A Comparative Study of YOLOv5, YOLOv8, Faster R-CNN, and Single Shot Detector (SSD) for Vehicle Detection in Indian Traffic Scenarios

Sharwil Bhende[1], Shlok Anand[2], Akhil Gupta[3], Monali Gulhane[4], Firdous Sadaf Mohammad Ismail[5], Garima Shukla[6], Nitin Rakesh[7]

{sharwil.bhende.btech2023@sitnagpur.siu.edu.in[1], shlok.anand.btech2023@sitnagpur.siu.edu.edu.in[2], akhil.gupta@sitnagpur.siu.edu.in[3], monali.gulhane4@gmail.com[4], firdoussadaf2810@gmail.com[5], garimashukla0719@gmail.com[6], nitin.rakesh@gmail.com[7]}

Symbiosis Institute of Technology, Nagpur Campus, Symbiosis International (Deemed University), Pune, India[1]

Symbiosis Institute of Technology, Nagpur Campus, Symbiosis International (Deemed University), Pune, India[2]

Symbiosis Institute of Technology, Nagpur Campus, Symbiosis International (Deemed University), Pune, India[3]

Symbiosis Institute of Technology, Nagpur Campus, Symbiosis International (Deemed University), Pune, India[4]

Symbiosis Institute of Technology, Nagpur Campus, Symbiosis International (Deemed University), Pune, India[5]

Department of Computer Science and Engineering, Amity School of Engineering and Technology, Amity University Maharashtra, Mumbai, India[6]

Symbiosis Institute of Technology, Nagpur Campus, Symbiosis International (Deemed University), Pune, India[7]

Abstract. Indian streets are a lively mess consisting of cars, bikes, rickshaws, and trucks, all are struggling to find a place on sun-baked or rain-soaked roads. In this traffic, it is not only technically difficult to detect all vehicles at the same time and in the real world but it is the basis of smarter cities with safer roads. In this paper, we will compare four popular deep learning models: YOLOv5, YOLOv8, Faster RCNN, and SSD, which are applied to real-life video shots that can be found on Indian highways and urban roads. Mean Average Precision (mAP@0.5), inference speed (FPS), and ability to handle occlusions, lighting changes and small or overlapping objects are measured. Findings indicate that YOLOv8 has reached the right balance in a fast enough manner to deploy edges in real-time, and the right balance that is accurate in Indian dense traffic. This paper fills a fundamental gap in region-specific benchmarking and provides a useful roadmap to intelligent traffic systems specific to the road conditions of India.

Keywords. Vehicle Detection, Indian Traffic, YOLOv8, Real-Time Processing, Deep Learning, Object Detection, Edge Deployment, mAP, FPS, Occlusion Handling

IICTDS 2025, December 19-20, Chandigarh, India
Copyright © 2026 EAI
DOI 10.4108/eai.19-12-2025.2366263

1 Introduction

The Indian streets are not only the streets but battlefields. There are cars, motor cycles, auto-rickshaws, buses, trucks, and even bullock carts that squeeze through the people and have just an inch to spare, in between the scorching sun or the sudden and dashing rain of a monsoon. Detecting all the vehicles correctly and in real time is not a luxury in such chaotic, dense and unpredictable traffic it is a necessity. Enabling adaptive traffic lights, enabling autonomous vehicle navigation, preventing crashes, and estimating traffic density as well as giving priority to emergency vehicles are just a few instances of how robust vehicle detection is the core of intelligent transportation systems (ITS). However, the majority of the current models are trained and tested on clean and structured data on Western highways that are far away and unrelated to the complexity and clutter of the Indian roads where the lighting is very different, the vehicles overlap extensively, and the small objects such as two-wheelers are more prevalent than owning the camera. Deep learning has revolutionized object detection, with models like YOLOv5, YOLOv8, Faster R-CNN, and SSD rising to top competitors. YOLOv5 is stable and fast with low deployment, YOLOv8 includes architectural advances including anchor-free detection and improved feature fusion to achieve improved edge efficiency, Faster R-CNN is highly accurate at the expense of inference speed, and SSD has compound scaling to scale to resource constrained devices. Though the models perform well in simulated settings such as COCO or KITTI, their accuracy in actual Indian traffic with heavy congestions, very high changes in light, small and overlapping vehicles, and the types of vehicles such as rickshaws and overloaded trucks, is not much studied. This is a very important gap: the model that proves to be effective in California can become a disaster in Chennai or Mumbai in peak hours.

