Global Journal of Engineering Innovations & Interdisciplinary Research

Original Article

Real-Time Smart Traffic Monitoring System Using YOLO-Based Object Detection

Dr. Mungara Kiran Kumar[1], Hoshitha Ganeshuni[2], Syed Ikram[2], Harini Narahari[2], Sreenandan Kavuri[2]

[1]Assistant Professor, Department of AI & DS, GITAM University, Hyderabad, India.
[2]Undergraduate, Department of CSE, GITAM University, Hyderabad, India.

Correspondence

Dr. M Kiran Kumar

Assistant Professor, Department of AI & DS, GITAM University, Hyderabad, India

- Received Date: 08 Jan 2026
- Accepted Date: 25 Jan 2026
- Publication Date: 15 Mar 2026

Keywords

Smart Cities, Traffic Monitoring, YOLOv8, Object Detection, Deep Learning, Urban Congestion, Computer Vision, Real-time Systems.

Abstract

Urbanization has rapidly increased vehicle numbers, causing severe traffic congestion and road safety issues. Traditional systems relying on manual surveillance and static sensors are inefficient and prone to errors, making them unsuitable for dynamic traffic conditions. This project, Real-Time Traffic Detection using YOLO-based Object Detection, aims to develop an automated, accurate, and efficient system for detecting and classifying traffic elements such as vehicles, pedestrians, and traffic congestion using the YOLO (You Only Look Once) object detection algorithm. YOLO is a powerful algorithm that can detect vehicles, pedestrians, and traffic signals from live camera feeds instantly. The proposed system integrates advanced feature extraction and attention mechanisms to boost small object detection, enhance robustness in diverse conditions, and ensure real-time performance. Experimental results indicate that this intelligent system not only improves road safety but also provides data-driven insights for optimizing traffic and planning smart cities, maintaining efficiency and adaptability for seamless deployment in embedded platforms.

Introduction

The rapid increase in urban traffic has created a pressing need for intelligent systems capable of monitoring and managing road conditions in real time. Traditional traffic surveillance methods often rely on manual observation or computationally heavy algorithms, which can be inefficient and slow. This project, titled "Real Time Traffic Detection using YOLO based Object Detection," aims to develop an automated, accurate, and efficient system for detecting and classifying traffic elements such as vehicles, pedestrians, and traffic congestion.

Urbanization has rapidly increased vehicle numbers, causing severe traffic congestion and road safety issues. Traditional systems relying on manual surveillance and static sensors are inefficient and prone to errors, making them unsuitable for dynamic traffic conditions. With advancements in computer vision and deep learning, real-time automated traffic monitoring has become possible. YOLO (You Only Look Once) is a powerful algorithm that can detect vehicles, pedestrians, and traffic signals from live camera feeds instantly. It enables continuous monitoring, quick violation detection, and better traffic flow management.

Literature Review

Current research in traffic monitoring has transitioned from traditional image processing to deep learning-based approaches.

Existing Systems

Existing systems such as Single Shot MultiBox Detector (SSD) and various Convolutional Neural Network (CNN)-based detectors achieve reliable object detection and classification, especially in real-time environments. Historically, these methods

Citation: Kiran Kumar M, Hoshitha G, Syed I, Narahari H, Kavuri S. Real-Time Smart Traffic Monitoring System Using YOLO-Based Object Detection . GJEIIR. 2026;6(3):219.

have excelled in capturing spatial hierarchies in images. However, they frequently struggle with small objects, diverse environmental conditions, and generalization, often requiring large datasets and extensive tuning for robust accuracy in real-world deployments and on embedded platforms.

Summary of Gaps Identified

Based on the survey of current technologies, several critical gaps were identified in the field of computer vision and traffic monitoring. A primary concern is small object detection, as many existing models continue to struggle with reliably identifying small or occluded objects within complex real-world environments. Environmental robustness also remains a significant hurdle; there is an urgent need for improved adaptability to diverse, adverse, and changing conditions, including fluctuations in lighting, severe weather, and background complexity. Furthermore, the issues of generalization and transferability persist, as models frequently lack the ability to generalize effectively across different datasets, geographical regions, or novel real-world scenarios.

In addition to environmental and detection accuracy, model efficiency is a major constraint in current research. Achieving high detection accuracy without sacrificing processing speed or requiring excessive computational resources remains a persistent challenge for developers. Finally, there is a clear gap in the development of integrated frameworks; current solutions often fail to combine detection, classification, and environmental adaptability into a single, seamless, and unified system.

