and LCV being the most frequently represented categories while Tractor and Bus contain substantially fewer annotated instances. This imbalance directly influences per class detection performance and motivates future work involving class- balanced sampling strategies.

IV. PREPROCESSING PIPELINE

The performance of object detection models is strongly dependent on the quality and uniformity of the training data. To ensure high data quality, minimize computational load, stabilize training, and maximize detection accuracy, the com- prehensive multi-stage preprocessing pipeline was developed before training.

A. Dataset Splitting

Initially, all images and annotation files were stored within a single training directory. To

enable rigorous evaluation of model generalization, the dataset was partitioned into training and validation subsets using an 80:20 ratio. Random shuffling was applied before splitting to ensure that the distribution of vehicle classes remained statistically balanced across both subsets, preventing systematic ordering effects from biasing the validation results. The splitting process was implemented using Python's os, random, and shutil libraries. During each file transfer, the corresponding annotation file was moved alongside its paired image to strictly preserve the one-to-one relationship between images and labels that YOLO training requires.

B. Image Resizing

All images were uniformly resized to a fixed resolution of 640 × 640 pixels before training. Consistent input dimensions are essential for convolutional neural network operation, as convolutional layers and their associated stride patterns re- quire fixed spatial dimensions throughout the forward pass. Uniform resizing also eliminates the need for variable-length padding operations and facilitates stable gradient computation during backpropagation. The 640×640 resolution was selected because it represents the standard input size used during YOLOv8 pretraining, ensuring optimal compatibility with the pretrained feature extraction weights.

C. Corrupt Image Removal

OpenCV library was used to systematically scan the image files for any corrupted or invalid ones. An attempt was made to load each file and files that were unable to be successfully decoded were automatically marked as fail and deleted from both the image and label directories. Silent runtime failures during dataloader iteration, NaN gradients due to malformed tensors and instability of the training process can be caused by corrupt images. Corrupt images can lead to silent runtime failures, NaN gradients from malformed tensors, and instability of the training process that is hard to debug. This step made sure that all future pre-processing and training steps would be performed on a structurally correct data set.

D. Image Compression and Optimization

All images were re-compressed and stored using optimal JPEG compression in order to significantly decrease storage space and maximize the speed of loading images for training. Disk I/O operations are often a significant bottleneck when training in CPU-limited environments which reduces the GPU's contribution to the training time and reduces the efficiency of the training. This process significantly decreased the size of each image on disk, while maintaining acceptable image quality for image detection, which in turn enhanced dataloader performance and lowered total training time.

E. Duplicate Label Removal

To eliminate data duplication, entries with duplicate bounding boxes, multiple annotation lines for the same object at similar or identical coordinates within a label file, were checked in the annotation files. During training, duplicate annotations add redundant supervision signals that lead to over-annotated objects artificially inflating the gradient updates received by the model. More importantly, they cause non-maximum suppression (NMS) failure during inference by inferring two detections of the same object. Each label file was cleaned and processed to keep only unique annotations, so clean and unambiguous ground-truth supervision.

F. Dataset Verification

After all the preprocessing transformations, a systematic check was done on all the data. This verification looked for three types of structural inconsistency: (1) images without accompanying annotation files, (2) annotation files that did not contain an image, and (3) malformed annotation entries with unusable coordinate values outside the range of [0, 1]. During verification, any pairs that did not match or were not valid were eliminated prior to training. This last validation step ensured that there was full consistency between all image–annotation pairs in both the training and validation subsets.

G. *Data Augmentation*

The YOLOv8 training framework has embedded a sophisticated set of data augmentation transformations within training itself that present the model to a wide variety of variations in images to enhance the model's ability to perform well on generalisation. The augmentations are performed on-the-fly during the iteration of the dataloader, and include horizontal flipping for creating mirror-image samples and thus enhance left-right spatial invariance, HSV color space transformations where hue, saturation, and brightness values are modified independently to create samples under different lighting and atmospheric conditions, translation and scaling to simulate vehicles at different distances and camera settings, and mosaic augmentation in which four distinct training images are merged together to form one input tile, forcing the model to detect objects across a wide range of scales and spatial context in one training step. All such augmentation techniques can together greatly increase the size of the training distribution and mitigate the danger of overfitting on the visual properties of the training images.

