IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 4, November 2025
Impact Factor: 7.67

Vehicle Counting for Traffic Management

Drishya A K[1] and Parimal Kumar K R[2]
Student, Department of Master of Computer Application[1]
Assistant Professor, Department of Master of Computer Application[2]
Vidya Vikas Institute of Engineering and Technology, Mysore
drishyadr54@gmail.com and parimal.kumar@vidyavikas.edu.in

Abstract: *The rapid growth of urban traffic has created significant challenges in monitoring and managing vehicle flow. Traditional approaches to vehicle counting, such as manual surveys and sensor-based systems, are often expensive, labour-intensive, and lack real-time adaptability. To address these issues, this research presents an automated vehicle detection and counting system based on the YOLOv8 deep learning model. The system processes video streams by applying preprocessing techniques, including frame resizing, grayscale conversion, and noise reduction, to enhance detection accuracy. YOLOv8 is then utilized to identify and track multiple vehicle categories, such as cars, trucks, buses, and motorbikes, while ensuring unique identifiers to avoid duplication across frames. A user-friendly Streamlit-based interface allows for seamless video upload, real-time visualization with bounding boxes, and the generation of structured detection logs containing object IDs, timestamps, and vehicle counts. Experimental results demonstrate that the proposed system achieves high accuracy and real-time performance, making it suitable for applications in traffic monitoring, congestion analysis, toll collection, and smart city development. The integration of deep learning, computer vision, and interactive deployment provides a scalable and cost-effective solution to modern traffic management challenges.*

Keywords: Vehicle Detection, Vehicle Counting, YOLOv8, Deep Learning, Object Tracking, Intelligent Transportation Systems, Smart Cities, Real-Time Traffic Monitoring.

I. INTRODUCTION

The exponential rise in vehicle ownership and rapid urbanization has created significant challenges for traffic regulation, road safety, and transportation infrastructure planning. Traditional vehicle counting methods, such as manual surveys, inductive loops, and infrared sensors, are often costly, labour-intensive, and lack adaptability to dynamic urban conditions. These limitations make it difficult to manage growing traffic volumes effectively and highlight the need for scalable, automated solutions.

Recent advancements in artificial intelligence (AI), particularly in computer vision, have provided new opportunities to address these challenges. Deep learning-based object detection models can extract meaningful features directly from raw images, outperforming conventional machine learning techniques that rely on handcrafted attributes. Among these, the You Only Look Once (YOLO) family of models has emerged as a highly effective solution for real- time vehicle detection due to its accuracy, speed, and ability to track multiple objects simultaneously. The latest version, YOLOv8, enhances detection robustness with improved tracking mechanisms, ID persistence, and efficient performance, making it suitable for deployment in real-world applications.

1.1 Problem Statement

The exponential increase in vehicles has made traffic regulation, monitoring, and planning increasingly difficult. Conventional methods such as manual counting, inductive loops, or sensor- based systems suffer from drawbacks like high costs, limited scalability, and lack of real-time adaptability. These shortcomings highlight the need for an automated, efficient, and intelligent solution for vehicle detection and counting. Inaccuracy and inefficiency of manual vehicle counting methods. High installation and maintenance costs of sensor-based systems. Inability of traditional

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 5, Issue 4, November 2025 Impact Factor: 7.67

methods to handle largescale real time data. Lack of flexible and lack of flexible and scalable solutions for modern traffic systems. To overcome these limitations, this project proposes a YOLO-based framework capable of delivering real-time, accurate, and cost-effective vehicle count detection.

1.2 Research Objectives

This research aims to design and implement a YOLOv8-based system for automatic detection and counting of vehicles in real-time. To develop a preprocessing pipeline that standardizes video data through resizing, grayscale conversion, and noise reduction. To integrate object tracking to ensure unique vehicle identifiers across frames, avoiding duplication. To evaluate system performance using metrics such as accuracy, precision, recall, F1-score, and mean Average Precision (mAP). To deploy the trained model through a Streamlit web application, offering real- time detection, visualization, and downloadable detection logs. To provide a scalable, cost- effective solution that supports traffic management, urban planning, and intelligent transportation systems.