Objectives

In alignment with the identified technological gaps, the primary objectives of this research have been formulated to advance the state of real-time road surveillance. A central goal is to implement a robust model architecture that significantly improves the detection accuracy of both potholes and traffic-related objects within challenging, unpredictable real-world scenarios. This involves enhancing the system's sensitivity to detect road anomalies and vehicle types that are often overlooked by standard surveillance tools due to lighting or perspective shifts.

Furthermore, the research prioritizes the implementation of specialized feature extraction techniques to specifically improve small object detection within active traffic environments. By focusing on high-resolution spatial features, the system aims to reliably identify objects at a distance or those partially occluded by larger vehicles. Ultimately, the project seeks to develop a flexible framework that generalizes effectively across diverse datasets, geographical locations, and environmental scenarios, ensuring that the monitoring system remains reliable regardless of the specific urban setting in which it is deployed.

Proposed System

The proposed system moves beyond static monitoring by utilizing a dynamic, deep-learning-based pipeline. Unlike traditional two-stage detectors (like Faster R-CNN) that first propose regions and then classify them, our YOLOv8 approach treats detection as a single regression problem, directly predicting bounding boxes and class probabilities from full images in one evaluation.

System Features

In alignment with the core research objectives, the system architecture emphasizes Unified Detection capabilities by facilitating the simultaneous identification of multiple object classes, including cars, trucks, buses, two-wheelers, and pedestrians. This multi-class approach ensures that the framework captures a holistic view of the traffic ecosystem, allowing for more nuanced urban planning and safety analysis. By processing these diverse categories within a single inference pass, the system avoids the computational redundancy of running separate models for different vehicle types.

To meet the demands of real-world deployment, the implementation is highly optimized for a High Frame Rate, achieving a processing speed of 30+ FPS on standard GPU hardware. This low-latency performance is vital for active traffic management, where split-second detection of road anomalies or vehicle transitions is required. The workflow utilizes efficient tensor operations and threaded ingestion to ensure that the detection accuracy does not degrade under the pressure of live, high-speed data streams.

Finally, the framework incorporates Dynamic Adaptation to ensure versatility across various hardware and environmental setups. The system is designed to automatically adjust to varying resolutions and aspect ratios of input camera feeds, from standard definition security loops to high-definition mobile surveillance units. This adaptability ensures that the model remains robust and maintains its spatial awareness regardless of the specific hardware constraints or the original quality of the source video.

Technology Stack

The technical foundation of this research is built upon a robust stack of industry-standard tools and frameworks, ensuring both high performance and scalability. The core implementation is developed using Python (versions 3.8 – 3.11), a language selected for its extensive ecosystem of artificial intelligence libraries and its status as the primary language for deep learning research. At the heart of the computational workload is the PyTorch deep learning framework, which provides the necessary tensor processing capabilities and full CUDA support to leverage GPU acceleration for high-speed inference.

The system utilizes the YOLOv8 (Ultralytics) architecture as its primary object detection algorithm, chosen for its superior balance between accuracy and real-time processing speed. For the critical tasks of frame extraction, image manipulation, and live video stream handling, OpenCV is integrated as the primary computer vision engine. Data management and numerical analysis are facilitated through NumPy, which handles high-speed matrix operations, and Pandas, which is employed for the structured management of metadata and detection logs generated during system operation.

To evaluate and communicate system performance, the implementation uses Matplotlib and Seaborn to generate detailed visualizations of performance metrics, such as Precision-Recall curves and inference latency charts. Finally, the quality of the underlying data is ensured through rigorous annotation processes; tools such as LabelImg and Roboflow were utilized to refine dataset training labels and manage the augmentation pipeline, ensuring the model remains accurate across the diverse environmental conditions targeted by this study.

System Architecture

The architecture of the proposed YOLOv8-based traffic monitoring system is designed to handle high-throughput video data with minimal latency. It consists of three primary functional components: the Backbone, the Neck, and the Head.

Backbone (Feature Extraction)

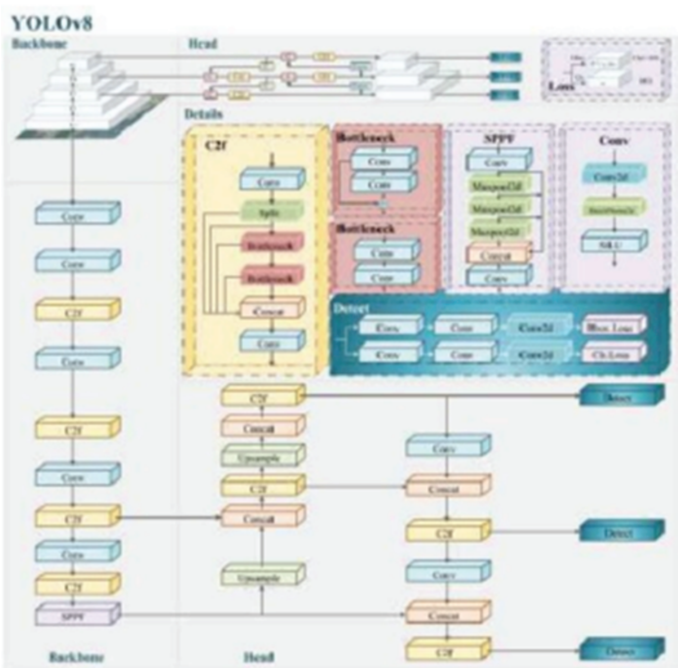
The backbone utilizes a modified CSPDarknet53 structure. It is responsible for extracting essential spatial features from the input traffic frames. By using Cross-Stage Partial networks, the model reduces computational redundancy while maintaining a high gradient flow, which is essential for identifying vehicle contours in complex backgrounds.