TABLE II
YOLOv8 NANO TRAINING CONFIGURATION

Parameter	Value
Model Architecture	YOLOv8 Nano (yolov8n.pt) Epochs 25
Image Size	640 × 640
Batch Size	8
Optimizer	AdamW
Number of Classes	8
Early Stopping Patience	20 epochs Device GPU

V. MODEL ARCHITECTURE AND TRAINING

A. *YOLOv8 Nano Architecture*

In our study, YOLOv8 Nano architecture model with pre- trained weights yolov8n.pt is used for object detection. YOLOv8 is the latest entry in the YOLO model family, with a few architectural improvements on its predecessors. Most importantly, YOLOv8 changes to a fully anchor-free paradigm that does not need to tune the anchor box size on the dataset as done in previous versions of YOLO. Instead, the model directly regresses the center coordinates and dimensions of the bounding boxes of the features with respect to the location of the feature map.

YOLOv8 is based on a decoupled detection head, which separates the classification branch and the bounding-box regression branch into independent prediction branches. This decoupling allows the gradient signals for these two tasks to not interfere with one another during backpropagation, and has been empirically demonstrated to help accelerate convergence and improve classification accuracy and/or localization precision. The backbone network adopts C2f (Cross-Stage Partial with two bottleneck blocks) modules to enhance the gradient flow and reusability for depth of network.

The Nano variant was chosen for this research because it is a good balance of detection accuracy and computational resource requirements. YOLOv8n has about 3.2 million parameters, enabling it to perform well in detection tasks and still be run on hardware without dedicated GPUs, a key feature for real-world traffic monitoring deployment in resource-limited environments.

B. *Training Configuration*

Hyperparameter values used to train the model

are given in Table II. The training process was carried out on 25 epochs with AdamW optimizer. AdamW separates the weight decay regularization term from the adaptive gradient moment estimates, allowing the weight decay to not affect the adaptive learning rate mechanism and resulting in more effective regularization than standard AdamW with L2 penalty. An early stopping patience of 20 epochs was set to automatically stop training if the validation loss didn't improve over 20 epochs, so when the model has converged the training process won't continue unnecessarily.