1.3 Contributions

The primary contributions of this work are YOLOv8-Based Framework: Implementation of a deep learning model optimized for multi-class vehicle detection and counting. Preprocessing Pipeline: Standardization of input video frames through resizing, grayscale conversion, and Gaussian smoothing for enhanced model robustness. Integrated Tracking: Use of ID persistence to maintain consistent vehicle identifiers across frames, preventing duplicate counts. Comprehensive Evaluation: Validation of detection performance using standard metrics to ensure reliability in diverse traffic scenarios.

Deployment for Accessibility: Development of a Streamlit-powered web interface enabling real- time detection, visualization, and data export. Practical and Scalable Solution: A lightweight, efficient, and user-friendly system that bridges the gap between research and real-world deployment.

II. LITERATURE REVIEW

2.1 Traditional Vehicle Count Approaches

Historically, vehicle counting relied on manual surveys and sensor-based systems such as inductive loops, infrared sensors, and ultrasonic detectors. While these methods provided acceptable accuracy in controlled environments, they were often costly, required extensive maintenance, and lacked adaptability to large- scale deployments. Video-based classical algorithms, including background subtraction and motion tracking, offered partial automation but struggled with occlusion, varying lighting conditions, and high-density traffic.

2.2 Machine Learning Approaches

Before the advent of advanced deep learning models, traditional machine learning techniques were widely applied to vehicle detection. These methods used handcrafted features such as color, edges, or motion patterns, combined with classifiers like Support Vector Machines (SVM) and Decision Trees. While moderately successful, these models faced challenges in scalability, robustness, and adaptability, especially in complex urban environments.

2.3 Deep Learning Approaches

The introduction of deep learning revolutionized vehicle detection. Convolutional Neural Networks (CNNs) enabled automatic feature extraction and robust detection under diverse traffic conditions. The YOLO series (Redmon et al., 2016; Bochkovskiy et al., 2020; Jocher et al., 2022) advanced real-time detection capabilities by offering fast and accurate recognition of multiple vehicles per frame. Enhancements such as YOLOv4 and YOLOv8 further improved tracking and ID persistence, making them well- suited for dynamic traffic scenarios.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 4, November 2025
Impact Factor: 7.67

III. METHODOLOGY

3.1 Dataset and Data Preparation

This study utilizes publicly available traffic video datasets and curated vehicle footage collected from surveillance cameras and open repositories. The dataset contains multiple vehicle categories, including cars, buses, trucks, and motorbikes. To ensure fair evaluation, the dataset was divided into three subsets: training (70%), validation (15%), and testing (15%), while maintaining class balance.Each video was pre-processed into frames of fixed dimensions (640 × 480) for compatibility with the YOLOv8 model. Basic preprocessing techniques such as resizing, grayscale conversion, and Gaussian blur were applied to enhance input quality and reduce noise. Data augmentation techniques such as flipping, scaling, and brightness adjustment were employed to increase dataset diversity and ensure the model's robustness in varying lighting and traffic conditions.

3.2 Data Pre-processing Pipeline

Our preprocessing pipeline integrates multiple steps to prepare data for effective Yolov8.

3.2.1 Frame Standardization

- Conversion of video frames into RGB format.
- Resizing frames to 640 × 480 pixels.
- Normalization of pixel values for stable convergence.
- Removal of corrupted or duplicate frames.

3.2.2 Data Augmentation Techniques

To improve model generalization, the following augmentations were applied:

- Random horizontal and vertical flips.
- Cropping and zoom variations.
- Brightness and contrast adjustments.

3.3 Model Architecture Design

3.3.1 Backbone Network

The system employs YOLOv8, a state-of-the-art deep learning architecture design for real-time object detection. YOLOv8 leverages convolutional backbones optimized for efficiency and accuracy, enabling simultaneous detection and classification of multiple vehicles per frame. Pretained weights on the COCO dataset were used to initialize the network, providing transfer learning benefits for faster convergence.

3.3.2 Detection and Tracking Module

To adapt YOLOv8 for vehicle counting, the detection head was combined with a tracking module that assigns persistent IDs to each vehicle across frames. This ensures that vehicles are counted only once, even why they reappear in consecutive frames. The system supports detection of four major classes: car, truck, bus, and motorbike.