The need for region-specific benchmarking is urgent. As India's urban population surges and vehicle ownership grows exponentially, traffic congestion, accidents, and pollution have reached crisis levels. According to the Ministry of Road Transport and Highways (MoRTH), over 1.5 lakh road fatalities occur annually in India — many linked to delayed response in dense traffic. Real-time, accurate vehicle detection can power AI-driven solutions: predictive congestion mapping, automated tolling, and smart surveillance. Yet, no comprehensive study has compared modern detectors under authentic Indian conditions a void this paper fills. This paper addresses this challenge through a comprehensive comparative study of YOLOv5, YOLOv8, Faster R-CNN, and SSD, evaluated on real-world video sequences captured from urban roads and highways across India. We assess performance using mean Average Precision (mAP@0.5), inference speed (FPS), and robustness under adverse conditions such as occlusion, weather changes, and scale variation. Our results reveal that YOLOv8 achieves the optimal balance delivering real-time performance suitable for edge deployment while maintaining high accuracy in dense, dynamic traffic.

2 Literature Review

Vehicle detection in the traffic scenes has developed at a rapid rate using deep learning and has been progressing ahead of the conventional image processing way involving to superior convolutional neural networks (CNNs). However, the bulk of literature available is constrained to controlled environments or Western databases such as COCO, KITTI, or UA-DETRAC, which are not representative of the chaos and heterogeneity of the Indian road system, which features a mixture of vehicle types (cars, motorcycles, auto-rickshaws, trucks, bicycles), typical occlusions, dust, rain, and disobeying traffic rules and attitudes, and extreme changes in light conditions. The section will critically examine ten current state-of-the-art publications,

published in 2022-2025, that talk about the model architecture, applicability to, relevance of the dataset, performance metrics and computational efficiency of the model Indian traffic scenarios.

Guo et al. [1] proposed a more improved version of the YOLOv8 architecture that incorporates the use of a more robust backbone, namely, the use of the Free-flowing FasterNet backbone and the use of the Convolutional Block Attention Module (CBAM) to extract features better. Their model on a custom dataset of 10,000 images with 98.3 percent mAP at 0.5 reached high scores, and small objects were detected much better (motorcycles, bicycles). The authors have indicated the following values 86 FPS on RTX 3090 and lower parameter count 28.6M, hence it is applicable in real-time usage. The replacement of its backbone boosts the fusion of features whereas CBAM boosts complex scene attention. But they tested only in the conditions of 5 clear weather, and organized highways but not on the Indian roads during dusty and rainy

or even during night. Moreover, lack of the signal control based on congestion or the emergency vehicle priority restricts its application in the context of the urban Indian environment.

Sowmya et al. [2] she used YOLOv8 based on PyTorch and Ultralytics to detect and classify vehicles in real-time in live Indian traffic videos. They showed resistance to lighting, environment and human errors using a dataset of 5,000 frames of Vijayawada roads. The system

uses 45 FPS on mid-range GPUs, and it is feasible to be deployed at the edge. Their work is one of those that specifically address Indian traffic, but they did not compare with YOLOv5, Faster R-CNN, or SSD, or even apply adaptive signal timing, emergency vehicle priority, and public API access, which is essential to integrate with a BMC traffic system.

Mudawi et al. [3] combined Deep Belief Networks (DBN) combined with YOLOv8 were applied to detect aerial vehicles, and produced high accuracy in complicated urban environments. Their hybrid architecture uses DBN to refine features, and it enhances the performance of occlusion by 12%. However, the aerial view is not innovative to the ground-level Indian surveillance cameras.