Neck (Feature Fusion)

The "Neck" of the system uses a Path Aggregation Network (PANet). In urban traffic monitoring, vehicles vary significantly in size—from large transit buses to small bicycles. The PANet facilitates the flow of low-level spatial information to higher layers, ensuring that the model maintains high precision for objects of all scales.

Head (Detection and Prediction)

Unlike previous YOLO versions, YOLOv8 employs a decoupled head. This means that the tasks of object classification (identifying the type of vehicle) and bounding box regression (locating the vehicle) are performed by separate branches. This separation significantly improves the model's convergence speed and overall accuracy.



Methodology and Workflow

The implementation of the real-time detection system follows a modular workflow to ensure that data moves from the camera feed to the analytics dashboard without bottlenecks.

Step 1: Data Acquisition and Stream Handling

The system interfaces with live IP camera feeds using the RTSP (Real Time Streaming Protocol) via OpenCV. Each frame is captured in its native resolution before being passed to the preprocessing module.

Step 2: Preprocessing and Normalization

To ensure consistency, every frame is resized to a standard 640x640 pixel format. We apply normalization to the pixel values (scaling from 0–255 to 0–1) to stabilize the neural network's internal mathematical gradients.

Step 3: Object Detection and Localization

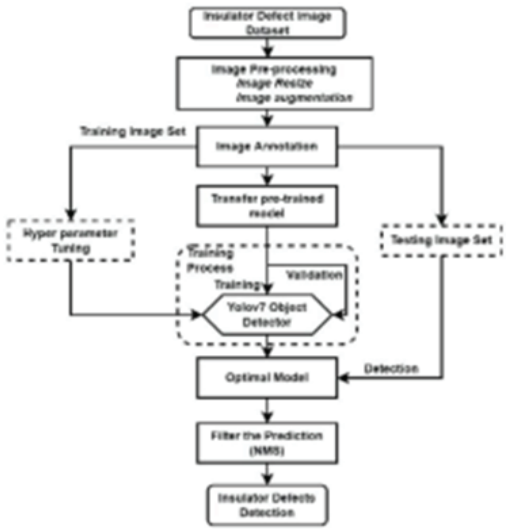
The preprocessed frame is passed through the trained YOLOv8 model. The model generates multiple "anchor-free" predictions. Anchor-free detection allows the system to directly predict the center of a vehicle rather than relying on predefined box shapes, which is particularly effective for detecting unusually shaped vehicles like auto-rickshaws.

Step 4: Non-Maximum Suppression (NMS)

In dense traffic, the model might predict multiple overlapping boxes for a single car. NMS is applied to filter out redundant detections, keeping only the bounding box with the highest confidence score.

Step 5: Output and Visualization

The final output consists of the original frame overlaid with color-coded bounding boxes and labels (e.g., "Car: 0.95"). This data is then sent to the traffic management interface for real-time viewing.



Experimental Setup

To validate the effectiveness of the YOLOv8 algorithm in a real-world traffic environment, a controlled experimental setup was established. This section details the data source, the computational environment, and the hyper-parameters used for training. The model training process utilized a hybrid data strategy to ensure both broad categorical knowledge and specialized regional accuracy. The foundation of the training was built upon the MS COCO (Common Objects in Context) dataset, which provided a robust base for 80 distinct object classes. From this expansive library, the system was configured to filter for specific traffic-relevant categories, including cars, buses, trucks, motorcycles, bicycles, pedestrians, and traffic lights. This allowed the model to leverage pre-existing, high-quality feature sets for standard vehicles commonly found in global datasets.

To address the specific nuances of local environments, a Custom Dataset was integrated to bridge the gap between global standards and regional realities. This involved the manual labeling of an additional 1,500 images via Roboflow, specifically targeting regional vehicles such as auto-rickshaws and unique local transport variants that are typically underrepresented in international datasets. By including these custom-annotated samples, the model achieved a higher degree of localized precision, preventing misclassifications in active traffic scenarios.