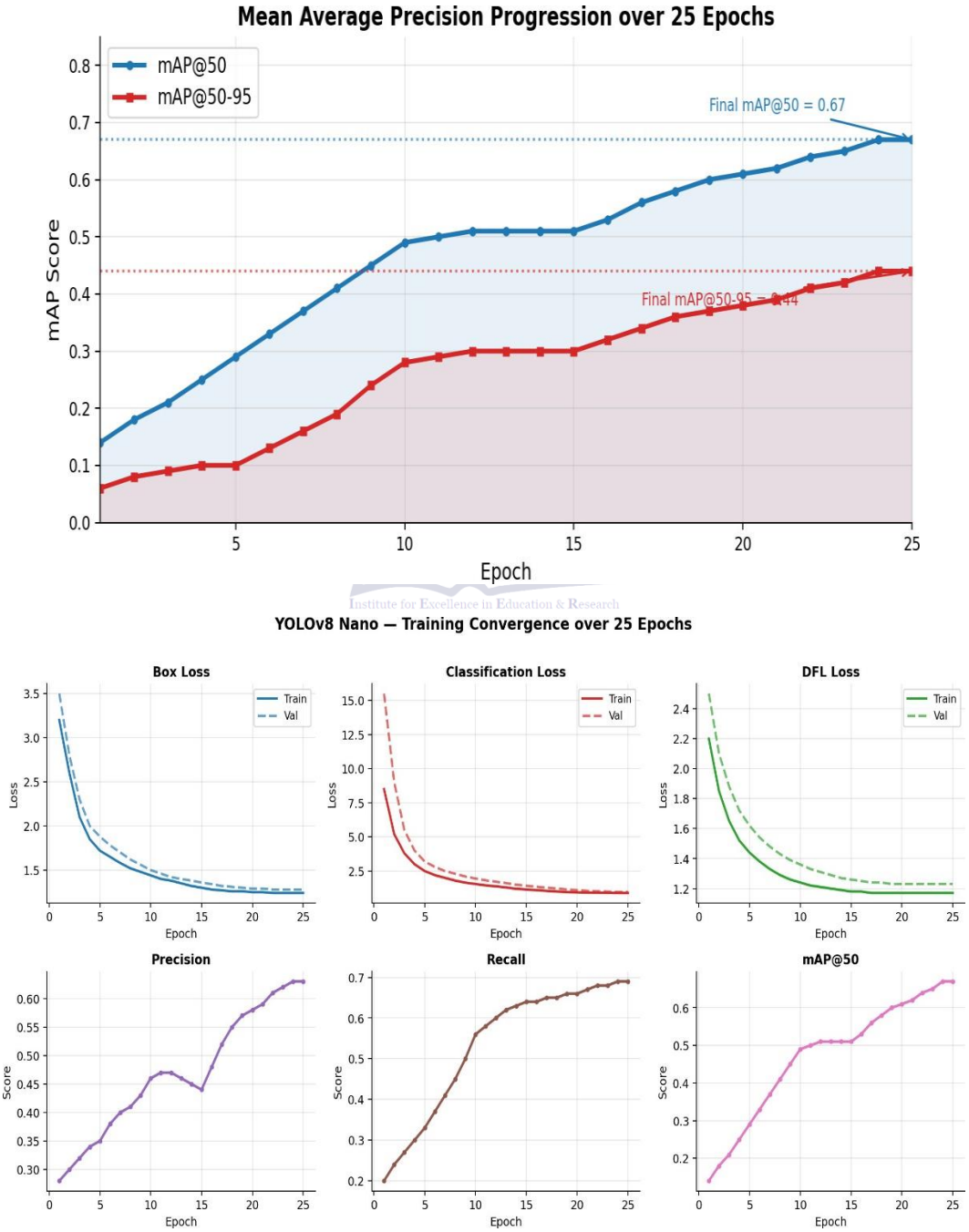

Fig. 1. Training and validation loss curves (box loss, classification loss, DFL loss) and metric progression

(precision, recall, mAP@50) over 25 epochs.

TABLE III
EPOCH-WISE DETECTION PERFORMANCE METRICS

Epoch	Precision	Recall	mAP@50	mAP@50-95
1	0.28	0.20	0.14	0.06
5	0.35	0.33	0.29	0.10
10	0.46	0.56	0.49	0.28
15	0.44	0.64	0.51	0.30
20	0.58	0.66	0.61	0.38
25	0.63	0.69	0.67	0.44

VI. EXPERIMENTAL RESULTS

A. Training Convergence Analysis

Training and validation metrics were monitored throughout all 25 epochs to assess convergence behavior and model generalization. Fig. 1 presents the training curves for all three loss components and three key detection metrics.

All three loss components — box loss, classification loss, and distribution focal loss (DFL) — exhibited consistent and steady decline in both the training and validation sets across the full 25-epoch training duration. The validation classification loss, which began at approximately 15.5 in epoch 1, decreased sharply to below 2.5 within the first five epochs, reflecting rapid early learning of the dominant class distributions. Both training and validation loss curves tracked each other closely throughout training without diverging, which is the characteristic signature of a well-regularized model that generalizes effectively to unseen data rather than memorizing training samples.

Table III summarizes the epoch-wise progression of the four primary detection metrics.