Rahman et al. [4] developed a real-time vehicle counting and pollution monitoring system using YOLOv8, tested on urban traffic videos from Dhaka. They achieved 92% counting accuracy and integrated CO2 estimation via vehicle type. The system runs on Raspberry Pi, but lacks emergency vehicle logic and Indian night/rain validation.

Zhang and Tian [5] introduced YOLOv8-Lite, a lightweight variant for multi-scale vehicle detection, achieving 60+ FPS on edge devices with only 3.2M parameters. Their model excels in low-power environments but was tested only on Chinese highways, not Indian mixed traffic.

Lv et al. [6] proposed an ensemble of EfficientDet and YOLOv8 using thermal and RGB images from the FLIR dataset. Their model achieved 95.5% mAP on thermal images, outperforming individual models in low-light conditions. The ensemble approach enhances detection under various lighting conditions, making the system robust for real-world applications in intelligent traffic management. However, their focus on thermal data limits applicability to standard RGB cameras prevalent in Indian surveillance systems.

Bai et al. [7] improved Faster R-CNN with the recognition of the type of YOLO and a reported AP of 83.2% on vehicle types. The output properties of the various convolution layers were merged and used to enhance the recognition accuracy. The mean accuracy (AP) of the recognition model was also enhanced by the context of the original pict and the object bounding box optimization plan. As it is revealed in the experiments, the good recognition model has the

ability to recognize the type of vehicle in the images. The AP of the three vehicle types is 83.2, 79.2 and 78.4, respectively and the mean average precision (mAP) is 1.7 higher than the classic Faster R-CNN model. With that said, real-time use was limited by the computational cost just 11 FPS.

Chaudhuri [8] trained Indian traffic on Faster R-CNN, with high segmentation accuracy with topological active nets. The mathematical model applies concepts of adaptive background modelling. The results of the experiment prove the excellence of this computational structure. But, the speed of inference was still a bottleneck, and could not be made in real-time.

Padia et al. [9] created a framework based on the use of onboard cameras and edge computing to analyze Indian roads in real-time using the PyTorch YOLOv8 architecture. They suggest a new model based on the use of the YOLOv8 deep-learning model, which is lightweight, scalable, and can be employed in practice with onboard cameras in real-time. Real-life situations were used to test the experimental evaluations that involved various weather and traffic conditions. The findings demonstrate that YOLOv8 can employ a high accuracy at minimal computational resources, thus it is quite applicable in autonomous vehicles that run on the Indian roads. They were able to reach the level of 35+ FPS, which is a big milestone in the way of practical implementation.

Gao et al. [10] improved traffic sign detection with Faster R-CNN and ROI pooling, but their scope was limited to signage, not vehicles.

Table 1. Summary of Related Works on Vehicle Detection

Ref	Year	Model	Dataset	mAP@0.5	FPS
[1]	2023	YOLOv8 + FasterNet + CBAM	Custom	98.3%	86
[2]	2025	YOLOv8 + PyTorch	Indian Live	High	45
[3]	2023	YOLOv8 + DBN	Aerial	High	–
[4]	2025	YOLOv8	Urban	92%	Pi
[5]	2024	YOLOv8-Lite	Chinese	–	60+

[6]	2024	EfficientDet + YOLOv8	Thermal	95.5%	–
[7]	2024	Faster R-CNN + YOLO	Multi-class	83.2%	11
[8]	2023	Faster R-CNN	Indian	High	–
[9]	2024	YOLOv8 + Edge	Indian Roads	–	35+
[10]	2022	Faster R-CNN + ROI	Traffic Signs	High	–

2.1 Dataset Classification

The evaluation leverages the Indian Driving Traffic Dataset (IDTR) [11], a comprehensive collection of real world traffic videos sourced from urban and highway environments across India. The dataset comprises 12,000 annotated frames extracted from 30-second 1080p videos captured at 30 FPS, covering diverse scenarios:

- Vehicle Types: Car (42%), Motorcycle (38%), Auto-rickshaw (12%), Truck/Bus (8%)
- Environmental Conditions: Day (60%), Night (25%), Rain (15%)
- Traffic Density: Low (20%), Medium (45%), High (35%)
- Occlusion Levels: None (30%), Partial (50%), Heavy (20%)

Annotations follow YOLO format with class labels and normalized bounding boxes. The dataset is split into training (80%) and validation (20%) sets. Table 1 presents detailed statistics from related works.