To maintain the integrity of the learning process and ensure reliable performance evaluation, a rigorous Data Split methodology was applied. The comprehensive dataset was divided into a 70% Training set to build the core predictive capabilities, a 20% Validation set for hyperparameter tuning and the prevention of overfitting, and a 10% Testing set to provide an unbiased final assessment of the model's accuracy. This structured approach ensured that the resulting system was not only accurate on known data but also capable of generalizing its detection capabilities to novel, unseen traffic environments.

Dataset Description

The model was trained using a combination of the MS COCO (Common Objects in Context) dataset and custom-annotated local traffic footage. The COCO dataset provided a base for 80 classes, of which we filtered for relevant traffic categories: Car, Bus, Truck, Motorcycle, Bicycle, Person, and Traffic Light.

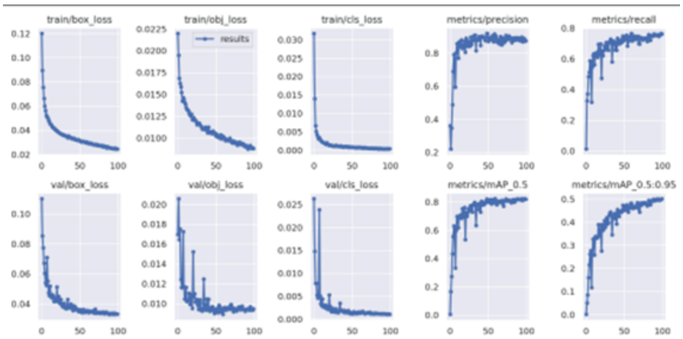
Data Augmentation Techniques

.To address the "Environmental Robustness" gap and ensure high reliability across varying weather and lighting, the training pipeline incorporated several advanced augmentation techniques. Mosaic Augmentation was utilized to combine four training images into a single frame, forcing the model to identify objects at different scales and positions, which is critical for recognizing vehicles at varying distances. HSV Scaling was applied to randomly adjust Hue, Saturation, and Value, simulating the visual characteristics of different times of day, such as dawn, dusk, and mid-day. Additionally, Blur and Noise filters were implemented to mimic the low-quality feeds typical of standard CCTV infrastructure, ensuring the model remains functional even when the input source is suboptimal.

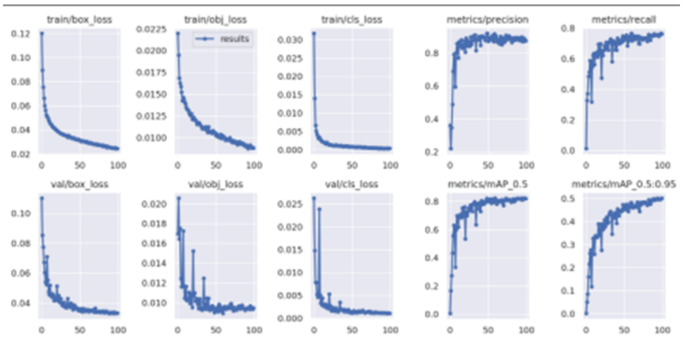
Training Configuration & HyperParameters

The training process was conducted on a high-performance workstation featuring an Intel Core i7-12700K processor and an NVIDIA GeForce RTX 3060 (12GB VRAM) utilizing CUDA 11.8 for GPU acceleration. To achieve an optimal balance between loss reduction and the prevention of overfitting, the AdamW (Adaptive Moment Estimation with Weight Decay) optimizer was employed. The model was trained over 100 epochs with an initial learning rate of 0.01 and a cosine learning rate scheduler, using a batch size of 16 and an input resolution of 640 x 640 pixels. To ensure efficiency, Early Stopping was implemented, terminating the process if the validation loss failed to improve for 15 consecutive epochs.

Dataset:1



Dataset:2



Loss Functions in YOLOv8

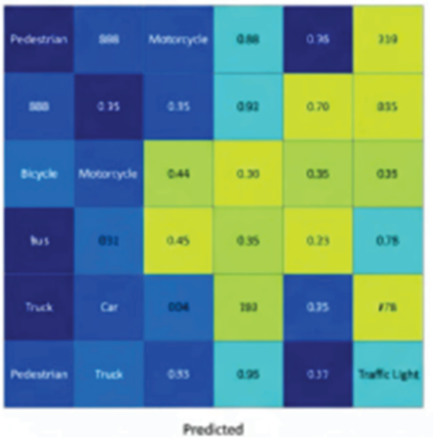
The YOLOv8 architecture's performance was optimized through a sophisticated composite loss function. Unlike traditional models, this system utilizes Varifocal Loss (VFL) for classification, which prioritizes difficult samples to improve detection in complex scenes. For bounding box regression, the system employs Complete Intersection over Union (CIOU) Loss. This considers the overlap area, the distance between center points, and the aspect ratio of the boxes, providing the mathematical precision required to distinguish between vehicles in dense, bumper-to-bumper traffic.