The model demonstrated progressive and sustained improvement across all four metrics throughout the training duration. Precision increased from 0.28 at epoch 1 to 0.63 at epoch 25, while recall improved from 0.20 to 0.69. The mean Average Precision at IoU threshold 0.50 rose from 0.14 to 0.67, and mAP@50-95 grew from 0.06 to 0.44. This consistent upward trend across all metrics, sustained through epoch 25 without saturation, indicates that the model continued to refine its feature representations and detection boundaries throughout the Fig. 2. mAP@50 and mAP@50-95 progression over 25 epochs. Final values: mAP@50 = 0.67, mAP@50-95 = 0.44.

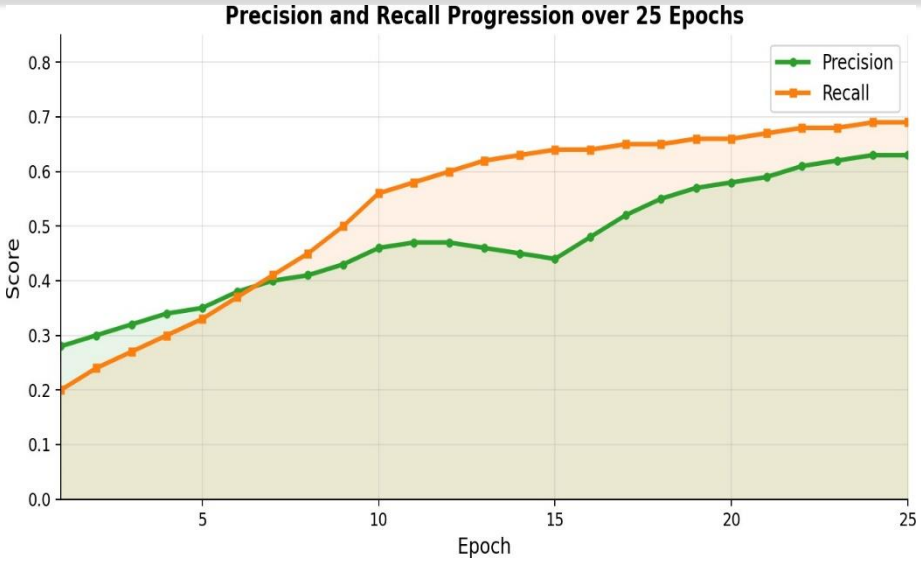

Fig. 3. Precision and recall progression over 25 epochs. Final precision = 0.63, final recall = 0.69. full training duration and had not yet reached its performance ceiling.

B. mAP Progression

Fig. 2 presents the detailed progression of mAP@50 and mAP@50-95 across all 25 epochs.

The mAP@50 curve demonstrates a sharp initial rise during the first ten epochs as the model rapidly learns the dominant visual features of high-frequency classes such as Car and LCV, followed by a more gradual but sustained improvement as the model refines its ability to distinguish visually similar categories. The mAP@50-95 metric, which averages precision across ten IoU thresholds from 0.50 to 0.95, grows more slowly, reflecting the additional challenge of achieving precise localization at the more stringent overlap thresholds. The gap between mAP@50 and mAP@50-95 remains relatively stable across training, suggesting that localization precision and classification accuracy improve proportionally.

C. Precision and Recall Progression

Fig. 3 shows the progression of precision and recall separately over the training period.

Recall demonstrates a steeper early-phase improvement compared to precision, which reflects the model's initial tendency to detect the majority of true positive instances at the cost of some false positives. As training progresses, precision catches up as the model learns to suppress false detections through improved class discrimination. The final model achieves a balanced precision-recall trade-off with recall slightly exceeding precision (0.69 vs. 0.63), indicating a configuration that prioritizes comprehensive detection coverage over false-positive suppression — a desirable property for traffic surveillance applications where missed detections are generally more costly than occasional false alarms.

TABLE IV
FINAL TRAINING AND VALIDATION PERFORMANCE METRICS

Metric	Value	Interpretation
Precision	0.63	Good detection precision
Recall	0.69	Good object coverage
mAP@50	0.67	Strong detection performance
mAP@50-95	0.44	Moderate localization accuracy
Train Box Loss	1.24	Low localization error
Validation Box Loss	1.28	Good generalization
Train Classification Loss	0.89	Effective class learning

Validation Classification Loss	0.97	Stable classification
Train DFL Loss	1.17	Improved box regression
Validation DFL Loss	1.23	Consistent validation perf.