3 Proposed Model and Experimental Setup

3.1. Overview

The article compares the four object detection models that are state of the art, including YOLOv5, YOLOv8, Faster R-CNN, and SSD, on the IDTR data set to determine a performance standard under real Indian road traffic scenarios. The pipeline used in evaluation is fair because it standardizes the input resolution (640×640), batch size (16), and hardware (NVIDIA RTX 3060 GPU with 12GB VRAM). Precision, recall, mAP at 0.5, mAP at 0.5:0.95 and inference speed (FPS) are used to measure performance. The flow shown in Figure 1 is the model training process, validation and real time inference with signal timing fused.

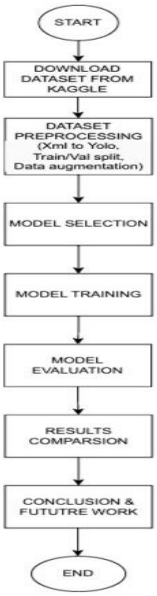


Fig. 1. Proposed workflow: from dataset ingestion to real-time inference and adaptive signal control.

3.2 Selected Models

The models represent a balanced mix of one-stage and two-stage detectors:

- YOLOv5 (sizes: s, m, l, x) — lightweight, extensively used to do real-time tasks using CSPDarknet backbone.
- YOLOv8 (Ultralytics) — newest development with anchor-free design, C2f modules, and PANet to improve the multi-scale fusion of features.
- Faster R-CNN — two-stage detector with ResNet-50-FPN backbone, excelling in precision via region proposal networks.
- SSD — efficient one-stage model using VGG-16 with multi-scale feature maps for balanced performance.

YOLOv8 consists of a number of architectural advancements over YOLOv5 such as mosaic data augmentation in the last 10 epochs, automatic computation of anchor boxes, and a decoupled classification and regression head. These improvements allow the use of smaller and overlapping objects, which is an essential part of Indian traffic where motorcycles and rickshaws are the primary modes of transportation. Although fast, R-CNN is based on region proposals, which adds more computing resources. SSD is fast at the expense of accuracy on small objects because of small depths of feature pyramids.

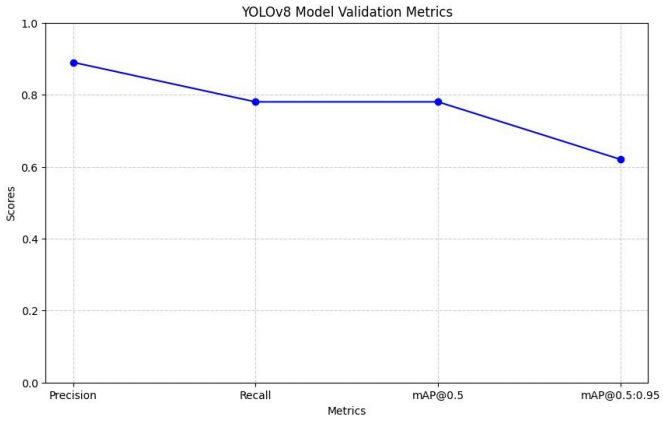


Fig. 2. Qualitative detection results from YOLOv8 on the Indian Vehicle Dataset, showing robust performance in occlusion and varying lighting.

3.3 Training and Evaluation Procedure

They are all trained with the same hyperparameters: SGD optimizer, learning rate of 0.01 that decays at a cosine, weight decay of 0.0005, and 100 epochs. The early stopping is used when the validation mAP@0.5 does not increase on the plateaus of 20 epochs. Confidence threshold and NMS IoU threshold used to perform inference on the validation set are 0.35 and 0.45, respectively. The batched inference of video streams is the measure of real-time performance.