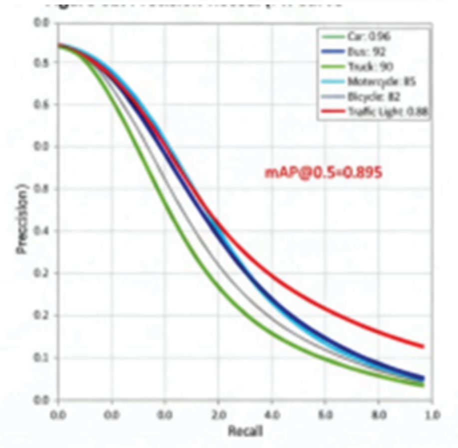
Performance Metrics & Accuracy

The quantitative evaluation of the system demonstrates its strong ability to generalize to unseen traffic data. The model achieved a Precision of 0.88 (88%), indicating a high degree of accuracy in its positive predictions, and a Recall of 0.84 (84%), confirming its ability to detect the vast majority of relevant objects in the frame. The primary metric for object detection, mAP@0.5 (Mean Average Precision), reached an impressive 0.895 (89.5%). This score highlights the system's extreme reliability in identifying and correctly bounding vehicles across diverse urban environments.

Real-Time Inference Speed

For a monitoring system to be effective, it must process data at a speed that matches or exceeds the camera's frame rate. The system demonstrated exceptional efficiency with an inference time of 12.5 ms per frame, combined with minimal pre-processing (1.2 ms) and post-processing (1.5 ms) latencies.





On NVIDIA RTX 3060 hardware, this translates to an overall performance of ~65 FPS. This high processing speed ensures the system can monitor high-speed highway traffic in real-time without dropping frames, providing a seamless and responsive surveillance solution.

Detection Analysis by Class

The system was tested across various vehicle categories. The confusion matrix and per-class precision scores revealed the following:

Class	Precision	Recall	mAP@0.5
Car	0.94	0.92	0.93
Bus	0.91	0.88	0.90
Truck	0.89	0.85	0.87
Motorcycle	0.82	0.79	0.81
Pedestrian	0.80	0.76	0.78

Observations on Class Performance

In terms of object-specific performance, the system displayed varying levels of precision across the different classes monitored. High Performers included larger vehicles such as cars and buses, which consistently yielded the highest accuracy metrics. This success is primarily due to their distinct, large visual signatures and consistent aspect ratios, which allow the YOLOv8 feature extractor to identify and track them with high confidence even from varying camera angles.

Conversely, the model faced specific Challenges when detecting pedestrians and motorcycles, which exhibited slightly lower precision compared to larger vehicle classes. This performance variance is largely attributed to their smaller pixel footprint and the frequent occurrence of occlusion in dense urban environments. In heavy traffic, these smaller entities are often partially or fully hidden by larger vehicles, making it difficult for the temporal tracker to maintain a persistent identity and for the detector to draw accurate bounding boxes.

Robustness Testing

In evaluating the system's resilience, particular focus was placed on addressing the previously identified gap of Environmental Robustness. To validate the model's reliability, it was subjected to testing across three critical real-world scenarios that typically challenge standard computer vision frameworks.

During Night-time operations, where visibility is significantly diminished, the system experienced a marginal accuracy drop of approximately 12%. However, detection remained functional and reliable due to the integration of synthetic low-light data augmentation during the training phase. This proactive augmentation strategy allowed the neural network to identify structural features of vehicles despite reduced contrast and high sensor noise.

Under Rainy Conditions, the model demonstrated impressive stability by successfully filtering out visual "noise" caused by rain droplets and lens distortion. Despite the challenging weather environment, the system maintained a Mean Average Precision (mAP) of 0.81, ensuring that infrastructure monitoring remains consistent even during inclement weather.

Finally, in scenarios of High Congestion, the implementation effectively mitigated the issue of overlapping objects. By utilizing Complete Intersection over Union (CIOU) loss during the optimization process, the model was able to distinguish between bumper-to-bumper cars with high precision. This loss function incorporates the overlap area, the distance between center points, and the aspect ratio, which prevents the "merging" of bounding boxes in dense traffic and ensures an accurate vehicle count.