3.4 Training Pipeline Implementation

3.4.1 Optimization Strategy

The training process employs a composite loss function comprising CIoU-based bounding box regression loss, objectness loss, and classification loss to ensure precise localization and accurate object detection. Optimization is performed using Stochastic Gradient Descent (SGD) with momentum to enhance convergence stability and overall model performance. The batch size is set between 16 and 32, depending on the available GPU memory. Training typically spans 50 to 200 epochs, with early stopping applied based on validation performance to prevent overfitting and ensure optimal generalization.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 5, Issue 4, November 2025 Impact Factor: 7.67

3.4.2 Train-Validation-Test Split

The dataset was divided into three subsets to ensure robust model development and evaluation. The training set, comprising 70% of the data, was utilized to optimize the YOLOv8 model parameters. A validation set accounting for 15% of the data was employed for hyperparameter tuning and to mitigate overfitting during training. The remaining 15% constituted the testing set, which was used to assess the model's performance in an unbiased manner. This structured split ensured reliable performance while minimizing data leakage.

3.5 Evaluation Methodology

3.5.1 Detection Metrics

The performance of the system was assessed using standard object detection evaluation metrics. Mean Average Precision (mAP) was employed to measure overall detection accuracy across varying IoU thresholds. Precision was used to quantify the proportion of correctly detected vehicles, while recall measured the extent to which actual vehicles were successfully identified by the model. The F1-score, defined as the harmonic mean of precision and recall, provided a balanced evaluation of detection performance. Additionally, a confusion matrix was analyzed to examine class-wise detection accuracy and identify potential misclassification patterns.

IV. EXPERIMENTAL SETUP AND IMPLEMENTATION

4.1 Experimental Setup

4.1.1 Hardware Configuration

The experimental setup for system development and testing utilized a mid-range computing platform to ensure practical deployment feasibility. The processor was an Intel Core i7- 10750H running at 2.60 GHz, paired with 16 GB of DDR4 RAM to handle model training and real- time inference efficiently. A dedicated NVIDIA GTX 1650 Ti GPU with 4 GB memory was used to accelerate deep learning computations. For storage used 512 GB SSD.

4.1.2 Software Environment

The software environment for system implementation was carefully selected to support deep learning and computer vision workflows. The experiments were conducted on Windows 10, providing a stable platform for development. The deep learning framework consisted of Ultralytics YOLOv8, NumPy, Pandas, Matplotlib, Streamlit enabling efficient model building, training, and deployment. For computer vision tasks, OpenCV 4.5.5 was used for image processing operations, while MediaPipe 0.8.10 facilitated accurate vehicle detection. The entire system was developed using Python 3.10, offering flexibility, ease of integration, and a wide range of libraries suitable for machine learning and real-time application development.

4.2 Hyper parameter Configuration

The YoloV8-based model was fine-tuned with the following configuration:

Class	Precision	Recall	F1-Score	Support
Car	0.96	0.95	0.95	500
Truck	0.93	0.91	0.92	450
Bus	0.91	0.89	0.90	400
Motorbike	0.92	0.90	0.91	380

Fig 1: Sample Performance Metrics

4.3 Training Procedure Training Protocol:

The dataset was first divided into training, validation, and testing subsets to ensure structured model development and unbiased evaluation. All frames underwent a consistent preprocessing pipeline to improve data quality and enhance

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 5, Issue 4, November 2025 Impact Factor: 7.67

model learning. The YOLOv8 model was initialized using pretrained weights and subsequently fine-tuned on the prepared dataset for domain-specific adaptation. Throughout the training process, key performance indicators—including accuracy, mAP, and loss curves—were continuously monitored to track model convergence and stability. Additionally, early stopping was implemented to prevent overfitting and maintain optimal generalization performance.

V. RESULT AND ANALYSIS

5.1 Model Performance Overview

The proposed YOLOv8-based system achieved strong results, effectively detecting and counting vehicles across multiple categories. Data augmentation and preprocessing improved model robustness, while the tracking module ensured accurate counting without duplication.

5.2.2 Overall Model Performance

The overall performance of the proposed YOLOv8-based vehicle detection and counting system demonstrates strong accuracy and real- time capability. The model achieved a mean Average Precision (mAP) between 93% and 96%, indicating highly reliable detection performance across multiple IoU thresholds. The precision score of approximately 94% reflects the system's effectiveness in correctly identifying vehicles, while the recall value of around 91% shows its ability to detect the majority of actual vehicles present in the scene. The resulting F1-score of nearly 92% confirms a balanced trade-off between precision and recall. In addition to these accuracy metrics, the system maintained an inference speed of about 60 frames per second, confirming its suitability for real-time deployment in practical traffic monitoring and intelligent transportation applications.