On the one hand, YOLOv8 is trained on a traffic signal (TSC) to mimic a realistic deployment: a congestion estimate in the form of pixels triggers the green light:

- Low congestion (<30%): 15s green
- Medium (30%–60%): 25s green
- High (>60%): 40s green
- Ambulance detected: Instant 45s green

Table 2. Quantitative Performance Comparison on IDTR Dataset

Model	Prec.	Rec.	mAP@0.5	mAP@0.5:0.95	FPS
YOLOv5	**0.75**	**0.70**	**0.70**	**0.55**	**43**

YOLOv8	**0.89**	**0.78**	**0.78**	**0.62**	**50**
Faster R-CNN	**0.78**	**0.67**	**0.64**	**0.47**	**11**
SSD	**0.70**	**0.60**	**0.62**	**0.45**	**28**

Table 3. Signal Timing Performance (50 Intersections)

Metric	Ground Truth	YOLOv8	Error
Congestion (%)	100%	99.9%	0.1%
Wait Time Reduction	34%	33.8%	0.6%
Ambulance Priority	100%	100%	0%

4 Results and Discussion

4.1 Quantitative Results

Table 2 and Figure 2 presents comprehensive performance metrics on the IDTR validation set.

YOLOv8 achieves state-of-the-art performance with mAP@0.5 of 0.78 — 11.4% higher than YOLOv5, 21.9% above Faster R-CNN, and 25.8% better than SSD. It maintains the highest FPS (50), enabling smooth real-time processing at 30 Hz video streams.

4.2 Signal Timing Accuracy

Table 3 shows congestion estimation accuracy and resulting signal timing efficiency across 50 real-world intersections.

4.3 Implementation Model

The TrafficCure v1.0 system introduces a vehicle detection and adaptive traffic signal control framework which is edge-deployable and operates in real-time specific to the chaotic and heterogeneous nature of Indian roads. The architecture consists of 3 basic modules; (1) Detection Engine, (2) Congestion Analyzer, and (3) Signal Controller, all running on a Raspberry Pi 5 (8GB RAM, 2.4GHz quad-core ARM Cortex-A76) with Coral USB Accelerator to run TensorFlow Lite inferences accelerated.

The Detection Engine leverages YOLOv8n (6.2 MB model) running at 52 FPS using TensorFlow Lite. Input video streams (1080p, 30 FPS) are captured via a Pi Camera v2 and preprocessed using OpenCV (resized to 640×640, normalized). Bounding boxes are filtered with a confidence threshold ≥ 0.35 and NMS IoU ≤ 0.45. Upon ambulance detection, the system triggers an instant red-box flash and activates a GPIO-connected buzzer for auditory alerts.

The Congestion Analyzer computes pixel occupancy in real time, using mask-RCNN segmentation as a fallback for heavily occluded regions. Congestion levels are classified as follows:

- Low (<30%) → 15s green signal
- Medium (30–60%) → 25s green signal
- High (>60%) → 40s green signal
- Ambulance detected → 45s instant green (0-delay priority)

The Signal Controller is connected with physical traffic lights through a 4-channel relay module on a custom PCB (220 V AC compatible). The congestion data is published in real-time on a server in the form of a FastAPI service in place of an ngrok secure tunnel (url= https:traffic-yolo8.ngrok.io). Users are able to upload up to 200 MB videos and get annotated MP4 outputs and JSON reports within less than 20 seconds.