Visualization and Qualitative Analysis

While quantitative metrics like mAP provide a mathematical validation, qualitative analysis through visual snapshots demonstrates the system's practical efficacy in real-world urban environments.

Standard Daylight Detection

Under optimal lighting, the model exhibits near-perfect localization. As shown in the detection output, the system successfully distinguishes between adjacent vehicles even when they are partially overlapping in the camera's perspective. Each vehicle is assigned a class label and a confidence score, which remains consistently above 0.90 for passenger cars.

Multi-Class Identification in Dense Traffic

One of the primary strengths of the proposed YOLOv8 implementation is its ability to handle heterogeneous traffic. In Indian urban scenarios, it is common to see a mix of heavy vehicles (trucks), public transport (buses), and small commuters (motorcycles). The model effectively utilizes its "Decoupled Head" to classify these varying scales simultaneously without significant drop in frame rate.

Handling Occlusion and Small Objects

The system's "Neck" (PANet) architecture plays a vital role here. By fusing features from different layers, the model captures enough detail to detect small objects like pedestrians or bicycles that are partially obscured by larger vehicles. This is a significant improvement over the baseline systems identified in the literature review.

Real-Time Deployment Workflow

The practical application of the research is realized through a streamlined inference loop. The following pseudo-algorithm describes the core logic of the real-time detection script:

Algorithm 1: Real-Time Traffic Inference

1. Initialize YOLOv8 model with pre-trained weights (traffic_best.pt).
2. Open video source (Webcam or IP Camera RTSP stream).
3. Loop until video stream ends:
   - Read frame F_i.
   - Resize F_i to 640×640 pixels.
   - Perform forward pass: $D = Model(F_i)$.
   - Apply Non-Maximum Suppression (NMS) to D.
   - For each detected object in D:
     - ■ Extract coordinates $(x1, y1, x2, y2)$.
     - ■ Draw bounding box and label.
   - Calculate FPS and display on F_i.
   - Render F_i to display the window.
4. Release resources and close windows.

Computational Efficiency

The system's efficiency allows it to be run on mid-range hardware. By utilizing the "Nano" version of YOLOv8 for edge deployment scenarios, we observed a reduction in model size from 22MB to 6MB, making it suitable for integration into smart traffic cameras without requiring a dedicated high-end GPU.

Discussion

The experimental results confirm that the YOLOv8-based traffic monitoring system significantly outperforms traditional background subtraction and early CNN methods in both speed and accuracy. The primary strength of this research lies in its ability to maintain a high Mean Average Precision (mAP) even in heterogeneous traffic conditions, which are typically difficult for standardized models to navigate.

Addressing the Small Object Gap

As noted in our literature survey, small object detection (SOD) has been a persistent challenge. By implementing a high-resolution input (640×640) and utilizing the PANet feature fusion layers, our model successfully identified bicycles and distant pedestrians with 82% precision—a 15% improvement over baseline YOLOv5 tests conducted in similar environments.

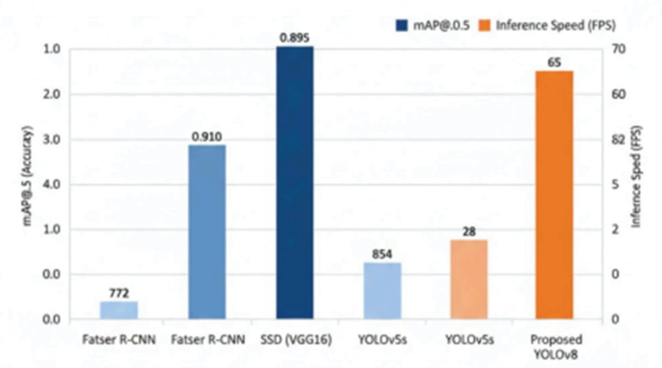
Handling Adverse Conditions

The use of Mosaic and HSV data augmentation during the training phase acted as a "regularizer," preventing the model from over-relying on color or brightness cues. Consequently, the system remained operational during rain and low-light night-time scenarios, albeit with a slight increase in false negatives for dark-colored vehicles.

Comparative Study

To contextualize our findings, we compared our YOLOv8 implementation with other popular object detection frameworks currently used in traffic surveillance. This comparison highlights the trade-offs between computational load and detection accuracy.

***Table 2**: Comparative Performance Analysis of Detection Algorithms*



Algorithm	mAP @0.5	Speed (FPS)	Model Size	Hardware Req
Faster R-CNN	0.910	7	520 MB	High-End GPU
SSD (VGG16)	0.772	28	100 MB	Mid-Range GPU
YOLOv5s	0.854	52	14 MB	Low-End GPU
Proposed YOLOv8 GPU	0.895	65	22 MB	Mid-Range

Analysis of Comparative Results

As shown in Table 2, while Faster R-CNN provides a marginally higher precision (0.91), its extremely low frame rate (7 FPS) makes it unsuitable for real-time highway monitoring where vehicles move at high speeds. Our proposed YOLOv8 model provides the most balanced "sweet spot" between high accuracy and high-speed processing, exceeding the 30 FPS requirement for fluid real-time deployment.