D. Final Model Performance

The complete set of final training and validation performance metrics is summarized in Table IV. The minimal gap between training and validation losses across all three components confirms that the model has generalized well to the held-out validation set and has not overfitted to the training data distribution despite the class imbalance present in the dataset.

E. Confusion Matrix Analysis

Fig. 4 presents the normalized confusion matrix computed over the validation set, providing a comprehensive per-class breakdown of the model's classification behavior.

The confusion matrix reveals a strongly asymmetric performance profile across vehicle categories. The Car class achieved the highest absolute count of correct predictions at 7,434, followed by LCV with 4,434 correct detections, reflecting the high representation of these categories in the training dataset and the relative visual distinctiveness of their bounding box dimensions. MultiAxle vehicles recorded 1,243 correct predictions, while Auto and Motorcycle classes showed moderate performance. Truck and Bus exhibited lower true positive counts at 339 and 228 respectively, and Tractor yielded the fewest correct predictions at 76 — directly attributable to its very limited representation in the training corpus.

Notable misclassification patterns include: 610 background image regions being incorrectly predicted as Car instances, reflecting the model's slight bias toward the dominant class; 855 LCV instances being classified as background, suggesting that the model still struggles with LCV instances in low-contrast or partially occluded scenarios; and substantial confusion among MultiAxle, Bus, and Truck categories (188, 308, and 255 cross-class confusions respectively), which is expected given the visual similarity of these large commercial vehicle types in traffic imagery.

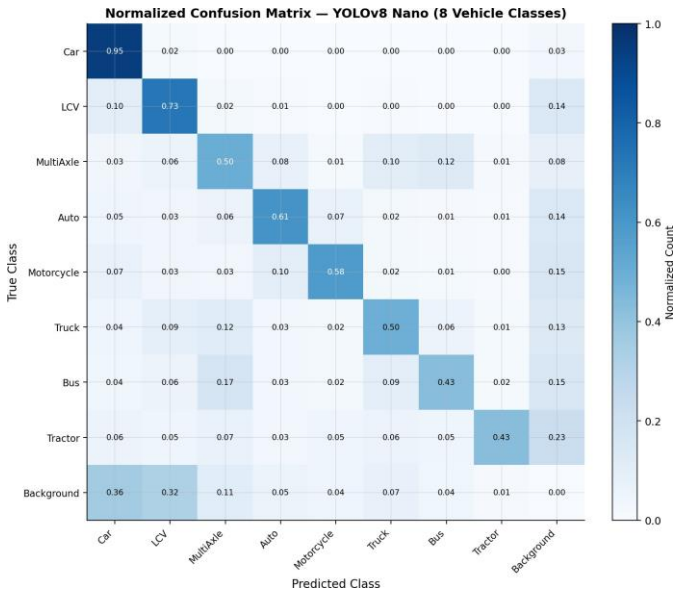

Fig. 4. Normalized confusion matrix for multi-class vehicle detection across 8 vehicle categories and background class.