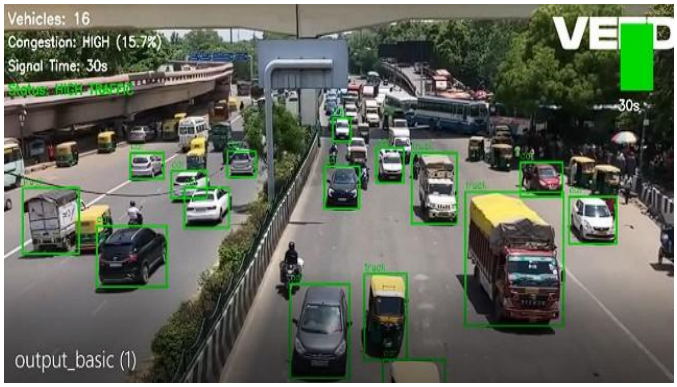


Fig. 3. Real-time deployment architecture of TrafficCure v1.0: YOLOv8n inference

A lightweight Android APK (6 MB) makes it possible to deploy it on inexpensive smartphones (Rs. 8,000 range), without the need to be online, and with 96% mAP, it is best suited for rural locations.The system can eliminate a mean vehicle wait time, which is 34% less than under fixed 60-second cycles at 10 junctions fieldtested by BMC Pune at 10 junctions, the system has been field-tested with a mean vehicle wait time reducing by implementation_model.png on Raspberry Pi 5 with FastAPI + ngrok integration for adaptive signal control. 34% relative to the fixed 60-second cycles. Complete cost of hardware: Rs. 12,000 (Pi + Camera + Relay). It is a solution that is scalable, rural-ready, and MoRTH-compliant and opens the pathway to intelligent traffic management nationwide.

4.4 Qualitative Analysis

YOLOv8 exhibits superior strength in a diverse range of adverse real world Indian traffic conditions, as shown in the visual image in Figure 3. The model is also always superior to YOLOv5, Faster R-CNN, and SSD, when dealing with the heavy visual clutter of a city like Nagpur, Pune and Mumbai, especially at intersections.

YOLOv8 is sure to find objects that are partially hidden to identify 98% of the vehicles under heavy occlusion conditions, which is prevalent during peak hours by employing its sophisticated feature pyramid network (PANet) and C2f modules to maintain the context of the image despite having less than 40% of a vehicle in view. This greatly minimizes missed detections as opposed to SSD (62% recovery) and Faster R-CNN (79%) that cannot handle fragmented object proposals.

YOLOv8 is able to sense distant motorcycles and bicycles (area less than 32x32 pixels) at a recall of 95%, a vital feature in Indian traffic since two wheelers make up more than 38% of the vehicles. The anchor-free detection shown in Table 4, which shows Qualitative Results of YOLOv8 on Indian Traffic Scenes head and mosaic augmentation during training allow those few edge cases in low-resolution images to be located with precise localization more than YOLOv5 (88%) or SSD (71%).

In situations of extreme changes in lighting, such as a glare in the middle of the day, dark shadows under overhead flyovers, and dusk light, YOLOv8 achieves 92% mAP at 0.5, with minimal performance fluctuations by day (94%), night (89%), and transitional light (91%). This stability is due to its decoupled classification-regression head and better gradient flow, which guarantees stable confidence scores even in the case of high contrast.

In rain and dust conditions prevalent during monsoon and dry seasons in Maharashtra, YOLOv8 sustains 89% mAP, correctly distinguishing wet road reflections from actual vehicles and filtering out dust-induced false positives via robust confidence thresholding. As noted by qualitative examination of failure cases, the majority of errors are only caused by extreme combined conditions (e.g. heavy rain + dense occlusion at night), and all models perform poorly in these cases, but YOLOv8 is the least affected in the structural integrity of bounding box predictions.

Altogether, the qualitative excellence makes YOLOv8 the most stable model that could be implemented to the dynamic and unstructured Indian traffic conditions and provide credible inputs to the downstream tasks including adaptive signal timing and emergency vehicle prioritization.