5.3 Comparative Analysis

5.3.1 Model Comparison

Model	Accuracy	mAP	FPS	Deployment
Classical ML (SVM)	78%	-	Low	Limited
CNN (custom)	85%	80%	15	Moderate
YOLOv4	90%	88%	40	Moderate
YOLOV8	95%	93%	60	Practical

5.4 Training Dynamics Analysis

The training dynamics demonstrated strong and stable model performance throughout the fine- tuning process. The training accuracy consistently improved and reached approximately 97% within 20–25 epochs, indicating effective learning of vehicle features. The validation accuracy stabilized around 95%, reflecting minimal overfitting and strong generalization to unseen data. The loss curves exhibited smooth and steady convergence without any signs of fluctuation or divergence, confirming that the optimization process remained stable across epochs. Furthermore, early stopping was typically activated around the 22nd epoch, when validation performance plateaued, ensuring that the model achieved optimal accuracy while preventing unnecessary training and overfitting.

5.5 Error Analysis and Model Insights

5.5.1 Common Misclassification Pattern

Buses and Trucks due to their similar body structures, some confusion occurred in dense traffic conditions. In cases of occlusion or poor lighting, smaller vehicles like motorbikes were occasionally misclassified as cars.

5.5.2 Observations

Vehicle categories with distinct and easily distinguishable visual features—such as motorbikes and buses—demonstrated consistently high detection accuracy across the experiments. Their unique shapes, sizes, and motion patterns enabled the YOLOv8 model to identify them reliably even in moderately complex traffic scenes. However, certain challenging conditions, including overlapping vehicles and poor nighttime illumination, occasionally led to

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 5, Issue 4, November 2025 Impact Factor: 7.67

detection errors or reduced confidence scores. These observations highlight the importance of further expanding the dataset to include a broader range of challenging scenarios, such as dense traffic, adverse weather, and low- light environments, to enhance the model's robustness and generalization capability.

5.6 Computational Performance Analysis

5.6.1 Scalability Metrics and Deployment

The computational performance of the proposed YOLOv8-based system further highlights its suitability for real-time deployment in practical traffic monitoring scenarios. The overall training process required approximately 3–4 hours on a mid-range GPU, demonstrating efficient convergence even with moderate hardware resources. During inference, the model achieved an average processing speed of nearly 0.05 seconds per frame, corresponding to roughly 60 frames per second (FPS), thereby fully supporting real-time vehicle detection and counting applications. Memory consumption during training remained modest at around 2 GB, ensuring compatibility with commonly available GPU configurations. Additionally, the final trained model occupied only about 25 MB of storage, making it lightweight and easily deployable across diverse platforms, including edge devices, cloud systems, and web-based applications.

VI. DISCUSSION

6.1 Practical Implication

The results demonstrate that a deep learning– based YOLOv8 system can serve as an effective and scalable solution for vehicle detection and counting. With overall detection accuracy above 93% and real-time inference speeds, the framework is suitable for applications in traffic management, congestion monitoring, toll collection, and smart city infrastructure. The system reduces reliance on manual counting or expensive sensor-based solutions, making it a cost-effective alternative for urban authorities.

6.2 Model Architecture Insights

The YOLOv8 backbone, combined with integrated tracking, proved highly effective in extracting discriminative features and maintaining persistent vehicle IDs across frames. Distinct vehicle types (e.g., motorbikes) were classified accurately, while vehicles with similar dimensions (e.g., trucks and buses) posed occasional challenges. This suggests the potential for fine- grained feature extraction or advanced tracking mechanisms to improve differentiation.

6.3 Practical Deployment Considerations

The system demonstrates strong potential for real- world deployment, supported by its efficient performance and architectural design. It can be seamlessly integrated into existing traffic monitoring centers or deployed on cloud-based platforms, enabling large-scale and centralized processing of traffic video streams. With real-time inference capability of approximately 60 frames per second, the system ensures low latency, making it suitable for live traffic surveillance applications. Moreover, the framework supports continuous learning, where newly acquired traffic videos can be incorporated to enhance model adaptability and maintain performance across evolving traffic conditions. Since the system focuses solely on vehicle detection rather than identifying drivers or individuals, privacy risks are minimal; however, adherence to relevant surveillance regulations and data protection policies remains essential to ensure ethical and lawful deployment.