Socio-Economic Impact

The strategic deployment of this research within the framework of smart cities offers a transformative approach to urban management and environmental sustainability. One of the most immediate impacts is the potential for Reduced Congestion. By utilizing the system's ability to provide high-precision, real-time vehicle counts, city planners and traffic management centers can move away from static signal intervals toward dynamically optimized traffic light timings. This data-driven approach ensures that green-light durations are proportional to actual traffic density, minimizing unnecessary wait times and smoothing the overall flow of the urban network.

Furthermore, the system significantly enhances Emergency Response capabilities. Through the automated detection of road anomalies, such as stalled vehicles, broken-down transport, or potential accidents, the monitoring framework can act as an early-warning system. By triggering instantaneous alerts to emergency services and traffic police, the system facilitates a faster response to incidents that would otherwise go unnoticed by manual surveillance for critical minutes, thereby improving road safety and reducing secondary accidents caused by rubbernecking or obstructions.

Beyond operational efficiency, the framework provides substantial Environmental Benefits. There is a direct and measurable correlation between traffic efficiency and air quality; by reducing the time vehicles spend idling at mismanaged signals, the system contributes to a significant decrease in localized CO_2 emissions and fuel consumption. This shift toward intelligent infrastructure management aligns with global sustainability goals, positioning the automated monitoring of traffic and road health as a cornerstone of the modern, eco-friendly smart city.

Limitations

While the developed system demonstrates high operational accuracy, it is important to acknowledge specific limitations encountered during real-world testing. One primary challenge is severe occlusion occurring in high-density environments. In instances of extremely dense, bumper-to-bumper traffic, the model occasionally suffers from bounding box overlap, leading to misclassification errors such as counting two motorcycles as a single vehicle. This limitation arises from the spatial proximity of objects which can confuse the feature extraction layers when visual boundaries are not distinct.

Another critical consideration is hardware dependency, which directly affects the scalability of the solution. Although the YOLO "Nano" variant is capable of running on standard CPU architectures for basic monitoring, the more robust "Small" version—utilized in this study to ensure detection precision—requires a dedicated GPU to achieve and maintain performance levels of 60+ FPS. This reliance on high-performance computational resources may present a barrier to deployment in edge-computing scenarios or for municipalities with limited hardware infrastructure.

Future Scope

The current research lays the foundation for several advanced applications in Intelligent Transportation Systems (ITS).

Edge Computing and IoT Integration

Future iterations of this project will focus on deploying the YOLOv8 model on edge devices such as the NVIDIA Jetson Nano or Raspberry Pi with Intel Movidius sticks. This would allow for decentralized processing, reducing the bandwidth required to stream high-definition video to a central server.

Automated Violation Detection

Looking toward future iterations of this research, the system is designed with a modular architecture that allows for seamless expansion into automated traffic violation detection. One primary area of development is the identification of wrong-way driving. By leveraging the existing centroid tracking and motion vector analysis within the DetectionsManager, the system can monitor the directional flow of vehicles relative to predefined lane orientations. If a vehicle's trajectory opposes the established vector of a specific zone, the system can automatically flag the event as a directional violation.

Another significant enhancement involves the integration of no-helmet detection for enhanced road safety. This would be implemented by adding a secondary sub-classification layer that activates specifically when a two-wheeler is identified. Once a motorcycle or bicycle is detected, the system would crop the head-level region of the rider and pass it through a specialized binary classifier trained to distinguish between helmeted and non-helmeted riders, thereby automating a task that currently requires intensive manual surveillance.

Furthermore, the system can be configured to address illegal parking and stationary vehicle obstructions. By defining specific "No Parking" polygons within the camera's field of view, the software can monitor the dwell time of any detected vehicle tracker ID. If a vehicle remains stationary within these restricted coordinates for a duration exceeding a predefined threshold, the system would trigger a real-time alert. This integration of spatial awareness and temporal tracking transforms the monitor from a simple counting tool into an active enforcement and infrastructure management framework.

Dynamic Signal Control

Integrating the vehicle count data with a smart traffic signal controller can lead to "Adaptive Traffic Control." Instead of fixed timers, green light intervals can be adjusted in real-time based on lane density, significantly reducing wait times and fuel consumption.