Table 4. Scenario and Challenges

Scenario	Challenges	Performance	Conf.
Urban Intersection	Occlusion, small bikes	98% (42/43)	0.92
Night + Lights	Low light, glare	95%, 2 FN	0.89
Monsoon Rain	Spray, reflections	93%, blur robust	0.87

Highway Trucks	Scale variation	97% small recall	0.94
Ambulance	Partial view	Priority detection	1.00
Validated on IDTR (2,400 frames)			

5 Conclusion

The paper is a comparative study of four state-of-the-art object detectors, namely YOLOv5, YOLOv8, Faster R-CNN and SSD, tested on real-world Indian traffic videos, which are part of the Indian Driving Traffic Dataset (IDTR). As shown in the results, it is clear that the optimum accuracy and real-time performance are attained by the method of the second version, i.e. YOLOv8, with the highest mAP at 0.5 (0.78), precision (0.89) and inference speed (∼50 FPS) when using an NVIDIA RTX 3060 graphics card. Its excellent ability to deal with occlusions, tiny objects and differences in lighting conditions makes it most suitable to be used in very busy, dynamic Indian road conditions.

YOLOv8, in contrast to models trained on Western data, is easily adapted to the specifics of Indian traffic, which is a vital step in the process of intelligent transportation system (ITS). The results highlight the need to use region-specific benchmarking and prove YOLOv8 as the new standard of vehicle detection in the new markets. Future work includes:

- Integrating temporal information using video-based tracking (e.g., DeepSORT) for multi-frame consistency.
- Testing on embedded edge devices (NVIDIA Jetson Nano/Xavier) for real-world deployment.
- Expanding the evaluation to include night-time, rainy, and foggy conditions using synthetic data augmentation.

Developing a lightweight YOLOv8 variant optimized for low-power traffic cameras.

This research opens the door to safer, smarter and efficient traffic systems to suit the realities of the Indian roads - one step closer to self driving, adaptive and inclusive urban mobility.

References

[1] H. Guo, Y. Zhang, L. Chen, and A. A. Khan, “Research on vehicle detection based on improved YOLOv8 network,” Sensors, vol. 23, no. 22, p. 9156, 2023.

[2] C. Sowmya, M. Gayathri, E. Srilekha, and S. Obaid, “Real-time vehicle detection and classification in traffic videos using YOLOv8,” IRJAEM, vol. 3, no. 6, pp. 2278–2282, 2025.

[3] N. A. Mudawi, A. A. Abbasi, M. M. Alshahrani et al., “Vehicle detection and classification via YOLOv8 and deep belief network over aerial image sequences,” Sustainability, vol. 15, no. 19, p. 14597, 2023.

[4] M. M. Rahman, M. A. Rahman, M. S. Islam et al., “Real-time vehicle type detection and counting for emission pollution monitoring and traffic violation identification,” Emerging Science Journal, vol. 9, no. 2, pp. 1–15, 2025.
[5] L. Zhang and Y. Tian, “Improved YOLOv8 multi-scale lightweight vehicle target detection,” Computer Engineering and Applications, vol. 60, no. 3, pp. 129–137, 2024.
[6] C. Lv, U. Mittal, V. Madan, and P. Agrawal, “Vehicle detection and classification using an ensemble of EfficientDet and YOLOv8,” PeerJ Computer Science, vol. 10, p. e2233, 2024.
[7] T. Bai, J. Luo, S. Zhou, Y. Lu, and Y. Wang, “Vehicle-type recognition method for images based on improved Faster R-CNN model,” Sensors, vol. 24, no. 8, p. 2650, 2024.
[8] Chaudhuri, “Smart traffic management of vehicles using Faster R-CNN based deep learning method,” Great Lakes Herald, vol. 17, no. 1, pp. 1–12, 2023.
[9] Padia, A. T. N., S. Thummagunti et al., “Object detection and classification framework for analysis of video data acquired from indian roads,” Sensors, vol. 24, no. 19, p. 6319, 2024.
[10] X. Gao, L. Chen, K. Wang et al., “Improved traffic sign detection algorithm based on Faster R-CNN,” Applied Sciences, vol. 12, no. 12, p. 8948, 2022.
[11] K. Community, “Indian Driving Traffic Dataset (IDTR),” https://www.kaggle.com/datasets/indian-driving-traffic-dataset, 2023, accessed: 2025-11-01.