VII. FUTURE WORK

7.1 Technical Enhancements

The future scope of this work includes several technical advancements aimed at further enhancing system accuracy, interpretability, and deployability. First, incorporating advanced architectures such as transformer-based detection models—including Vision Transformers and DETR—could significantly improve contextual understanding and long-range feature representation in complex traffic environments. Additionally, integrating visual attention mechanisms would enable the model to focus more effectively on critical vehicle regions, thereby improving both detection robustness and interpretability. From a deployment perspective, optimizing the model for edge devices through

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 5, Issue 4, November 2025 Impact Factor: 7.67

techniques such as pruning and quantization would reduce computational overhead and model size, facilitating real-time inference on resource- constrained embedded platforms such as the NVIDIA Jetson Nano and Raspberry Pi. These enhancements collectively offer promising directions for developing more efficient, scalable, and context-aware vehicle detection systems.

7.2 Dataset and Evaluation Improvements

Future enhancements to the system can be achieved by expanding the dataset to include a wider range of traffic conditions, particularly challenging scenarios such as night-time, fog, and rain, which would improve model robustness under real-world environmental variations.

Additionally, synthetic data generation using generative models can be employed to enrich underrepresented vehicle categories, such as bicycles and rickshaws, thereby enhancing class balance and improving detection accuracy for minority classes. Furthermore, cross-domain testing across diverse geographical regions and varying traffic environments will enable a comprehensive evaluation of the model's generalizability, ensuring that the system performs reliably beyond the conditions of the training dataset.

VIII. CONCLUSION

This research presents a real-time vehicle detection and counting framework based on the YOLOv8 deep learning model. The proposed system achieved ~95% detection accuracy and maintained real-time performance (~60 FPS), surpassing traditional methods and earlier YOLO versions. Application of YOLOv8 with integrated tracking for accurate and persistent vehicle counting. Standardized preprocessing and augmentation to ensure robustness under varied traffic conditions. Significant improvements in accuracy, precision, recall, and F1-scores compared to baseline methods. A streamlit based application for real-time traffic analysis, enabling practical use beyond research prototypes.

While challenges remain particularly in handling dense traffic occlusion and low-light conditions this study establishes a strong foundation for automated, scalable traffic monitoring systems. Future enhancements such as advanced detection models, edge deployment, and expand datasets will further strengthen its applicability in intelligent transportation systems and smart cities.

REFERENCES

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

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

[3] G. Jocher, et al., "YOLOv5 and YOLOv8: Advancements in Real-Time Object Detection," Ultralytics Open Source Release, 2022.

[4] J. W. Hsieh, L. C. Chen, D. Y. Chen, and W. Y. Chiu, "Automatic Vehicle Counting via Vehicle Detection and Tracking," IEEE Transactions on Intelligent Transportation Systems, vol. 21, no. 1, pp. 1–10, 2021.

[5] X. Li, Y. Zhang, and K. Liu, "Deep Learning- Based Vehicle Detection and Counting for Toll Plaza Surveillance," Journal of Advanced Transportation, 2022.

[6] K. Prasanna, R. Aravind, and A. Kumar, "Vehicle Detection and Counting in Traffic Scenes Using Support Vector Machines," International Journal of Computer Applications, vol. 135, no. 7, pp. 1–6, 2016.

[7] R. Aravind, et al., "Deep Learning-Based Vehicle Detection and Traffic Flow Analysis Using CNNs," International Journal of Intelligent Transportation Systems Research, vol. 17, no. 2,pp. 95–104, 2019.

[8] S. Chowdhury, M. Rahman, and T. Ali, "Vehicle Detection and Counting Using YOLO and Deep Learning Approaches," ResearchGate Preprint, 2021.

[9] O. Russakovsky, et al., "ImageNet Large Scale Visual Recognition Challenge," International Journal of Computer Vision, vol. 115, no. 3, pp. 211–252, 2015.

[10] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. MIT Press, 2016.