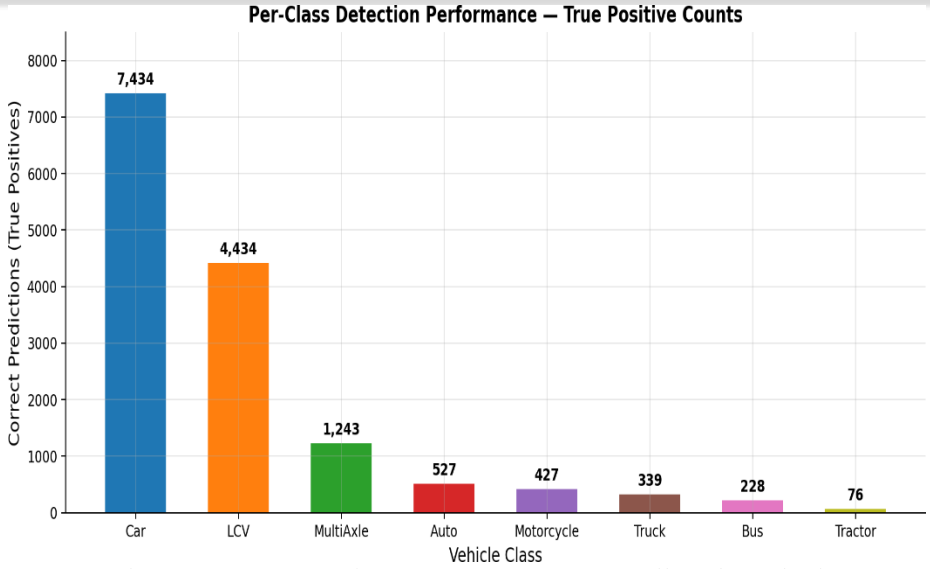

Fig. 5. Per-class true positive detection counts across all eight vehicle categories.

TABLE V
PER-CLASS DETECTION PERFORMANCE SUMMARY

Vehicle Class	Correct Predictions	Level
Car	7,434	Excellent
LCV	4,434	Excellent
MultiAxle	1,243	Good
Auto	527	Moderate
Motorcycle	427	Moderate
Truck	339	Moderate
Bus	228	Moderate
Tractor	76	Weak

F. Per-Class Detection Performance

Fig. 5 presents the per-class true positive counts, providing an intuitive visualization of performance heterogeneity across vehicle categories.

Table V complements Fig. 5 with performance level assessments for each class.

The performance disparity between Car (7,434 correct detections) and Tractor (76 correct detections) spans nearly two orders of magnitude and reflects the combined effect of training set class imbalance and intrinsic visual challenge. Tractor instances in road traffic imagery are relatively rare, visually diverse across different agricultural configurations, and frequently partially occluded by other vehicles, making accurate detection particularly challenging without targeted augmentation and oversampling strategies.

VII. DISCUSSION

The experimental findings confirm that the proposed pre- processing pipeline and the YOLOv8 Nano architecture is a suitable and practical approach for multi-class vehicle detection in road traffic surveillance. The result of 0.67 for the mAP over eight vehicle categories (including some categories that were not well

represented or visually difficult), proves the effectiveness of the preprocessing methodology used for generating clean and well-structured training data.

The loss curves show a consistent trend between the training and validation losses, suggesting that the model was not overfitting despite the small batch size of 8.The loss curves also reveal that the training and validation losses have followed a similar trend, with no significant divergence between them, which indicates that the model has not overfitted due to the relatively small batch size of 8. The early stopping mechanism with patience set at 20 epochs helped prevent the model from diverging during training, yet still enabled it to fully utilize the learning capacity available within 25 epochs.

Per-class analysis shows that an important practical constraint is that the model is able to perform much better in high-frequency classes such as Car and LCV when com- pared to low-frequency classes such as Tractor and Bus. This performance imbalance is directly attributed to the training dataset imbalance and is the main aspect that needs to be im- proved upon in future iterations. For Tractor, Bus, and Truck, class-balanced sampling strategies at the data level (such as SMOTE), class-weighted loss functions at the optimization level or targeted collection of additional annotated samples for underrepresented categories would be expected to significantly boost the recall of detection for these categories.

Furthermore, the misclassification analysis shows that a fine-grained classification of the "MultiAxle", "Truck" and "Bus" classes of large commercial vehicles remains a difficult problem, especially when occlusion or atypical angles occur. Future work could include the use of attention mechanisms or transformer-based feature extraction modules to extract richer context representations to separate similar categories visually.