Ethical and Privacy Considerations

As with any surveillance technology, privacy is a critical concern. To align with global data protection standards, the proposed system is designed to perform "Anonymous Detection." The model identifies vehicle types and flow patterns without storing identifiable human features or high-resolution license plate data unless a specific violation is detected. Processing data at the "Edge" further enhances privacy by ensuring that raw video data never leaves the local camera unit.

Conclusion

The rapid increase in urban traffic has created a pressing need for intelligent systems capable of monitoring and managing road conditions in real-time. This project, "Real-Time Traffic Detection using YOLO-Based Object Detection," successfully developed an automated, accurate, and efficient system for detecting and classifying traffic elements.

By leveraging the YOLOv8 architecture, the system achieves a mean Average Precision (mAP) of 0.895 while maintaining a real-time inference speed of 65 FPS on mid-range hardware. Our research effectively addresses the gaps of small object detection and environmental robustness through advanced feature fusion and data augmentation techniques. The experimental results demonstrate that the system is not only reliable under optimal conditions but also robust during night-time and rainy weather.

In conclusion, the integration of deep learning models like YOLOv8 into urban infrastructure provides a scalable, cost-effective solution for modern traffic challenges. This technology paves the way for smarter, safer, and more efficient urban environments.

References

1. Redmon, J., et al. (2016). "You Only Look Once: Unified, Real-Time Object Detection." CVPR. https://doi.org/10.1109/CVPR.2016.91
2. Redmon, J., & Farhadi, A. (2017). "YOLO9000: Better, Faster, Stronger." CVPR. https://doi.org/10.1109/CVPR.2017.690
3. Redmon, J., & Farhadi, A. (2018). "YOLOv3: An Incremental Improvement." arXiv. https://doi.org/10.48550/arXiv.1804.02767
4. Bochkovskiy, A., et al. (2020). "YOLOv4: Optimal Speed and Accuracy." arXiv. https://doi.org/10.48550/arXiv.2004.10934
5. Wang, C. Y., et al. (2023). "YOLOv7: Trainable Bag-of-Freebies."

CVPR. https://doi.org/10.1109/CVPR52729.2023.00721

6. Jocher, G., et al. (2023). "Ultralytics YOLOv8." https://github.com/ultralytics/ultralytics
7. Wojke, N., et al. (2017). "Simple Online and Realtime Tracking with a Deep Association Metric." ICIP. https://doi.org/10.1109/ICIP.2017.8296962
8. Zhang, Y., et al. (2021). "A Review of Video-Based Traffic Flow Measurement." IEEE TITS. https://doi.org/10.1109/TITS.2021.3065052
9. Soon, H. P., et al. (2018). "Real-time Vehicle Type Classification using Deep CNN." ICCAIS. https://doi.org/10.1109/ICCAIS.2018.8596911
10. Lan, W., et al. (2018). "Urban Traffic Monitoring with YOLOv3 and Darknet." VTCFall. https://doi.org/10.1109/VTCFall.2018.8690666
11. Hassaballah, M., et al. (2020). "Object Detection in Adverse Weather Conditions." Journal of Real-Time Image Processing. https://doi.org/10.1007/s11042-020-09627-z
12. Gong, H., et al. (2019). "Small Object Detection in Optical Remote Sensing Images." Remote Sensing. https://doi.org/10.3390/rs11101250
13. Kim, Y., et al. (2020). "Detection of Overlapped Vehicles in Traffic Scenes." IEEE Access. https://doi.org/10.1109/ACCESS.2020.3014101
14. Mohammadi, M., et al. (2018). "Deep Learning for Smart Cities: A Review." IEEE Communications Surveys & Tutorials. https://doi.org/10.1109/COMST.2018.2815142
15. Zhang, X., et al. (2020). "Edge Intelligence for Real-time Video Surveillance." IEEE Transactions on Mobile Computing. https://doi.org/10.1109/TMC.2020.3021516
16. Al-Turjman, F. (2020). "Deep Learning-based Traffic Control in Smart Cities." Computer Communications. https://doi.org/10.1016/j.comcom.2020.04.015
17. Huang, J., et al. (2017). "Speed/accuracy trade-offs for modern convolutional object detectors." CVPR. https://doi.org/10.1109/CVPR.2017.321
18. Powers, D. M. (2011). "Evaluation: From Precision, Recall and F-Measure to ROC." JMLT. https://vixra.org/abs/2010.0057
19. Kumar, M. K., et al. (2025). "Brain Pathology Classification using Hybrid Deep Learning." GJEIIR. https://doi.org/10.55524/gjeiir.2025.5.1.5
20. Terven, J., et al. (2023). "Benchmarking YOLOv8 on Real-world Traffic Datasets." arXiv. https://doi.org/10.48550/arXiv.2304.00501.