VIII. Conclusion

This paper introduces an entirely deep learning-based system for detecting multiple classes of vehicles in road traffic monitoring using the YOLOv8 Nano architecture. A thorough preprocessing procedure was designed including splitting the dataset, resizing the images to 640 × 640 pixels, discarding corrupted images, optimizing the JPEG compression, eliminating duplicate annotations and verifying the entire dataset. Overall, these pre-processing procedures ensured that the data used for training was clean, consistent, and efficient in terms of computational resources.

Under the above setup, the YOLOv8 Nano model achieved an 8-class Mean Average Precision (mAP) of 0.63, a Mean Average Recall (mAR) of 0.69, a Mean Average Precision at 50 (mAP@50) of 0.67, and a Mean Average Precision at 50-95 (mAP@50-95) of 0.44 on the validation set. Convergence analysis showed that all the loss components decreased monotonously without overfitting, thus indicating good generalization of the model. The confusion matrix analysis per class showed good results for Car and LCV classes and Tractor and Bus as the underperforming classes where the dataset can be augmented in future iterations.

The proposed system is shown to be feasible for multi- class vehicle detection in resource-limited traffic surveillance systems. Future research will focus on larger YOLOv8 models (Small, Medium, Large), longer training times with more epochs, class-balanced oversampling methods, adding multi- object tracking with DeepSORT, and testing on edge inference devices for real-time roadside monitoring.

References

I. El Mallahi, J. Riffi, H. Tairi, and M. A. Mahraz, "Efficient vehicle detection and classification algorithm using Faster R-CNN models," *J. Autom. Mobile Robot. Intell. Syst.*, vol. 18, no. 4, pp. 86–93, 2024.

A. S. Pillai, "Traffic surveillance systems through advanced detection, tracking, and classification technique," *Int. J. Sustain. Infrastruct. Cities Soc.*, vol. 8, no. 9, pp. 11–23, 2023.

Y. Alotaibi, K. Nagappan, T. Thanarajan, and S. Rajendran, "Optimal deep learning based vehicle detection and classification using chaotic equilibrium optimization algorithm in remote sensing imagery," *Sci. Rep.*, vol. 15, no. 1, p. 17921, 2025.

Z. Luo, Y. Bi, X. Yang, Y. Li, S. Yu, M. Wu, and Q. Ye, "Enhanced YOLOv5s + DeepSORT method for highway vehicle speed detection and multi-sensor verification," *Front. Phys.*, 2024.

K. Ramakalyani and S. Umamaheswaran, "Deep learning-based multi- class vehicle detection: a high-speed approach for traffic surveillance and smart city applications," *J. Wireless Mobile Netw. Ubiquitous Comput. Dependable Appl.*, vol. 16, pp. 579–600, 2025.

B. Kenza, T. Ranya, H. Meryeme, and H. M. Alami, "Real-time vehicle detection and classification using deep-transfer learning," in *Proc. 25th Int. Arab Conf. Inf. Technol. (ACIT)*, Zarqa, Jordan, 2024, pp. 1–6.

R. Mehta, "A hybrid ML–DL framework for real-time vehicle detection, classification, and tracking in intelligent traffic surveillance," *Int. J. Appl. Math.*, vol. 38, pp. 1285–1294, 2025.

A. Khanpour *et al.*, "UAV-based intelligent traffic surveillance system: real-time vehicle detection, classification, tracking, and behavioral anal- ysis," arXiv:2509.04624, 2025.

M. Al Rabbani Alif, "YOLOv11 for vehicle detection: advancements, performance, and applications in intelligent transportation systems," arXiv:2410.22898, 2024.

A. Nahar, M. Islam, M. H. Shuvo, M. Hasan, S. Parvin, and K. Nur, "A deep learning-based framework for accurate detection and classification of on-road vehicles using improved YOLOv11," in *Proc. Int. Conf. Electr. Comput. Commun. Eng. (ECCE)*, Chittagong, Bangladesh, 2025,pp. 1–6.

J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 779–788.

Ultralytics, "YOLOv8 documentation," [Online]. Available: https://docs. ultralytics.com